University of Ulm | 89069 Ulm | Germany

Faculty of Engineering and Computer Science
Institute for Databases and Information Systems

A Generic Approach for Developing and Executing Electronic Questionnaires on the iOS Platform

Master's thesis at University of Ulm

Submitted by:
Dominik Deuter
dominik.deuter@uni-ulm.de

Reviewer:
Prof. Dr. Manfred Reichert
Dr. Vera Künzle

Advisor:
Johannes Schobel

2014

Version of August 12, 2014

PDF-$\LaTeX\,2_\varepsilon$

Abstract

The creation of questionnaires is a very time-consuming and costly task when developing a study. In most cases they are created in a paper-based form. However, this paper-based approach leaves less scope for interaction with the participant to motivate them when filling in the questionnaire. Moreover, data collection and subsequent evaluation is very cumbersome because all data has to be transferred manually to electronic worksheets.
To deal with these issues, this master's thesis presents a concept and implementation of an electronic questionnaire application to solve the mentioned problems. Thereby, different generic approaches are introduced and compared with each other. In general, questions and answers are determined using an XML document. In addition, a generic XML schema is provided to validate the specified questionnaires during the creation. When running such a questionnaire, user interface elements like *textfields*, *checkboxes*, *radiobuttons* or *sliders* are automatically generated by the application. Moreover, the *results* of completed questionnaires can be *viewed* and *exported* for later analysis.

Requirements and *styleguides* for an iOS specific application development are determined an discussed. The theoretical characterization ends in a practical part whereby a possible scenario for the utilization of the electronic questionnaire application is shown. In addition, the steps to create, deploy and enact such an electronic questionnaire are discussed.

Acknowledgments

At this point I would like to thank everyone who has supported me during the preparation of this master's thesis and during my student days.
Special thanks go to Prof. Dr. Manfred Reichert who has always supported and promoted me during my academic studies. To my advisor Johannes Schobel, who supported me excellent through his feedback and well-founded know-how during the master's thesis. To Rüdiger Pryss, who gave me several assistant jobs which enabled me an exciting insight in the work at the Institute of Databases and Information Systems, whereby I could gain a lot of experiences.
I also wish to thank my friend and fellow student Marcel Reichersdörfer. Through the mutual support and motivation, we successfully completed our study. Also I would like to thank Aday Onar I have get to know during the study, whereby, a very good friendship has developed.
My greatest thanks although go to my parents which always supported me and enabled me the possibility to study through their financial support.

Nördlingen, July 2014

Contents

1 Motivation

In many domains such as science, healthcare, psychology or industry data collection is done by using specifically tailored questionnaires to get information about specific circumstances. In addition, questionnaires are used for different kinds of studies or serves as checklists for maintenance jobs. The data collected is then evaluated and further processed using electronic worksheets or statistical applications.

In former times most of the processing of questionnaires was carried out paper-based both, the completion as well as evaluation. This caused in a very cumbersome way to handle hand-written text and even led to errors when transcribing the data collected. For this reason electronic questionnaires provided using a web-browser or running on smart mobile devices are preferred nowadays. In this form, more comfort for the addressed aspects will be guaranteed through the new opportunities by using different software tools.

The motivation for this master's thesis is the optimization of developing, executing and analyzing electronic questionnaires.

It turned out that the addressed process is currently combined with a high effort in time, resulting in more costs. This is because every element like question-labels, buttons, text fields and other user interface-objects in a questionnaire have to be aligned in detailed work separately. Moreover, these elements have to be linked and integrated in a complete application. In addition, the applications are hard-coded, which means, that every single questionnaire has to be created individually and after updating an electronic questionnaire every time the complete application has to be newly deployed which actually requires an enormous effort. However, as soon as an electronic questionnaire approach is used, IT experts are necessary for the implementation, whereby domain-specific information about the questionnaire potentially getting lost. In contrast to paper-based approaches the domain expert was the developer of questionnaires at the same time.
For this reasons, thoughts were aloud for an optimization of this process to reduce time and costs for developing such electronic questionnaire applications and make the usability more comfortable even for non-IT experts.

1.1. Contribution of this Master's Thesis

As mentioned in the previous section the goal of this thesis is to provide a concept to fasten the development and maintenance of electronic questionnaires. The approach provided here is to move away from a static and individual, hard-coded implementation. Instead a generic solution is pursued. Using this approach, it will be possible to generate fully automatically questionnaires for different application domains. Thereby, various benefits not only with respect to time and costs to develop and maintain a questionnaire but also in scalability and simplicity are provided.

The fundamental concept is based on a typical model in computer science called *Input-Process-Output-Model* which is going to be explained in detail in chapter 2.3. Modified to the generic questionnaire issue the idea is to specify a textual document in *XML* notation. This document represents a questionnaire, its structure, all questions and their possible answer values. It serves as the input for the actual electronic questionnaires application running on a smart mobile device. This application allows the enactment of the questionnaire. The implementation of that electronic questionnaire is carried out on Apple's *iOS*-based iPad. The main concept, however, is assignable to other platforms like Google's *Android* or Microsoft's *Windows Phone* without any major adaptations. However, a native development for a specific platform is preferred over a cross-platform approach using web technologies like *PhoneGap* to guarantee the *look & feel* for a platform specific application. To sum it up, a generic software application for the electronic data collection and evaluation for such mobile devices use is developed.

1.2. Structure of this Thesis

This section provides an outline on the further course of this scientific work. In chapter 2, all necessary terms in context of the topic of this thesis are listed and explained in more detail. This ensures a basic knowledge for reading this thesis. Chapter 3 summarizes other approaches and technologies in scientific research on the topics to be addressed. Based on this knowledge the requirements and design for the software application are elaborated in chapter 4. The concept is oriented to the *Reference Model of Software Development* in *Usability Engineering* [Off12] which structures the whole software engineering cycle into different phases and process-steps. Moreover, different approaches to solve the discovered problems are discussed and evaluated.
Based on one selected approach, chapter 5 gives further insights into the technical implementation. Furthermore, the application architecture will be explained. For an better imagination with respect to the final implementation, chapter 6 describes a complete application scenario. Chapter 7 provides a critical discussion for the prototype application and highlights insights gained during the development and implementation. Chapter 8 concludes this thesis with a summary and provides an outlook on further improvements.

2 Fundamentals

This chapter deals with fundamentals required for the further understanding of this thesis and provides detailed information. It assures a fundamental comprehension for the reader. A number of several techniques and basic models of computer science which are used are explained. Furthermore, an introduction to the development for *iOS* as well as *Usability Engineering* is given. Finally, topics such as *Data Mining* and *encryption* in software applications are introduced.

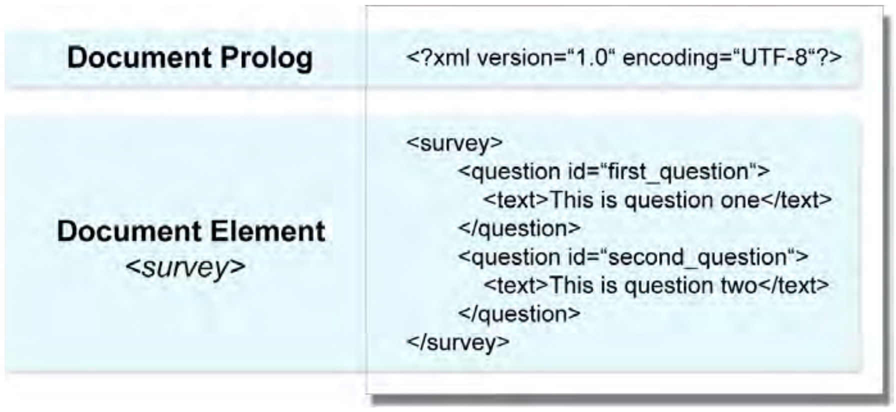
2.1. Extensible Markup Language

Extensible Markup Language (XML) was developed by the *World Wide Web Consortium* (W3C) in 1996. It is a construct designed for data handling which is used to store information. In the following an extract of design goals for the XML technology is linked: [BSM96, Shi]

- **Human-Readable:** The structure and content of an XML document should be easily readable for humans as well as for computers.
- **Fast Preparation:** The XML design should be prepared without a high effort. Thereby the compilation of the actual content moves into focus.
- **Easy Creation:** XML documents should be easy to create without having to use special software. A common text-editor is sufficient.
- **Distributed:** XML can be used as a distributed technology which shall be simple usable over the Internet.
- **Wide Range of Applications:** XML shall support a wide variety of applications which can be easily interpreted and processed by them.
- **Simple Processing:** It shall be easy to write programs which can process XML documents. By its structure it is possible to parse the contents from the document without a high effort.

By complying and pursuing these goals, XML became an essential technology in data processing nowadays. For a better understanding XML will be described in more detail at this point.

An XML document consists of two parts [Ray03]. The *document prolog* represents the header which contains different metadata information about the file (e.g., the character encoding or version of the document). The second part is the *document element* which defines the actual information storage. It is the root element which contains other XML elements with their information content in a hierarchic structure. For a better illustration figure 2.1 shows a possible set-up.



```
<?xml version="1.0" encoding="UTF-8"?>

<survey>
    <question id="first_question">
      <text>This is question one</text>
    </question>
    <question id="second_question">
      <text>This is question two</text>
    </question>
</survey>
```

Figure 2.1.: Structure of an XML Document

An XML element is represented by using so called *tags*. The actual information is contained between an opening and closing *tag*. Referring to figure 2.1 the *document element* `survey` (defined by `<survey></survey>`), contains multiple `question` elements which finally contain the information content "*This is question #*", defined in the XML element `<text></text>`. In that way it is possible to transform any real world object into a machine readable form. In addition, every XML element can be enriched with additional attributes which ensuring more details. For example, all `question` elements have an attribute `id`, providing an unique identification.
Figure 2.2 shows the XML syntax of one *(container-)element* and the syntax of an attribute.

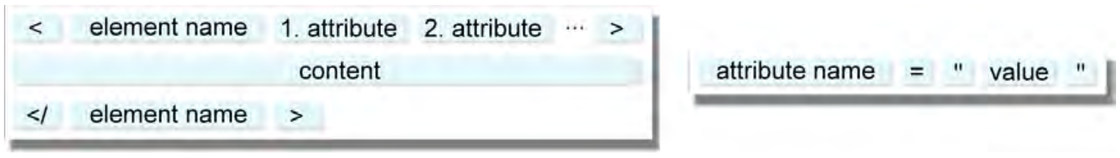


Figure 2.2.: Syntax of Elements (left) and Attributes (right)

As mentioned in chapter 1, in this thesis XML is used for the description of an individual electronic questionnaire. In chapter 6 the creation and use of XML documents as questionnaires for the application will be shown in a more practical context. After designing an XML document it can be validated with the help of a scheme, a so called XSD. This is described in more detail in the next section 2.2.

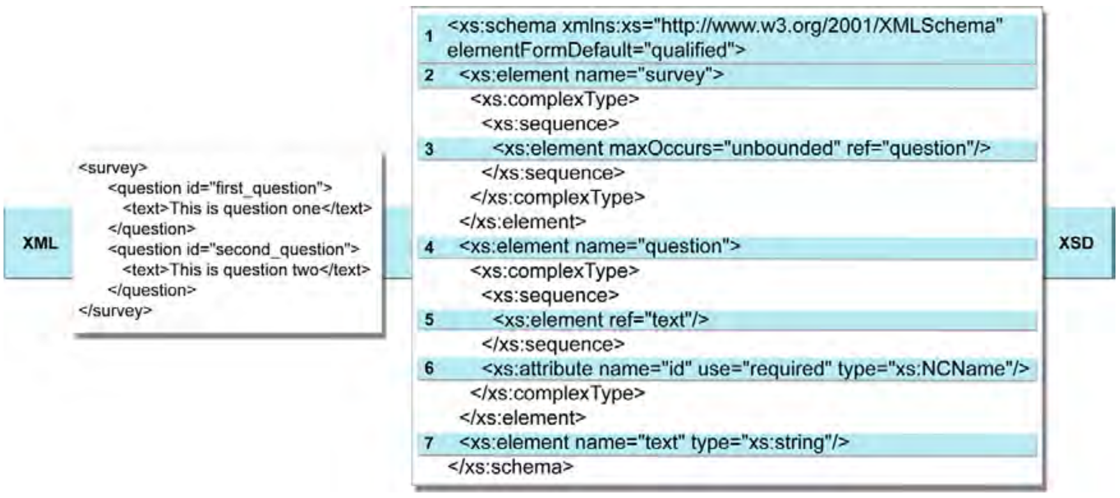
2.2. XML Schema Definition

To make sure a specific XML document (which represents a questionnaire) is suitable for the electronic questionnaire application, it has to be validated. For this reason, it is necessary to define a global scheme which is readable for the application. All created XML documents must be checked and validated against an *XML Schema Definition*, in short XSD. Only if a specific XML document is valid in respect to the given schema it can be correctly interpreted and processed by the application.

XSD has been adopted by the *World Wide Web Consortium* in May, 2001. Similar to DTD (*Document Type Definition*), it defines rules for the syntax and structure of XML documents [HKS02] (more information about DTD in [FAQ12] and [KA09]). In comparison to DTD, XSD provides more complex rules to describe a document. In the following, numerous advantages and characteristics of *XML Schema Definition* are provided [HKS02]:

- **No Additional Effort:** Every XML schema is a XML document by itself. So no additional syntax is needed or must be learned.
- **Accurate Definition:** The required contents of *tags* and *attributes* can be precisely specified. [Ker03]
- **Determination of Data Types:** XML schema contains a big set predefined data types and enables the user to define own data types.
- **Used Concepts:** XML schema supports inheritance and substitution such as in object oriented languages.
- **Encapsulation:** The XML schema guarantees modularization and reusability.

If such a XML schema was defined, all XML documents following this schema can be validated. Figure 2.3 presents an example of such an XSD. Note that the XML document provided in figure 2.1 is valid in respect to this XSD.



```
<survey>
    <question id="first_question">
        <text>This is question one</text>
    </question>
    <question id="second_question">
        <text>This is question two</text>
    </question>
</survey>
```

```
1 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
2  <xs:element name="survey">
     <xs:complexType>
       <xs:sequence>
3        <xs:element maxOccurs="unbounded" ref="question"/>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
4  <xs:element name="question">
     <xs:complexType>
       <xs:sequence>
5        <xs:element ref="text"/>
       </xs:sequence>
6      <xs:attribute name="id" use="required" type="xs:NCName"/>
     </xs:complexType>
   </xs:element>
7  <xs:element name="text" type="xs:string"/>
  </xs:schema>
```

Figure 2.3.: XML and XSD Comparison

Every element in a XSD will be declared as an `<element>`. Basically, elements which only contain a simple information (like a string or a number) will be defined using the attribute `type` (e.g., the `text-element` in line 7). Nested elements (e.g., the root element `survey` or the element `question`) which contain different information or attributes needs to be defined as `complexType` (like line 2 and line 4) [Ker03]. In line 1 of the given XSD document the used elements are requested by the namespace of `http://www.w3.org/2001/XMLSchema`. Line 2 defines the questionnaire (`survey`) which contains all other elements. Line 3 references a question element and defining its maximal occurrence. The question element itself is defined in line 4. Note, that a questionnaire can have an infinite amount of questions. Furthermore, line 5 reveals the affiliation of the element `text` with an attribute `id` in line 6 for the question. The `text-element` is defined as `type="xs:string"`.

All XML documents which serve as questionnaires have to pass the validation through such a schema definition. If this test is passed, the application is able to create an executable instance of an electronic questionnaire. Thereby, the *correctness by construction* principle is used which allows to prevent any syntactical errors when integrating in the application. To sum it up, XML is a common technique in the connection with XML schema to store data in a hierarchical structure.

2.3. Input-Process-Output-Model

In software engineering as well as in the entire domain of computer science the *Input-Process-Output*-Model is a basic principle which regulates the whole data-processing-cycle. It serves as a basic of the concept of the electronic questionnaire application. The model consists out of three components which are described below [Som10]. Moreover, figure 2.4 illustrates this concept.

- **Input:** An input component serves as data reception for the application. It reads the incoming information (e.g., from a file or database) and passes it to the processing component.
- **Processing:** This component is the main part of the application, as it performs the computation of the data. New data records will be created from the results of the computations. Finally, the processed data will be passed on the output component.
- **Output:** The output component reads the processed data and transforms them into the desired output format (e.g., a printed document, a database entry or a regular file).

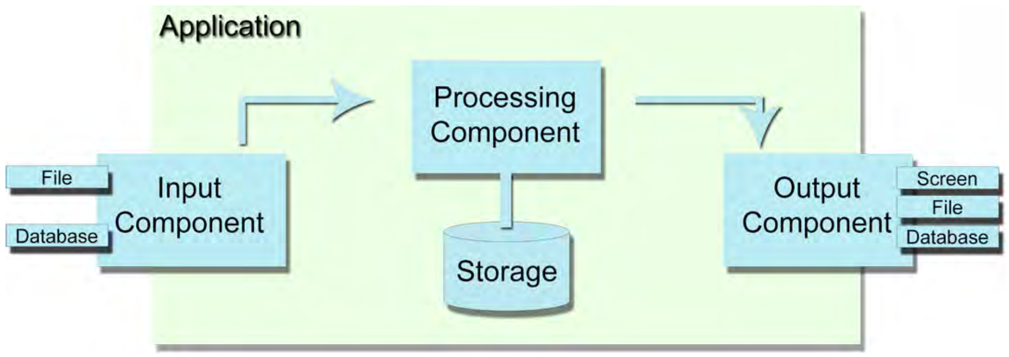


Figure 2.4.: Input-Process-Output-Model

2.4. Apple & iOS Development

As already mentioned, a prototype application based on the developed approach is exemplarily implemented for Apple's operating system *iOS* on an up-to-date iPad device. For this reason, this section introduces iOS development and its programming language *Objective-C* briefly.

As Apple introduced the iPhone in 2007 and iPad in 2010, no one could have guessed how these smart mobile devices would change our daily life. Now it was possible to request information on the way and carry out applications on places that had not been feasible in former times.
According to a statistic in figure 2.5, Apple is currently (2013) the number one in *world-wide tablet sales to end users by vendor* and number two behind Android for the *worldwide tablet sales to end users by operating system*. [Gar]

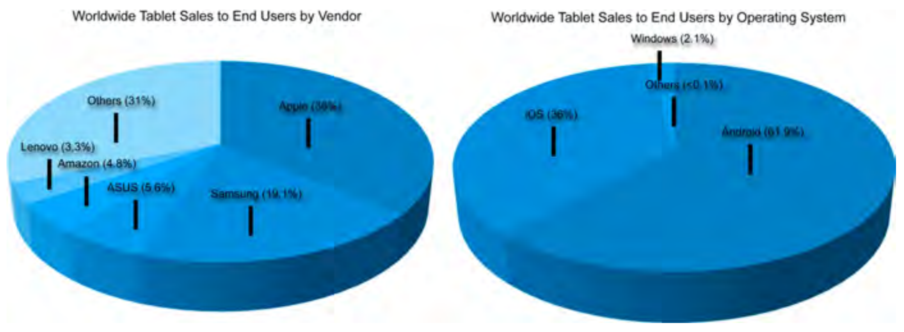


Figure 2.5.: Market Share of Sold Tablets (Worldwide) 2013

Under this conditions it makes sense to use the iOS platform for implementing the approach for an generic questionnaire. However, it should be mentioned that the concept provided in this thesis is kept generic, so it can be easily transferred to other platforms.

For an application running iOS devices (like the iPad) the best way to implement is to use the programming language *Objective-C* [Obj12] in combination with the *Cocoa Touch-Framework* [Coc12]. This object oriented language is an extension of *C* and was developed in the 1980s. Code examples of *Objective-C* will be shown in chapter 5. Currently *Objective-C* is available in version 7.1.1 (May, 2014). However, through

the generic concept the software will be supported by version 5 and higher. As a development environment Apple's in-house software *Xcode* in version 5.1 (March, 2014) is used.
In the next section an introduction to *Usability Engineering* and the general planning-process of project development will be given.

2.5. Usability Engineering

Usability Engineering is a part of computer science that concerns to improve the ergonomic of (software-)products. Jacob Nielsen defines *Usability* as follows [Nie12]:

"Usability is a quality attribute that assesses how easy user interfaces are to use. The word 'usability' also refers to methods for improving ease-of-use during the design process".

Usability is traditionally associated with five usability attributes defined in [Nie93]:

- **Learnability**: The system should be easy to learn so that the user can rapidly start getting some work done with the system.
- **Efficiency**: The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.
- **Memorability**: The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- **Errors**: The system should have a low error rate, so that users make few errors during the use of the system, and if they make one, they easily recover from them. Further, catastrophic errors must not occur.
- **Satisfaction**: The system should be pleasant to use, so that users are subjectively satisfied when using it; they like it.

Considering the whole system, acceptability and *usability* play an important role which have been taken into account in software development as shown in figure 2.6.

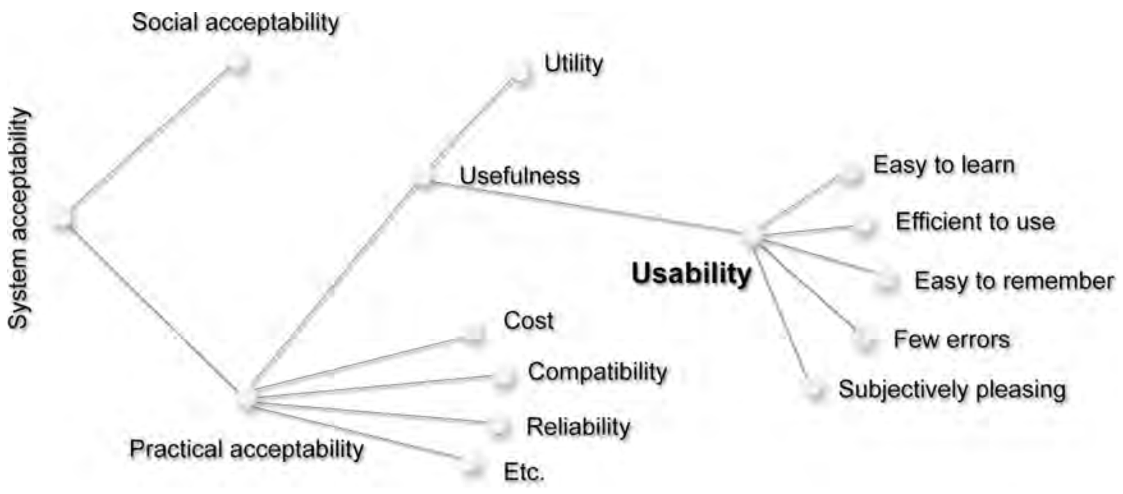


Figure 2.6.: A Model of the Attributes of System Acceptability [Nie93]

To guarantee usability in the electronic questionnaire application, a great effort is put into *Usability Engineering*. The user should be able to use the application without any prior knowledge and any technical capabilities.

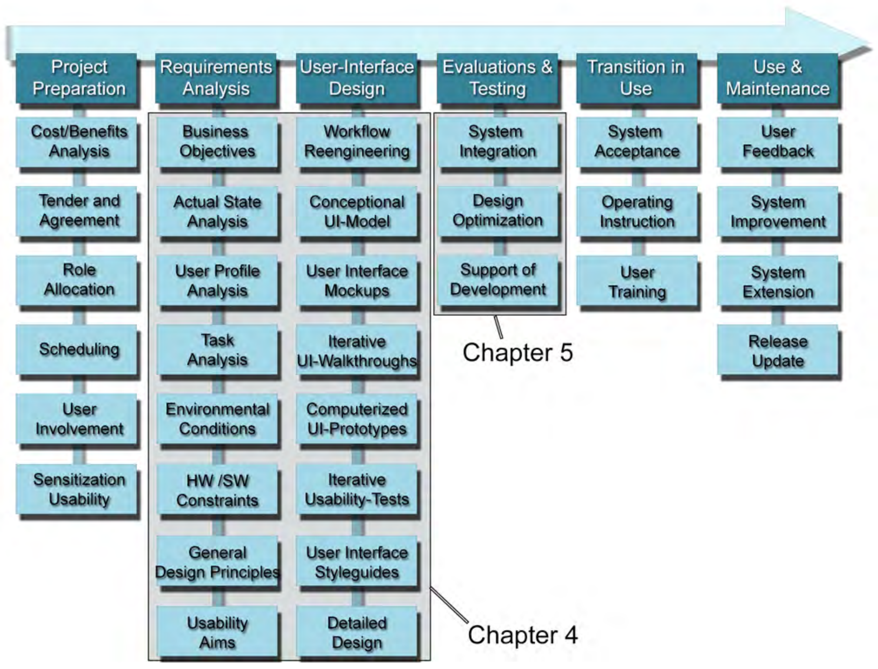


Figure 2.7.: Reference Model of Software Development [Off12]

[Off12] addresses the so called *Reference Model of Software Development* in *Usability Engineering* (see figure 2.7). This model describes a whole system engineering process and is structured into different phases containing several process-steps. Fundamentally, the development of the project described in this thesis refers to a modified form to this model. The phases *Requirement Analysis*, *User-Interface Design* and *Evaluation & Testing* are discussed in chapter 4 and 5.

2.6. Mobility and Privacy

As mentioned before an application for enacting electronic questionnaires running on tablet computers (e.g., the Apple iPad) should be developed. Hence, a deeper insight into the research of tablets is done. The question therefore arise, what *mobility* is and how it is defined in research.

The world of wireless connectivity gives people the flexibility to do business anywhere they are. Moreover, this enables global connectivity anywhere and anytime. Instead of being tied to a desk with a normal computer, people will be contactable around the world. This is primary because of research and development in wireless communication technologies, such as *WLAN* or *cellular networks* [Wie07]. In science, this field is called *Mobile Computing* which can be defined as following [Web13]:

- *Mobile Computing is using a computer while being on the move.*
- *Mobile Computing is when a (work) process is moved from a normal fixed position to a more dynamic position.*
- *Mobile Computing is when a work process is carried out somewhere where it was not possible previously.*

This results in new ways for (business-)application and communication, so it is the start of a new period. For a characterization, different *types of mobility* are specified (basing on [Sch03] and [Sch05]):

- **Wireless non-mobility:** Using wireless communication at a fixed place without any mobility aspect.

- **Static mobility:** Using wireless devices of different places. For Example, moving the notebook from the office to a conference room.

- **Dynamic mobility:** Using devices while on the move. For Example, surfing the web on the railroad.

- **Ubiquitous computing / pervasive computing:** Using devices without noticing them. For Example, automatic payment, while walking out of the supermarket.

Another view, with respect to *Mobile Computing*, is *data mobility* and the associated *privacy* issues. When using a device which is connected with a (wireless) network, people leave digital traces in information systems. This allows to track human activities in a territory due to the pervasiveness and positioning accuracy. Smart mobile devices which are integrated ubiquitously and pervasively in the environment (e.g., smart home or monitoring systems) and applications carried out everywhere and always by people, data can be sensed and collected in a specific territory [GP08].
Mobile communication networks are described as an infrastructure to gather mobile data if the location of its users is tracked at different times. However, it is quiet complex to extract knowledge from mobile data. In another field of research, called *Data Mining* (described in section 2.7), scientific approaches and techniques for extracting knowledge from data are researched. For the development of an electronic questionnaire it is necessary to consider user's privacy.

The authors of [BS93] and [Lan01] have established privacy principles for *Ubiquitous and Mobile Computing*. With consideration to this guidelines the user can have a pleasant and secure feeling in utilization of the application. Figure 2.8 shows different classifications of privacy guidelines, which should be attached to a great importance in development. On the *user level*, the user must be getting feedback and notice from the application and must be have the full control about all privacy settings. On the *system level*, the application must be guaranteeing that no context information such as *proximity & location* have to be transmitted without explicit consent of the user.

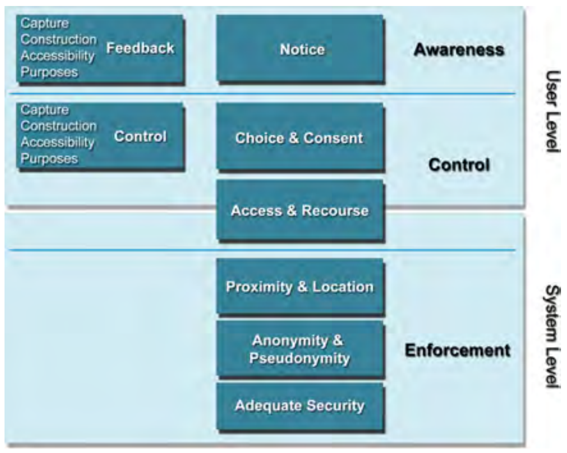


Figure 2.8.: Privacy Guidelines [Web13]

2.7. Data Mining

In the field of *Data Mining* researchers try to extract knowledge from different sets of data with various methods. *Data mining* refers to *extracting* or *mining knowledge* from large amount of *data* [HK00]. Furthermore, it is a sub process of *Knowledge Discovery in Databases (KDD)*. [FPSS96] defines *KDD* as follows:
"Knowledge Discovery in Databases (KDD) is the non trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data."
During the development of electronic questionnaires, *data mining* could be used for processing and evaluating data after collecting. At this point different approaches for knowledge extraction will be discussed. Figure 2.9 illustrates the whole *KDD Process*.

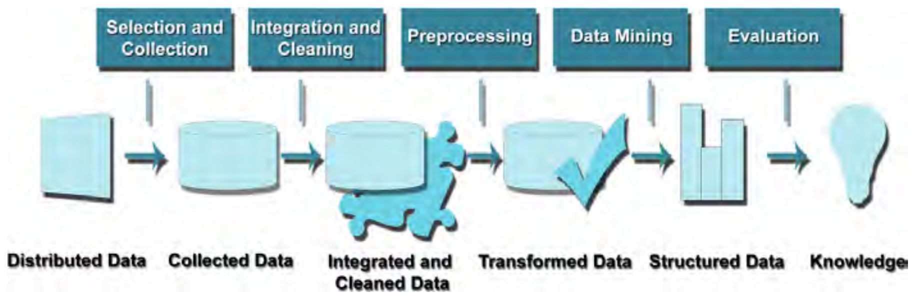


Figure 2.9.: KDD Process [FPSS96]

In the first step of this process, all distributed data from different sources have to be collected. Afterwards these data must be integrated in the actual application and will be prepared to transform into a unified format. In the next step the actual *data mining* takes place. In that steps, data is compared with each other to derive similarities and classify them into groups. Finally, these classes are evaluated and knowledge can be extracted.

There are different approaches of structuring data using *data mining*. For example, classification with the help of an *decision tree* and *clustering*. These approaches are discussed in more detail in the following.

2.7.1. Decision Tree

A *decision tree* helps to derive classes (contained in the leaves of the tree) from a set of data. Thereby rules will be defined for classification. In figure 2.10 an example is illustrated which is adapted to a questionnaire topic. It can be defined as following:

Classes:
$Health\ Assessment \in \{Invulnerably,\ Tolerable,\ Harmful,\ Irresponsible\}$

Attributes:
$Gender \in \{male,\ female\}$
$Alcohol \in \{no,\ less,\ much\}$
$Pregnant \in \{no,\ yes\}$

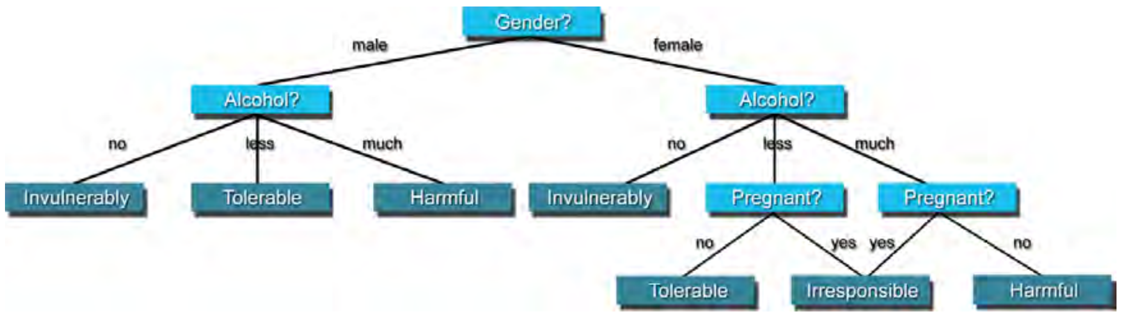


Figure 2.10.: Decision Tree for a *Health Assessment*

A classification can be made by starting at the root element and descending the decision tree until a leaf element is reached. Classes can be converted into rules by using *if-statements*. A possible rule for the decision tree in figure 2.10 could be the following:

***if**(Gender == male)* $\wedge$ *(Alcohol == much)* ***then** class = Harmful*

For constructing a so called perfect *decision tree* it is necessary to measure the information gain of every attribute. In descending order an attribute divides the data set in a subset which containing only elements of the same type. A well-known algorithm for constructing a perfect *decision tree* is called *ID3 algorithm* which determines the information gain with the help of *entropy*. Further information to *entropy* and *ID3 algorithm* can be found in [YS95].

2.7.2. Clustering

Another approach of *data mining* (for getting knowledge of a data set) is *clustering*. As the name indicates, the input data is partitioned into different clusters. That means data within a cluster are similar. Algorithm for *clustering* transforms data into objects which are represented by a matrix. This matrix contains information about distances (similarities) between objects and the data element for the object. Hereinafter, a mathematical definition of *clustering*:

Given: n objects $G = \{e_1,\, e_2,\, ...,\, e_n\}$ with property vectors $\{x_1,\, x_2,\, ...,\, x_n\} \in \mathbb{R}^d$

Wanted: Clustering C with $k \in \{1,\ ...,\, n\}$ clusters $C = \{C_1,\, C_2,\, ...,\, C_n\}$
with $C_j \subset G$ not empty, $C_i \cap C_j = \emptyset$ for $i \neq j$ and $G = C_1 \cup ... \cup C_k$

The theoretical approach for creating clusters is described as follows:

1. Select distance or similarity measure and specify a rating function D to evaluate the quality of the cluster C.
2. Select a number of clusters k.
3. Calculate the optimal clustering with $C_{opt} = min(D(C))$.

For *clustering* there are two well-known algorithms called *hierarchical (agglomerative, divisive)* [Joh67] and *partition-based (k-means) clustering* [HW79].
With experiences of *data mining* in relation to the electronic questionnaire a derivation of knowledge from data which are raised by questionnaires is possible. In that way, it is easier and more efficient to evaluate collected data.

2.8. AES Encryption

The data for the electronic questionnaire application is encrypted by the *AES Algorithm*. (Information is based on [Wil04, For14]). *Advanced Encryption Standard* is a symmetric block cipher which means that one key is used for decryption and encryption of data. Thereby, there are different *key length* and *block length* (128, 192, 256 bit). Cracking a 128 bit *AES* key with a state-of-the-art supercomputer would take longer than the presumed age of the universe (about 14 billion years) [box14]. The *AES Algorithm* is executed in *rounds*. The count of the *rounds* is depending by the *key and block length*.

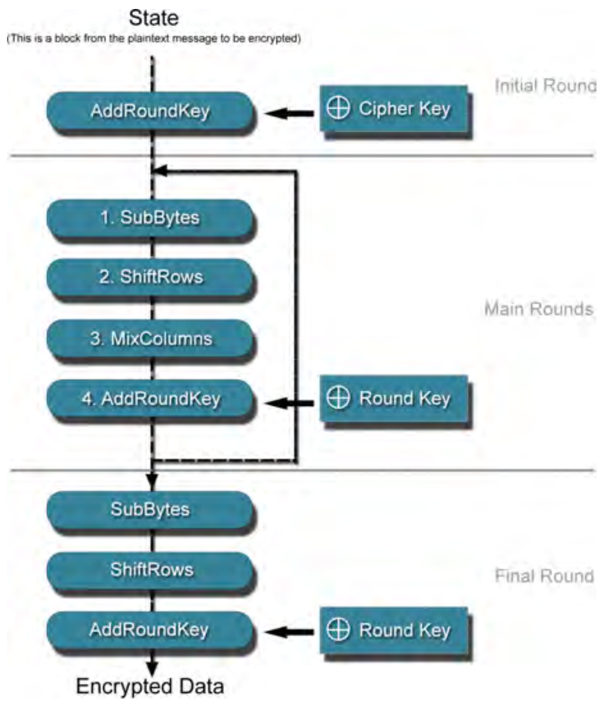


Figure 2.11.: AES Encryption Process [For14]

Figure 2.11 gives a general overview of the steps and components of the *AES Algorithm*. Thereby, a *State block* from the *plaintext message* serves as input and is *XOR*-concatenated with a defined *Cipher Key* by the *AddRoundKey* component. Afterwards, depending on the used *key* and *block* length, the further components are repeated in *rounds* whereby every *AddRoundKey* call is *XOR*-concatenated with a *round* generated key. After the *AES Algorithm* was executed, input data is encrypted. A decryption of *AES* can be done with an *inversion* of the steps. The *AES Algorithm* is used in the electronic questionnaire application for the encryption of the data (e.g. results of completed questionnaires) stored in a database.

3 Related Work

This chapter deals with different research approaches for developing electronic questionnaires. Furthermore, design rules for questionnaires in general are determined and software & implementation aspects of mobile applications are presented.

The following sections are divided into topics, mentioned to get an overview about different perspectives of developing generic electronic questionnaires.

3.1. Questionnaire Design

For the development of electronic questionnaires, it is important to get a general understanding of how to create and design a questionnaire. This includes finding a suitable medium as well as items for data collection.

In [DGF^{+}95] an interesting study was performed which evaluated the use of pen-based electronic and conventional paper-based questionnaires. The experiment involved 46 participants between 17 and 81 years, throughout three visits in a test environment. During the first visit the participants were familiarized with the paper questionnaire. In the subsequent visit they completed the electronic as well as the paper questionnaires in a randomized order. During the last visit they completed the preferred questionnaire again. The result of the study showed a high acceptance of electronic questionnaires: 57% of participants chose the electronic whereas only 13% preferred paper. The remaining 30% expressed no explicit preference at all. All participants found both, paper and electronic media easy in handling. Data collection was more accurate on electronic questionnaires (100%) than on paper (99.1%). In summary, the study showed major benefits in respect to data completeness, data flow and further processing of the data obtained by the use of such electronic questionnaires.

Another study evaluated the data collection of sensitive information with handheld computers (PDAs) compared to regular paper-based questionnaires [BO^{+}08]. In the results of 200 participants (18-29 years), 86% showed a general agreement of both methods for data collection. However, when using paper-based questionnaires, the number of inconsistencies and missing values were significantly higher. The electronic questionnaire contained a set of data entry types like *pop-up lists, multi-option answers, one-option answers,* etc. Some questions were only asked if the response to a previous question met a predefined rule. Moreover, the application allowed participants to return to previous questions within the same section.

Beside the two studies explained in the previous paragraphs, [Bru12] adds further interesting aspects in the comparison of paper and electronic questionnaires. The author notes that a paper-based questionnaire is strictly organized as the researcher has to contribute the questionnaire directly to their participants. However, an electronic questionnaire has more advantages in its way to handle the process of collecting data. For example, it is possible to publish the questionnaire on the internet which increases the the number of participants. Moreover, an electronic questionnaire allows to protocol important meta information like timestamps to get information about the answering time of a question. In paper-based questionnaires it is more complicated to such meta information.

After analyzing paper-based, electronic questionnaires, scientific approaches of finding suitable questionnaire-items for data collection are considered.
[RJ07] discusses how to determine items in questionnaire design and development for the health care domain. It describes the situation that the number of questionnaires developed by nurses has increased significant during the last years. As such nurses are usually not trained in using state-of-the-art questionnaires an improvement and guideline for that issue is needed. Therefore, the paper examines the process by which a reliable and valid questionnaire can be developed.
At the beginning, the authors ask themselves what the questionnaire will measure. They noticed that nurses use questionnaires to measure knowledge, emotion or behavior of a patient. When developing a questionnaire, different items or questions have to be generated for the participants. In the next step, the responses of participant are converted into a numerical representation in order to allow for a statistical evaluation. In table 3.1 a guideline for item generation and scale construction is illustrated. Thereby, different questions are asked about the motivation of the questionnaire inclusive *key issues* which offers a variety of different suitable techniques and methods.
However, the authors criticize questionnaires in this paper. They explain that questionnaires are not always the best method for data collection. Especially, when little is known about the subject or topic area, qualitative methods (like hermeneutic methods) may be more appropriated. The authors conclude that the design and development of questionnaires must be supported by a logical, systematic and structured approach.

Questionnaire Development	**Key Issues**
What will the questionnaire measure?	Knowledge Attitude/beliefs/intention Cognition Emotion Behaviour
What types of scale can be used?	Frequency Rasch [SCCP02] Thurstone [PMGC81] Guttman [PMGC81] Mokken [VS03] Likert type [PMGC81] Multiple choice
How do I generate items for my questionnaire?	Ensure relevance of items? Wording issues Which response format is best? Which types of question are possible? Free text options? Does your measure have subscales? Questionnaire layout

Table 3.1.: Item Generation and Scale Construction [RJ07]

3.2. Implementation Aspects

In this section, aspects for implementing mobile applications are discussed. For this reason an overview of different platforms, which are conceivable for the implementation of electronic questionnaire application is given. A comparison between Google's Android and Apple's iOS and guidelines for considering security issues in application development is shown.

3.2.1. Operating Systems

First of all, [GR11] provides a comparison between hardware requirements. The authors thereby criticize that iOS development is restricted to Apple's Macintosh running a MAC

OS X. In contrast, Android applications can be developed using any major operating systems like Windows, Linux or MAC OS X.
iOS apps have to be written using the *Xcode SDK* provided by Apple. *Xcode SDK* allows developing for iOS as well as to implement applications for MAC OS X. Beside Android Studio, Eclipse is the recommended and most used development environment for Android. It is necessary to install the *Android SDK plugin* for Eclipse to write apps. Also the SDK offers a debugger and GUI builder for user interfaces. The programming language for iOS is *Objective-C*, an extension of *C*. However, Android applications are implemented in *Java*. The API references of both are vast, precise, comprehensive and free. For deploying a developed application on a real device it is necessary to get a *developer account* for 99$ a year. For deploying Android on real devices is for free (for a deployment in the Google Play Store a membership of 25$ is necessary). Table 3.2 summarizes characteristics of iOS and Android.

Characteristics	**iOS**	**Android**
Operating System for Development	MAC OS X	Windows, Linux, MAC OS X
Development Devices	Apple Devices (iPad, iPhone)	Many Vendor Devices (Google, Samsung, HTC, etc.)
IDE, GUI Creation	Xcode	Eclipse (Plugin), Android Studio
Main Language	Objective-C, Swift	Java
Reference Website	developer.apple.com	developer.android.com

Table 3.2.: Characteristics iOS vs. Android [GR11]

Another approach for mobile application development are so called *cross-platforms* which correspond to the *application development*. With such frameworks it is possible to implement the application just once and deploy it on all supported platforms. The advantage is saving time and costs by only developing one single implementation for every platform. However, it is an disadvantage at the same time due different architectures and concepts (such as platform depending guidelines). So the *look & feel* differs from platform specific applications which are especially developed for a single platform (keyword: *native development*). [SSP$^+$13] gives an overview about different architectures and a description between *native*, *hybrid* and *web* (webpages, which act as a mobile application) approaches.

3.2.2. Security in Software Projects

In this section, security issues for software development are represented [AP05] and [GWSB03]. Usually, project life cycles follow an iterative process of analysis, design, implementation, testing and maintenance as already mentioned in chapter 2. To ensure security in software, [AP05] describes an approach where in every stage (without maintenance) of the project life cycle security concerns are inserted. Figure 3.1 illustrates the *secure software development life cycle*. From a security viewpoint, projects generally start with *security requirements and analysis* in which the software security environment and objectives are defined, which results in a security policy that is used to evaluate risks. After that the *security design* takes place. In this phase common design methods (e.g., *UML*) are adapted to security concerns such as critical data or communication security by using *UMLsec*. Afterwards the *implementation* proceeds in which through using secure algorithm, security vulnerabilities should be avoided. When the implementation is finished a *testing security phase* takes place. Thereby, targeted security attacks are performed to determine security vulnerabilities. At this stage, the project is operational and enters a *maintenance phase*, which often loops back to requirements and analysis of new features or errors.

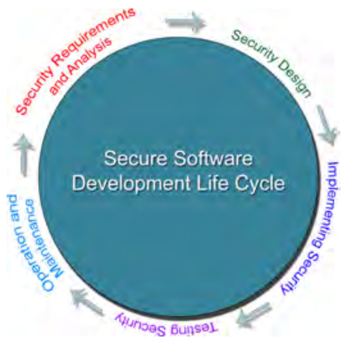

Security Requirements and Analysis
Security Design
Implementing Security
Testing Security
Operation and Maintenance
Secure Software Development Life Cycle

Figure 3.1.: Secure Software Development Life Cycle
[AP05]

[Man02] found out that a bad programming style and inadequate software development life cycles often leads to the development of poor software among others for weaknesses in security. For this, [GWSB03] has focused on defining a checklist for security in software (see table A.1) which should be considered when developing software products.

3.3. Similar Electronic Questionnaire Applications

In this section different *electronic questionnaire applications* are described which deal with the same aim of developing electronic questionnaires. Thereby, these applications are analyzed and compared with the intention of the concepts described in this thesis.

- **Snap** [sna14] features by creating interactive questionnaires for mobile or browser applications but also supports the generation of paper-based questionnaires. The application offers a questionnaire designer in which questions (e.g., *single-choice or multi-choice*) can be selected via drag & drop. This designer allows to create multi-language questionnaires, including right-to-left Arabic and Hebrew character sets, as well as Asian languages. *Snap* also supports media elements like *videos* and *selectable pictures* for questions. These interactive elements increase the motivation for enacting electronic questionnaires.
  For completed questionnaires, *Snap* offers analysis and reporting possibilities. The results are visualized in tables and diagrams to evaluate the participant's data. Furthermore, collected data can be integrated in *Excel* or databases such as *MS Access* or *SQL*.

  The approach of *Snap* is to make electronic questionnaires available over the internet (but is also usable in offline mode). In this way, participants world-wide can enact questionnaires online and therefore real-time analysis can be done in online mode. The electronic questionnaire application concept described in this thesis covers the offline and not the online scenario. The application must be installed on the device for enacting questionnaires in contrast to *Snap*. Afterwards, collected data must be manually uploaded to a server. So no real-time analysis is possible. Basically, both concepts (*Snap* and the approach described in this thesis) are similar. They offer possibilities for designing and enacting electronic questionnaires. Furthermore, an export for the data collected into databases is realized.

- **padCAPI** [pad] is an iPad questionnaire application for mobile data collection. The manufacturer points out that using an electronic questionnaire application, like *padCAPI*, a 20% shorter interview time in contrast to paper-based questionnaires is guaranteed. That is because the interviewer is relieved by many tasks such as simple tipping instead of writing with a pen. So the interviewer is fully able to focus on the participant.
  In contrast to *Snap*, *padCAPI*, however, does not contain a questionnaire designer. In order to transform a questionnaire set to an interactive application, it has to be transmitted to the company where the electronic questionnaire will be developed for the iPad specifically. Afterwards, the individual application is synchronized with the iPad via iTunes and can be used for enacting questionnaires. After collecting data, the application synchronizes and transmits the data to a web-based online portal for doing evaluations in table views as well as export possibilities for *Excel*. Moreover, *padCAPI* offers a multi-language support for enacting an electronic questionnaire in different languages as well as the concept described in this thesis.

- **SurveyPocket** [sur] is an application for enacting electronic questionnaires on iPad, Android tablets, iPhone and other smart devices. Above all it is distinguished by a huge selection of different question elements (50+). Beside standard elements like *checkboxes*, *radiobuttons* or *free-text* questions it offers *drop down menus*, *star ratings*, *barcode scannings*, *drag & drop rank orders* and even interactive *audio, video or image uploads and visualizations*. For designing questionnaires it is necessary to access a member web-portal. Afterwards, the designed questionnaire will be synchronized and is available on different mobile applications for enacting the questionnaire in online or offline mode (in offline mode the collected data are stored on the device locally). *SurveyPocket* allows (real-time) analysis in the web-portal and even on the mobile application itself. Furthermore, export possibilities such as *raw data*, *Word* or *CSV* are offered by *SurveyPocket*.

In general, all these electronic questionnaire applications support the phases of designing (except *padCAPI* where it is not possible to create questionnaires on your own), enacting and analyzing questionnaires. The applications allow to work in offline or online mode to do real-time analysis and distribute the collected data. Thereby, the data is synchronized to a web server. In general, the concept of an electronic questionnaire described in this thesis provides a similar approach which also includes the steps of designing and enacting questionnaires. However, the toolbox of questionnaire elements (e.g., *checkboxes*, *radiobuttons* or *textfields*) is much smaller. Also an analysis of collected data is featured to other software applications like *SPSS* or *Excel*. In contrast to the electronic questionnaire application, only an export as *CSV* or as raw data stored in a database is offered.

3.4. Summary

In this chapter some issues related to the development of electronic questionnaires were discussed. Statistical analyzes of *questionnaire design* with the help of studies were considered and a useful checklist for creating questionnaires was introduced.
Implementation aspects such as a comparison of different *operating systems* and the consideration to *security* in the *software development life cycle* were introduced. Last but not least, other *electronic questionnaire applications* which aim at the development of electronic questionnaires were described.

4 Application Design & Conception

This chapter covers basic issues in *application design and conception* which is crucial when developing a software application.
At first, all necessary requirements for implementing the smart mobile application are specified, which are divided into *functional* and *non-functional* requirements. Afterwards, technical guidelines as well as design styleguides are described. In addition, different technical approaches for the development of electronic questionnaires are discussed and evaluated.

4.1. Requirements

For a precise and accurate development for any kind of software application it is essential to define requirements for the application. Therefore, tasks which can be performed by users have to be determined. For this a *task analysis* is done to derive the application features. Beside the actual functions it is also important to define so called *non-functional* requirements (such as *usability targets*) which describe aspects affecting the application. Technical guidelines determine the use of questionnaire elements which are compliant with different versions of the iOS platform. Furthermore, styleguides guarantee the conformity of *User Interface Design* and increase usability.

4.1.1. Functional Requirements

For the development of an application it is required to determine features which characterize the application. Thereby, a *task analysis* is carried out which observes tasks performed by the end-users. On this basis, *use cases* are defined which the application should be able to process. These *use cases* are typically visualized in an *use case diagram* and specified with the help of *context questions* (see Appendix). In this *task analysis* the general workflows for using questionnaires were determined by investigating psychological and medical studies. This repertoire of features defined should guarantee an optimal usage of an electronic questionnaire application that should make the work of questionnaire design and enactment easier.

In general, after the questionnaire is created, using predefined elements (e.g., questions or texts), participants fill in the questionnaire. Subsequently, the data collected is processed and evaluated. The application reveals essential features illustrated in the *use case diagram* in figure 4.1, followed by a detailed description.

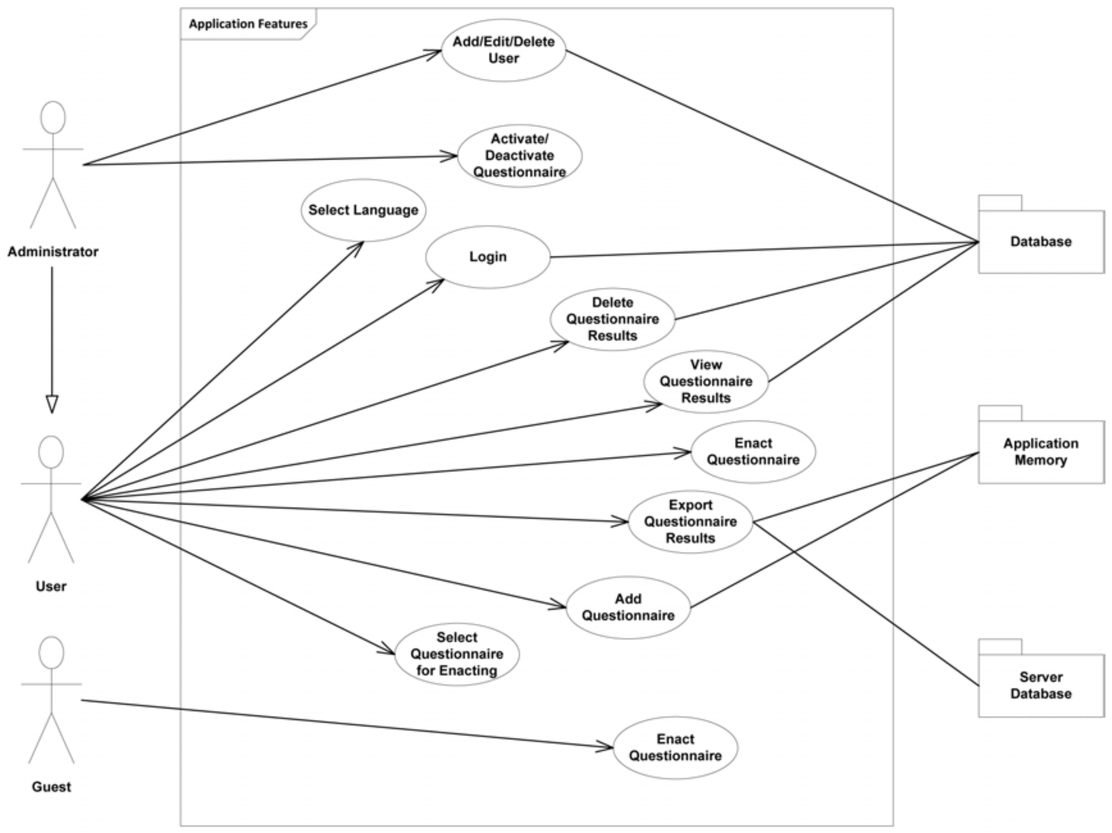


Figure 4.1.: Use Case Diagram for Questionnaire Design

For the application three different user roles are defined, namely the `User` (responsible head for data collection), `Administrator` and `Guest`. The `Administrator` can perform the same tasks like the `User` but additional *maintenance* tasks (presented by the inheritance arrow in the *use case diagram*). The `Guest` is only able to `enact questionnaires`. To manage the users and their roles, the application supports an *user management*. Only an `Administrator` is able to `add` and `delete` users for making an assignment between user and enacted questionnaire. This way it is possible to restrict the rights to a specific user. To realize the *user management* it is necessary to store users into a database (e.g., Apple's *Core Data* which is mentioned in chapter 5). In addition, application should be able to provide different languages for the execution of the questionnaire. This will allow the user to `select the language` which he wants. Furthermore, a *questionnaire management* allows `activating` and `deactivating` questionnaires during runtime. The application should allow for `integrating` new questionnaires, represented as XML documents. This enables the use of many various questionnaires within one single application. The execution of a questionnaire is only possible if the user is `logged in`, so that an assignment between user, questionnaire and results can be guaranteed (a guest user is only able to enact a questionnaire if a user defined in the application is logged in). If that condition is satisfied an `activated` questionnaire can be `enacted` by the participant (in the application mentioned as `Guest`). The results after the enactment of the questionnaire, can be `viewed` by the authorized user. All results should be stored within a database to prevent the loss of data collected. The application offers the possibility to `delete` or `export` results. There are two possibilities for exporting the data collected. The results should be downloadable from the internal `application memory` via a specific generated `CSV` file. Further, an upload of questionnaire results to a `server database` should also be available.

4.1.2. Non-Functional Requirements

After specifying the key features using a *task analysis* it is also required to define *non-functional requirements*. These requirements determine aspects affecting the application [Kle11]. For this reason *usability goals* are defined which aim at a common understanding between developer and end-user in respect to the *user interface* and usability. Typically, these goals are divided into different *quality characteristics* (defined in `DIN-ISO-9126` [DIN03]). Table 4.1 lists these characteristics which are considered during the development of the application.

These *usability goals* play an important role during the development of the electronic questionnaire application and provide the basis for an optimal usability. Beside these goals the user should get an application, whereby he can develop questionnaires in less time without high technical requirements. So it is conceivable to offer an special editor (for non-IT experts without knowledge of XML development) in terms of concepts like *drag & drop* and *What You See Is What You Get (WYSIWYG)*. Thereby, a graphical editor with different questionnaire elements (e.g., *textfields, radiobuttons or checkboxes*) are shown to the user. From this *repertoire* the user is able to create his questionnaire.

Classification	**Characteristic**	**Description**
Availability	Availability	The application must be fully functional at any time.
	Robustness	If the device fails (e.g., low battery) no data should be lost.
Task-Orientated	Usefulness	The application should provide support for the user to execute tasks.
	Comfort	The application should provide a comfortable operation to the user.
Orientation	Clarity	The user interfaces should be clearly designed so that the user is able to quickly find his way around.
	Self-Description	The application should be designed for an intuitive handling.
	Expectation	The user should not be surprised unexpectedly when using the application.
	Fault-Tolerance	Incorrect entries should not result in a crash of the application.
Influence	Individualization	It should be possible to customize the application to users needs.

Table 4.1.: Quality Characteristics

Using such an approach it is very comfortable to generate electronic questionnaires even for non-IT experts. With these *functional-* and *non-functional requirements* defined the application is capable of supporting the user in executing questionnaires in a comfortable and user-friendly way. In the next chapter, technical and design requirements for the application concept are defined.

4.1.3. Technical Guidelines

The concept of the application is designed for tablet devices. Originating from the iPad there are many devices with different software versions on the market. It has proven to be difficult to develop an application which is running on every iOS version and device generation. However, this is not possible all the time, since older devices are not able to provide necessary resources or application frameworks. Therefore, one goal of developing an electronic questionnaire application must be to avoid unnecessary resource consumption. This can be achieved using ingenious algorithms which minimize the memory consumption and keep down the running time determined by the *big O notation* (further information in [HR05]).

Through the generic approach to construct questionnaires it is essential that layout and structure automatically adjust for the resolution of the device. In particular, this is important for Apple's *iPad mini*. This specific device has a screen size of `1024 x 768` pixels, compared to the normal iPad with a screen size of `2048 x 1536` pixels. Additionally, these tablets support an *landscape mode* where screen height and width have to be swapped. All these adaptions must be considered by the arrangement of questionnaire elements such as *textfields, radiobuttons or checkboxes*.

Along with the aforementioned factors the problem with different iOS versions is also challenging. In newer versions of the operating system functions are often deprecated and do not work correctly anymore. This often leads to a complete redesign and the implementation of different modules or even the whole application. Therefore, it is essential to use the *APIs* and programming standards as a solid basis for the development of the application.
The same applies to *user interface guidelines* which are determined by the underlying platform and iOS version. The next section will discuss different user interface guidelines, such as the Apple *iOS Human Interface Guidelines*. The developed electronic questionnaire application supports all iPad generations with iOS 5 and higher. However, it is possible that further adjustments are needed for upcoming iOS versions. During this year developer conference (*WWDC 14*), Apple already announced its new iOS 8.

4.1.4. Styleguides (User Interface Design)

In the following, styleguides had been determined which regulate design aspects for the user interface of an application running on a smart mobile device. This section is split in fundamental styleguides which should be generally considered by the design of *user interfaces* and vendor specific styleguides (e.g., Apple's iOS devices). Furthermore, different *look & feel* design approaches for special questionnaire items are illustrated.

Ben Shneiderman's *Eight Golden Rules of Interface Design* are one of the basic guidelines which should be followed by the design of *user interfaces*: [Shn03, Was]

1. **Strive for consistency:** Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.
2. **Enable frequent users to use shortcuts:** As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.
3. **Offer informative feedback:** For every action, there should be some application feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.
4. **Design dialog to yield closure:** Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the end of a group of actions gives the users the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and actions from their minds, and an indication that the way is clear to prepare for the next group of actions.
5. **Offer simple error handling:** As much as possible, the application should be designed so that the user cannot make a serious error. If an error is made, the application should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.
6. **Permit easy reversal of actions:** This feature relieves anxiety, since the user knows that actions can be undone; thus it encourages exploration of unfamiliar actions. The units of reversibility may be a single action, a data entry, or a complete group of actions.
7. **Support internal locus of control:** Experienced users strongly desire the sense that they are in charge of the application and that the application responds to their actions. Design the application to make users the initiators of actions rather than the responders.

8. **Reduce short-term memory load:** The limitation of human information processing in short-term memory requires that displays are kept simple, multiple page displays are consolidated, window-motion frequency is reduced, and sufficient training time is allotted for codes, mnemonics, and sequences of actions.

Beside this general styleguide Jenifer Tidwell provides additional rules. In [Tid10] different patterns, also including patterns for user interfaces for desktop computers as well as for smart mobile devices are described. These patterns show, how elements should look for an optimal use of their function. In addition, they describe different challenges for the mobile design:

1. **Tiny screen sizes:** Mobile devices do not offer much space for information so it is not possible to display sidebars or big images. Therefore, it is necessary to consider carefully which design elements should be displayed.
2. **Touch screens:** It is hard to touch small targets accurately with fingers. So elements must be large enough to hit easily. At a minimum targets should be at least 1cm (0.39 inches) on each side with same space between them. However, this reduces the available space for other contents.
3. **Difficulty of typing text:** It is very unpopular to write text on touch screens. Therefore, typing should be avoided or used very limited.
4. **Challenging physical environments:** People use their mobile devices in all kind of places: outside in the bright sun, in conference rooms, buses or trains. Different lighting conditions and movement contributes to the design of elements so tiny text is hard to read, for example.

For this challenges Jenifer Tidwell describes different approaches to styleguides in a mobile context.

- **Striping down the application to its essence:** It is important to skip the user interface down to its essential elements. Everything not necessary should be omitted. Just focusing on the few tasks that users will need from the current view.

- **Use of the device hardware:** It should be preferred to use features available in the packages of the development environment supported by the hardware of the device to guarantee an optimal compatibility.
- **Linearization of the content:** There is not enough space for all elements like in a desktop application. It is necessary to accept this fact and the content will end up being more vertical instead of horizontal.
- **Optimization of the most common interaction sequences:** After defining tasks which can be performed using the application the developer should focus on making these tasks as easy as possible by the following heuristics:
  - eliminating or reducing of typing.
  - using as few page loads as possible.
  - reducing of scrolling and sideways dragging where it eliminates page loads and typing.
  - reducing the number of taps for reaching an user target.
- **Desktop-First and Mobile-First:** Two opposed approaches for the design of graphical user interfaces. In *Desktop-First* the design is made for desktop screens and is afterwards adjusted to mobile screens. In *Mobile-First* it is exactly the opposite.

These issues mentioned show the limits of mobile user interfaces. Therefore, new concepts of designing *UI*-elements must be developed, whereby Jenifer Tidwell's styleguides for mobile devices provide useful approaches for an optimal and user-friendly interface design.

Turning specifically to iOS platform, Apple has published the *iOS Human Interface Guidelines* which determines design principles for applications, running on the latest iOS operating system [App14]. For this reason, the most important design elements with respect to the electronic questionnaire application are introduced.

Navigation Bar

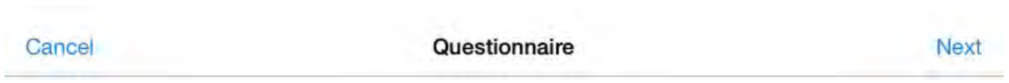


Figure 4.2.: Navigation Bar

A *navigation bar* enables the navigation through different views of the application and is placed at the top of the screen.

If the user navigates to a new level in hierarchy, the navigation bar title should change to the new level's title, if appropriate. Also, a back button with the title of the previous level should appear on the left side of the *navigation bar*.

Picker

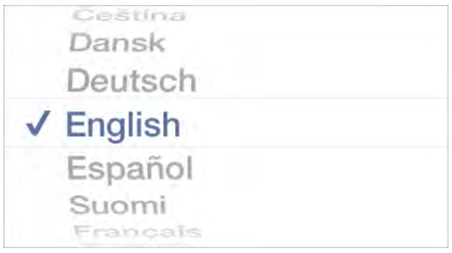


Figure 4.3.: Picker

A *picker* displays a set of values in a scroll wheel from which the user has to pick one. A dark text displays the current value in the middle of the view. A *picker* should only be used if a limited number of values exists. In the electronic questionnaire application a *picker* is used for the language settings.

Image View

An *image view* displays one image or an animated series of images. It should be taken care of that all images in the *image view* have the same size and use the same scaling otherwise images may be rendered incorrectly.

Table View

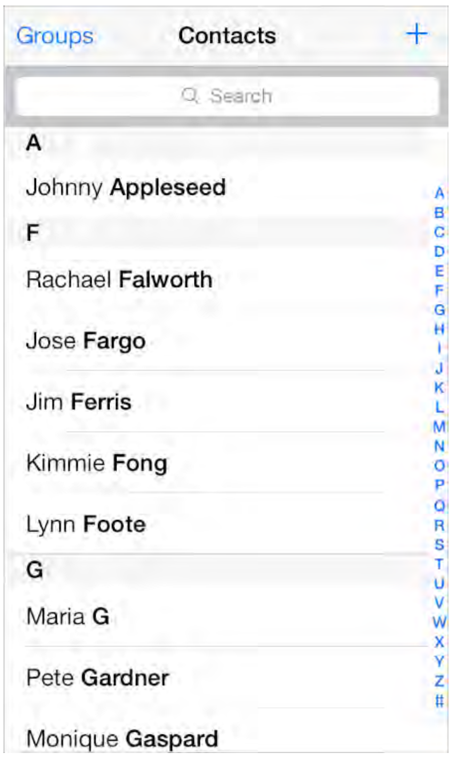


Figure 4.4.: Table View

A *table view* presents data in a scrolling single-column list of multiple rows. Furthermore, listed data can be divided in sections or separated into different groups. A *table view* is useful for displaying a large number of entities (e.g. contacts in address book). In the electronic questionnaire application a *table view* is used to display the results for a questionnaire.

Scroll View

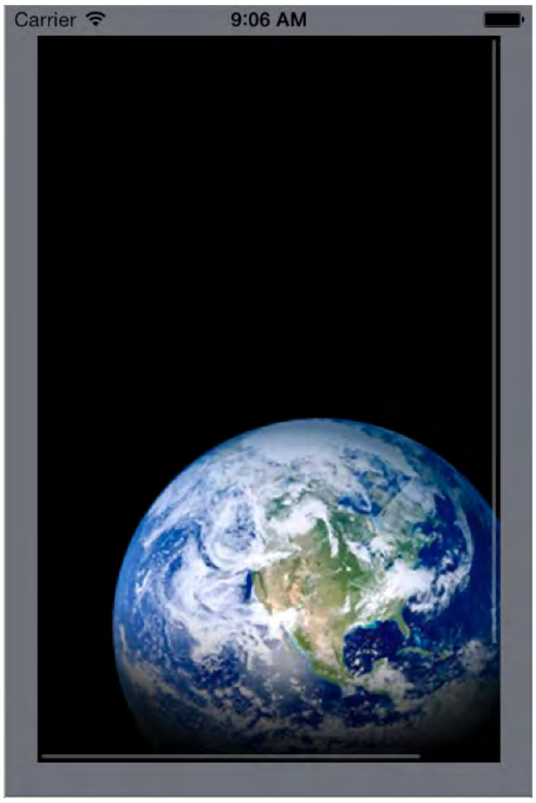
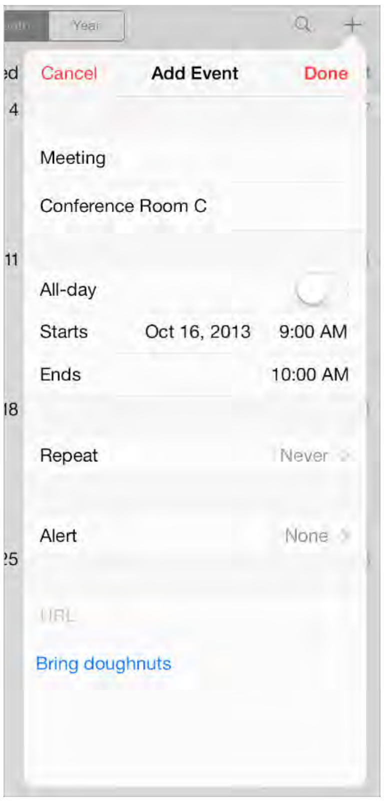


Figure 4.5.: Scroll View



Meeting
Conference Room C
All-day
Starts Oct 16, 2013 9:00 AM
Ends 10:00 AM
Repeat Never
Alert None
URL
Bring doughnuts

Figure 4.6.: Popover

A *scroll view* helps to visualize content which is larger than the scroll view's boundaries. In development, it only one *scroll view* should be used in a *user interface*. This is because people often make large swipe gestures when they scroll, so it can be difficult for them to avoid interacting with a neighboring *scroll view* at the same time. In the electronic questionnaire application a *scroll view* displays one page of a questionnaire. This page contains all question elements used for data collection such as *labels*, *buttons*, *textfields* and *sliders*.

Popover

A *popover* is a transient view that can be revealed when people tap a control element. It always displays an arrow that indicates the point from which it emerged and can contain different *views*. A *popover* should be used to display additional information or a list

of items related to the focused or selected object. Furthermore, the size of a *popover* should not be too big. Note that only one *popover* should be displayed at the same time to avoid confusion. The electronic questionnaire application uses popovers to display the user registration, for example.
The following iOS specific elements are used for the actual representation of the questionnaire running on the smart mobile device.

Label

A *label* displays a static text. In general, this element is used to name or describe parts of the *user interface* or provide short texts (e.g., introductions) to the user. It is useful to use labels in a reasonable font and size to guarantee the readability.

Button

Figure 4.7.: Button decorations: Standard Button, Radiobutton, Checkbox

A *button* calls an specific action. By default (using the latest *iOS Human Interface Guidelines*), a *button* has no border or background. However, it can contain an icon or a text label to describe the action of the element. When creating a *button* it is essential to use a verb or verb phrases to describe the action the *button* performs. In the electronic questionnaire application a *button* is used in different custom decorations: as default button (left), *radiobutton* (middle) and *checkbox* (right).

Textfield

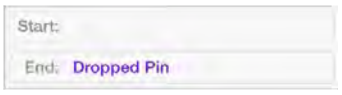


Figure 4.8.: Textfield

A *textfield* accepts a single line of user input with an optional placeholder and automatically displays a keyboard when user selects the element. A *textfield* is the most basic input field to collect data.

Slider

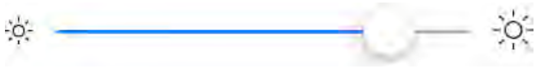
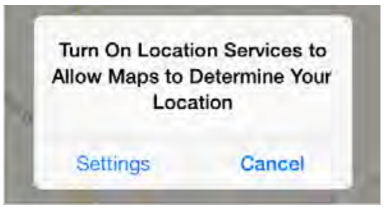


Figure 4.9.: Slider



Turn On Location Services to Allow Maps to Determine Your Location

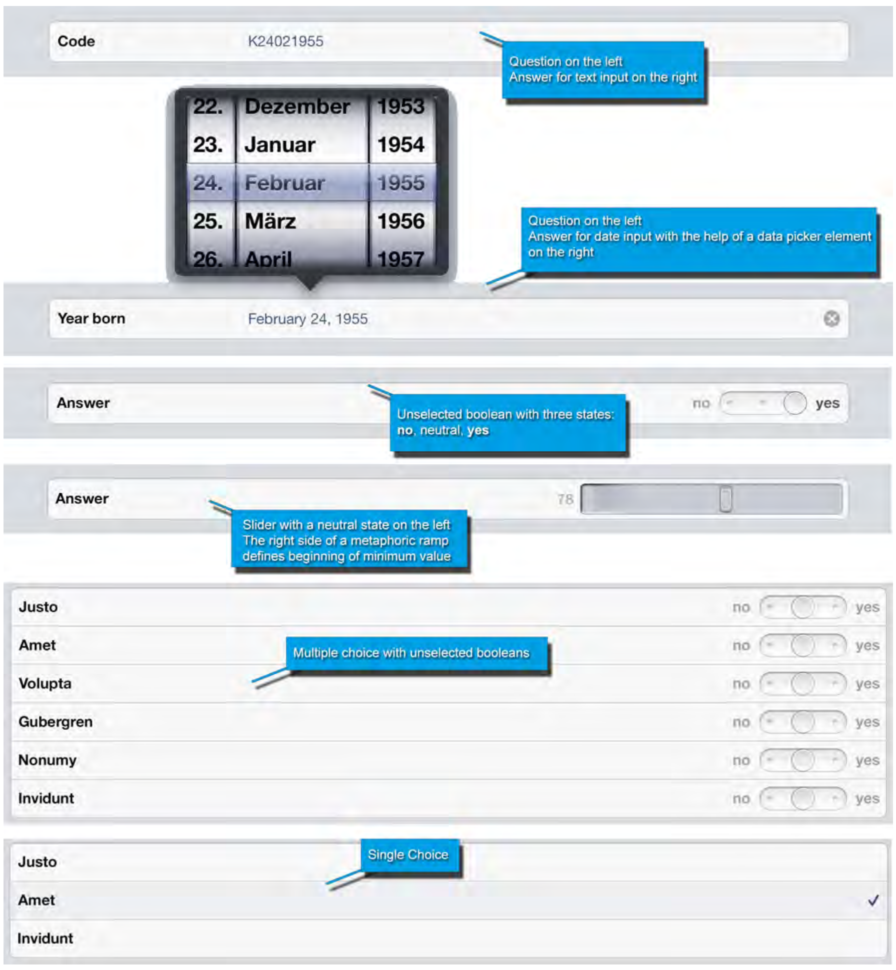
Figure 4.10.: Alert

A *slider* allows users to make adjustments to a value or continuously slide throughout a range of allowed values (shown in figure 4.9 with custom images on the left and the right). It consists of a horizontal track and fills the portion of the track between the minimum value and the maximum value. For example, in the electronic questionnaire application a *slider* is used for answering questions to collect different states (e.g., the participant's state of mind).

Alert

An *alert* (see figure 4.10) provides important information for the user that affects the use of an application. Furthermore, it can contain buttons to interact. For example, in the electronic questionnaire application an *alert* is used if a question marked as required is not answered by the participants.

These iOS specific *UI*-elements were originally styled and developed by Apple. However, it is also possible to customize these elements. Steffen Scherle [Sch14] spent a lot of thoughts on an optimal and user-friendly design for elements for an electronic questionnaire application, based on standard iOS elements. Figure 4.11 illustrates the design of these *UI*-elements.



Code K24021955

Question on the left
Answer for text input on the right

22.	Dezember	1953
23.	Januar	1954
24.	Februar	1955
25.	März	1956
26.	April	1957

Question on the left
Answer for date input with the help of a data picker element on the right

Year born February 24, 1955

Answer

Unselected boolean with three states:
no, neutral, **yes**

Answer 78

Slider with a neutral state on the left
The right side of a metaphoric ramp
defines beginning of minimum value

Multiple choice with unselected booleans

Justo
Amet
Volupta
Gubergren
Nonumy
Invidunt

Single Choice

Justo
Amet ✓
Invidunt

Figure 4.11.: Design Proposals of Questionnaire Elements [Sch14]

Taking these requirements and styleguides listed in this section into account, a *graphical user interface* for visualizing and interacting with questionnaires is created. This is described in detail in section 4.2

4.2. Graphical User Interfaces

In this section the *graphical user interface* for the electronic questionnaire application is illustrated. As figure 4.12 shows the application is divided in three main components beside the two *menu interfaces* which are not highlighted.

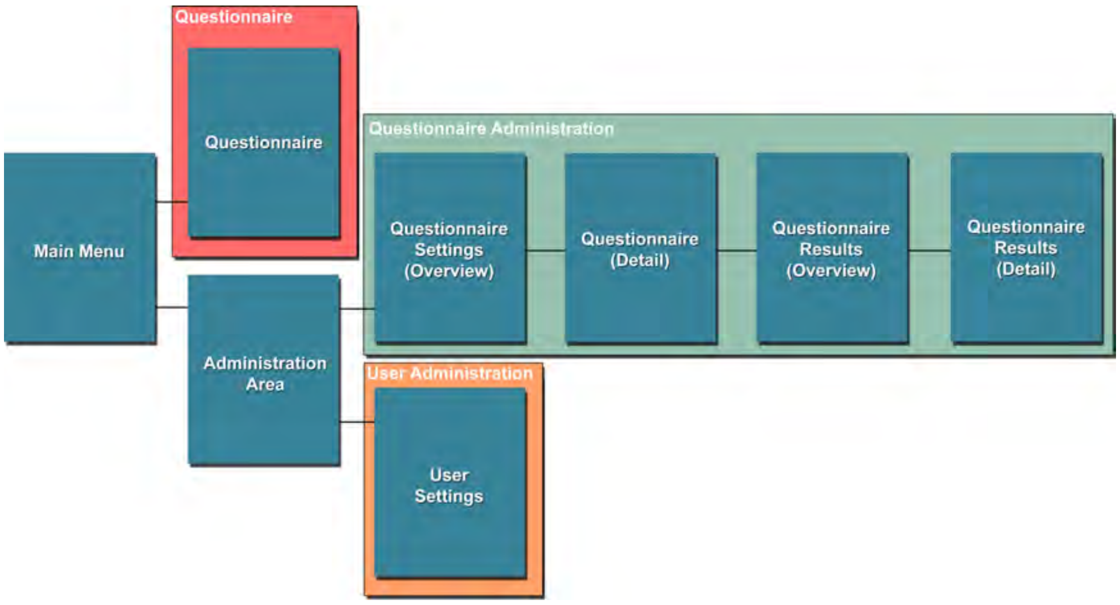

Questionnaire
Questionnaire
Main Menu
Questionnaire Administration
Questionnaire Settings (Overview)
Questionnaire (Detail)
Questionnaire Results (Overview)
Questionnaire Results (Detail)
Administration Area
User Administration
User Settings

Figure 4.12.: User Interface Navigation – Overview

In general, the application consists of the `Questionnaire` component for enacting the actual questionnaires. In the `Questionnaire Administration` different settings for the questionnaires can be made including the possibility to the view results for completed questionnaires. Within the `User Administration` component user settings can be adjusted. It includes the user registration and deletion.
Hereinafter, all *graphical user interfaces* as shown in figure 4.12 are described in more detail (grouped by the introduced components) by using standard user interface elements of Apple's styleguide because these elements are automatically adapted to newer iOS versions. Furthermore, these elements are well established so users know immediately

how to use and interact with them. This *graphical user interfaces* where particularly designed with the *Prototyper* tool [PtJ14].

4.2.1. Main Menus

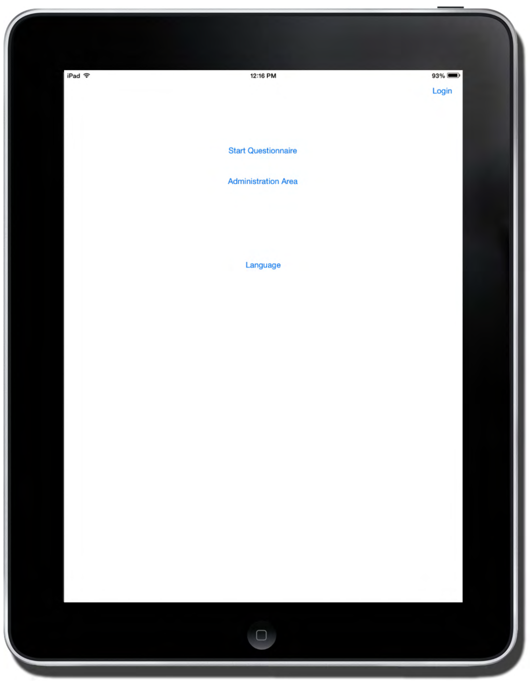


Figure 4.13.: Main Menu – Overview

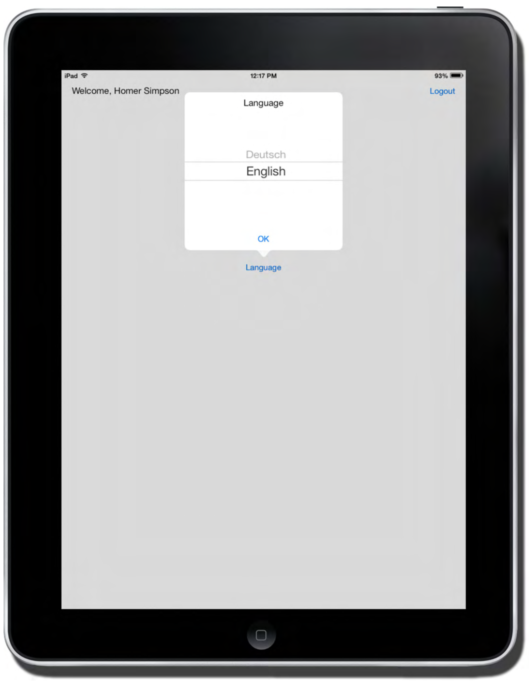


Figure 4.14.: Main Menu – Language Selection

Figure 4.13 illustrates the *main menu* of the application. Starting here, a registered user can *login* to *start a questionnaire*. A page of an actual questionnaire is shown in figure 4.16. Moreover, an access to the *administration area* is offered. Furthermore, the user can change the language of the application to fit his needs.

In figure 4.14 the language selection appears in a *popover* after touching the *Language* button. With the help of a *picker* the application language can be selected by an *up-and-down swipe gesture* by the user. In this case the languages *English* and *German* are currently available, but further languages can easily be added.

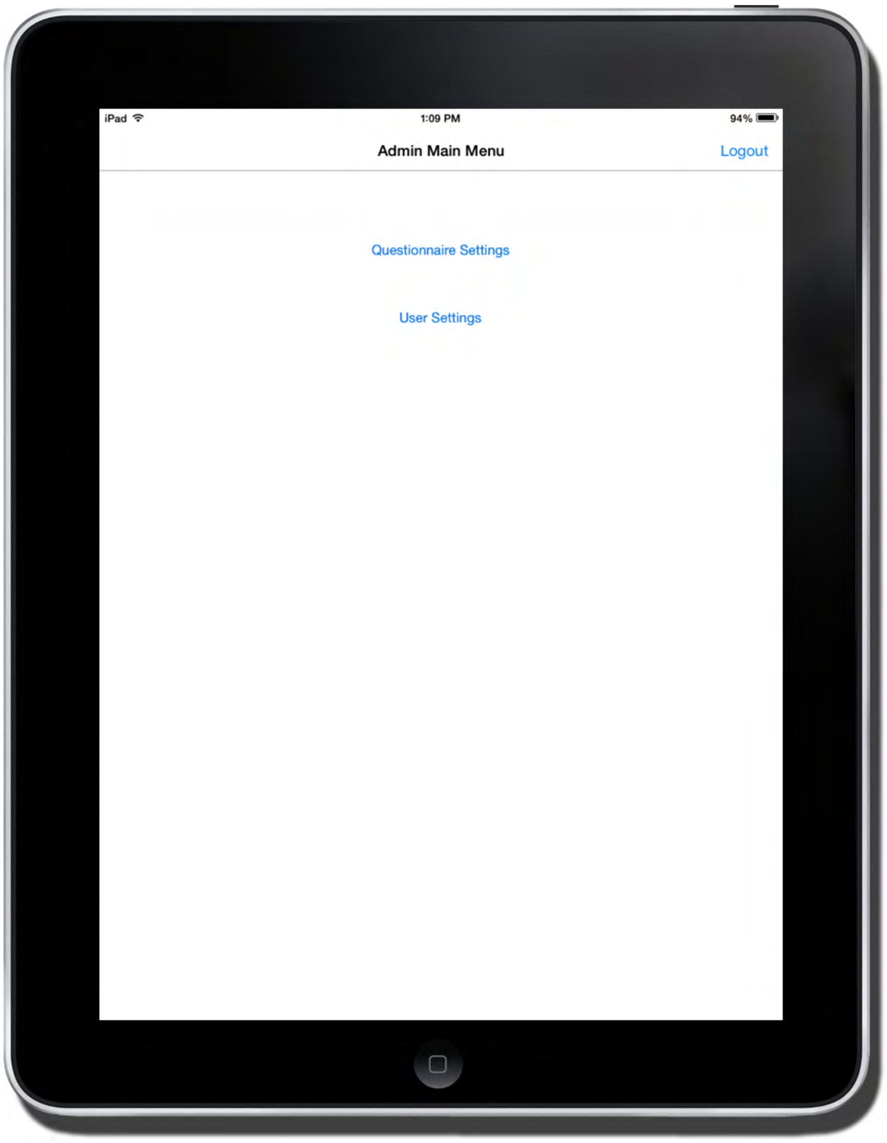
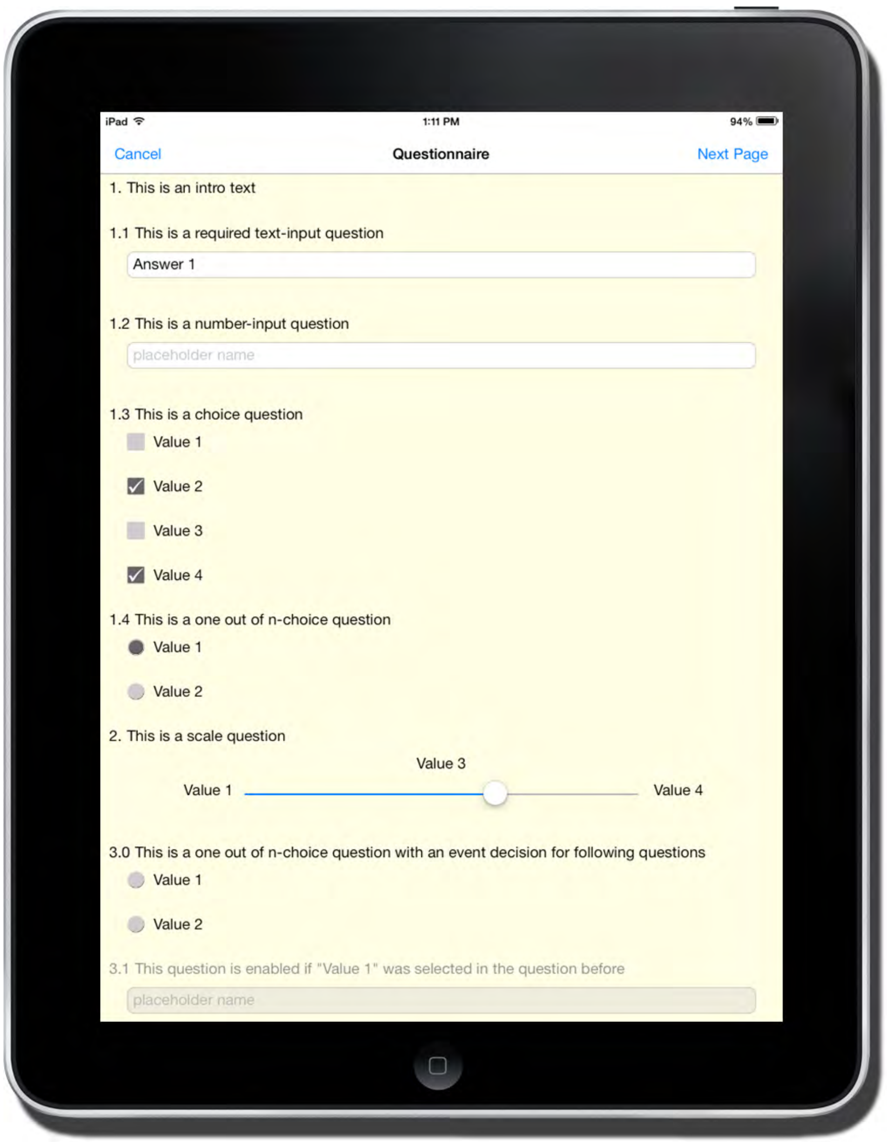


Figure 4.15.: Administration Area – Overview

Figure 4.15 illustrates the *administration area*. This *user interface* is only accessible for users with the *administrator* role. In this password secured area it is possible to adjust the settings of a questionnaire as well as to manage users.
Also a login as *standard user* to an *administration area* is possible. Note, that a normal user is not allowed to do *user settings* and limited *questionnaire settings*. Figure 4.18 shows the *user interface* for a *standard user* after login to the *administration area*.

4.2.2. Questionnaire



1. This is an intro text

1.1 This is a required text-input question

Answer 1

1.2 This is a number-input question

placeholder name

1.3 This is a choice question

- ☐ Value 1
- ☑ Value 2
- ☐ Value 3
- ☑ Value 4

1.4 This is a one out of n-choice question

- ● Value 1
- ○ Value 2

2. This is a scale question

Value 1 — Value 3 — Value 4

3.0 This is a one out of n-choice question with an event decision for following questions

- ○ Value 1
- ○ Value 2

3.1 This question is enabled if "Value 1" was selected in the question before

placeholder name

Figure 4.16.: Questionnaire

In this section the actual electronic questionnaire is illustrated. Figure 4.16 shows the structure and questionnaire design, besides, all elements are created with Apple's iOS standard design as described in section 4.1.4. The electronic questionnaire is divided in various pages which can be shift using the *Next Page* button in the *navigation bar* on top of the page. One page contains a *scroll view*, providing the possibility to scroll down if the height of the screen is insufficient for the content to be displayed. Moreover, every questionnaire element is generically generated from the predefined XML document. It is also possible to cancel a questionnaire, however, all data collected up to the current state is stored in a local database.

4.2.3. Questionnaire Administration

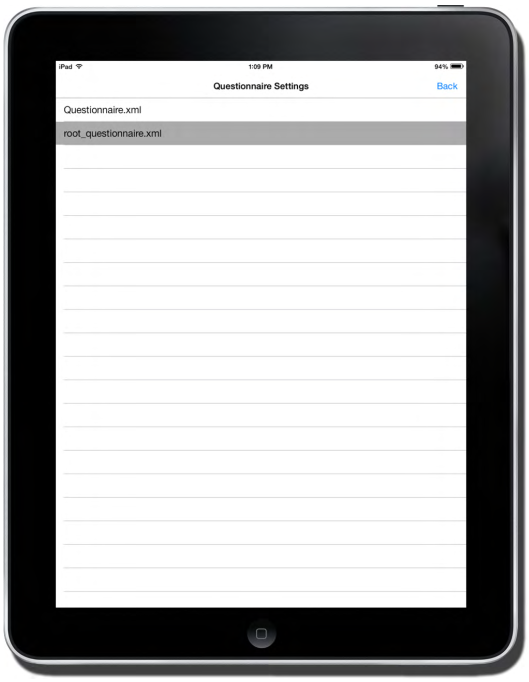


Figure 4.17.: Questionnaire Settings – Overview

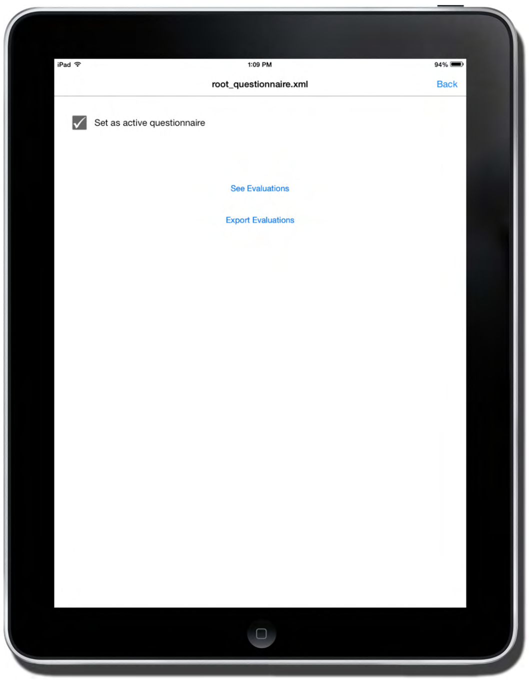
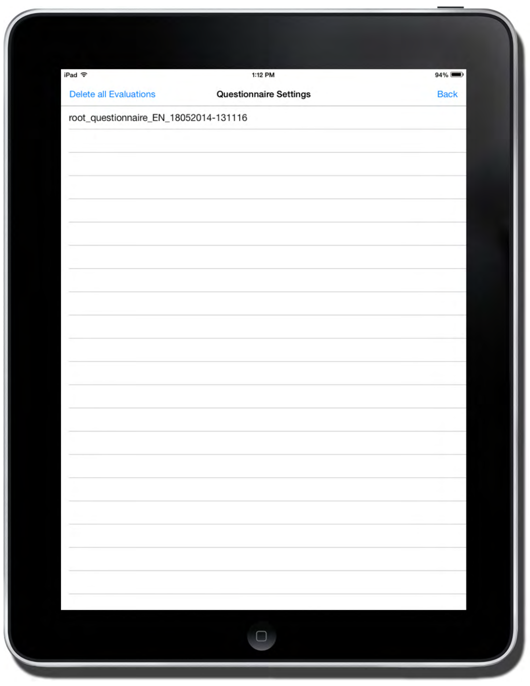


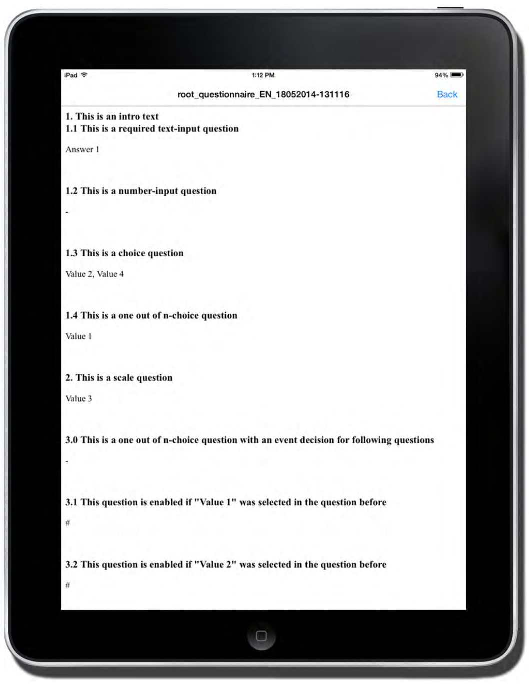
Figure 4.18.: Questionnaire Settings – Detail

The *questionnaire administration area* (see figure 4.17) gives an overview about all XML documents available within the application. For integrating such a questionnaire it is necessary to connect the iPad with a computer and upload the XML document using *iTunes*. After transferring the document it is listed in the administration area. If the user selects one entry from the list he is able to change to the settings of the specific questionnaire (shown in figure 4.18). If a questionnaire is set as active, it can be executed by other users. In this way, various questionnaires can be offered at the same time. Deactivated questionnaires are temporarily not available for an execution, so a flexible use of different questionnaires is guaranteed. Moreover, the results of this questionnaire can be viewed by clicking on the *See Evaluations* button. Of course, the data collected can be exported. It should be kept in mind that a *regular user* can use the same *user interface* like an *administrator* but without the functionality to mark a questionnaire as active.



root_questionnaire_EN_18052014-131116

Figure 4.19.: Questionnaire Results – Overview



root_questionnaire_EN_18052014-131116

1. This is an intro text
1.1 This is a required text-input question

Answer 1

1.2 This is a number-input question

-

1.3 This is a choice question

Value 2, Value 4

1.4 This is a one out of n-choice question

Value 1

2. This is a scale question

Value 3

3.0 This is a one out of n-choice question with an event decision for following questions

-

3.1 This question is enabled if "Value 1" was selected in the question before

#

3.2 This question is enabled if "Value 2" was selected in the question before

#

Figure 4.20.: Questionnaire Results – Detail

After selecting the *See Evaluations* button a list with all completed questionnaires is shown (see figure 4.19). Moreover, the results can be viewed in detail. An enacted questionnaire entry is described as follows: `questionnaireName_language_timestamp[dateAndTime]`, which results in a identifier like `car-maintenance_EN_06102014_142143`.

Figure 4.20 visualizes the details for a completed questionnaire, including the questions and given answers. Thereby an answer marked with an '–' (e.g., as seen for question 1.2) means that the question is not answered while '#' (e.g. for question 3.1) signals that this question can only be answered if the previous question was answered. So it is not possible to answer that question, logically.

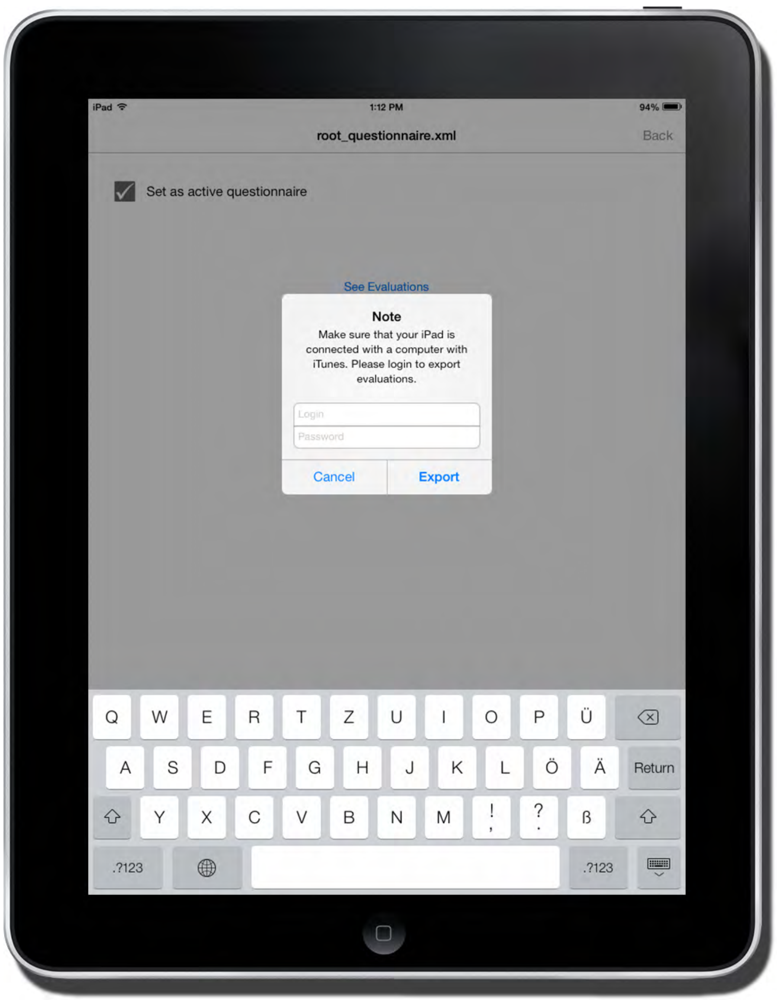

iPad
1:12 PM
94%
root_questionnaire.xml
Back
Set as active questionnaire
See Evaluations
Note
Make sure that your iPad is connected with a computer with iTunes. Please login to export evaluations.
Login
Password
Cancel
Export

Figure 4.21.: Questionnaire Results – Export

Last but not least, the *Questionnaire Administration Area* also provides the functionality to *export results* by connecting the smart mobile device to a computer or by uploading the results to a remote server database.

4.2.4. User Administration

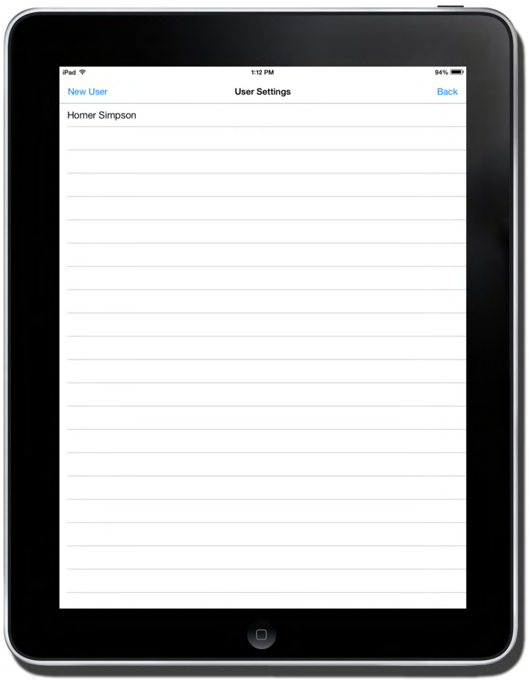


Figure 4.22.: User Settings – Overview

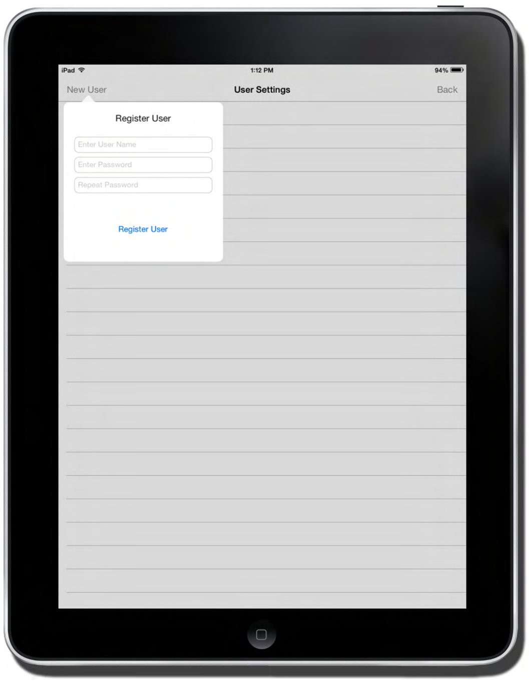


Figure 4.23.: User Settings – User Registration

Figure 4.22 illustrates the *user settings* where an *administrator* can add, edit or delete them. To increase the user-friendly interaction with the application, it is possible to delete an user by a iOS typical wiping gesture to the left on an user entry in the list. This *user interface* is only accessible for *administrators*.
For adding a new user the *administrator* clicks the *New User* button in the *navigation bar* (see figure 4.23). After that a *popover* opens where a new user can be created.

4.2.5. Summary

In this section the main *graphical user interfaces* of the application were introduced. The *user interfaces* can be divided into three different areas. The *User Administration* for a management of the users using the application. These can be registered and deleted in there. In the *questionnaire administration* area different settings concerning questionnaires can be made. It also covers the activation and deactivation of questionnaires the export and viewing of the results for completed questionnaires. Last but not least, the actual *Questionnaire* presents the *graphical user interface* in which a selected questionnaire can be enacted to collect data. In the next section different approaches to implement a generic questionnaire based on an XML document are discussed.

4.3. Approaches to Develop Generic Questionnaires

After the design of *graphical user interfaces* for the application, approaches for the technical development of generic questionnaires are described. Therefore, two different approaches are introduced. These are evaluated against each other and finally one of them is chosen for the implementation of the electronic questionnaire application. Considering both approaches, it is important to use one kind of generic and automatic approach. Moreover, stability in respect to upcoming iOS versions have to be kept in mind.

4.3.1. First Approach: Code Integration for Apple's *Interface Builder*

In this approach a generic questionnaire is implemented with the help of *XIB (XML Interface Builder) & NIB (NeXT Interface Builder)* files. However, these *XIB & NIB* files have to be explained first. That approach is illustrated in figure 4.24 and explained afterwards in detail.

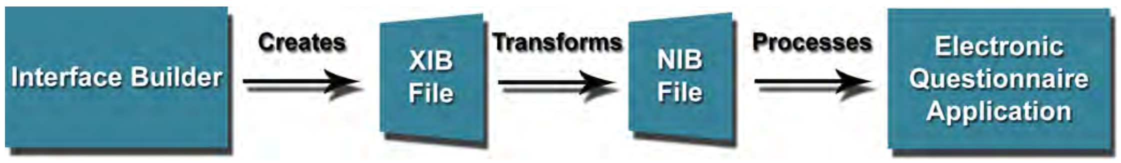


Figure 4.24.: General Process

XIB files contain the views and design elements for the *Interface Builder* including the controls (the functionality of the design element), their layout (the arrangement and size in coordinates), and properties (the characteristics and identification). The *Interface Builder* is a graphical editor for designing *graphical user interfaces* via *drag & drop* for Apple's operating systems such as MAC OS X and iOS. In general, a *XIB* file is an XML based file which describes a *graphical interface*. Moreover, object graphs are stored. That means a *XIB* file, is a hierarchical set of object descriptions. During runtime, all objects defined in the *XIB* file get instantiated, configured and connected as described in the *XIB* file [Mar13].

The approach described here is to define a *XIB* file automatically, including the structure and elements (such as *radiobuttons, checkboxes, sliders or textfields*) of a questionnaire. Afterwards this *XIB* file is loaded into the *Interface Builder*, which then describes a graphical view with design elements for the iOS application, based on the information within the *XIB* file. This structure is then converted in a so called *NIB* file, a binary representation for the runtime environment. Figure 4.25 illustrates a *graphical user interface* with its underlying XML definition in listing 4.1.

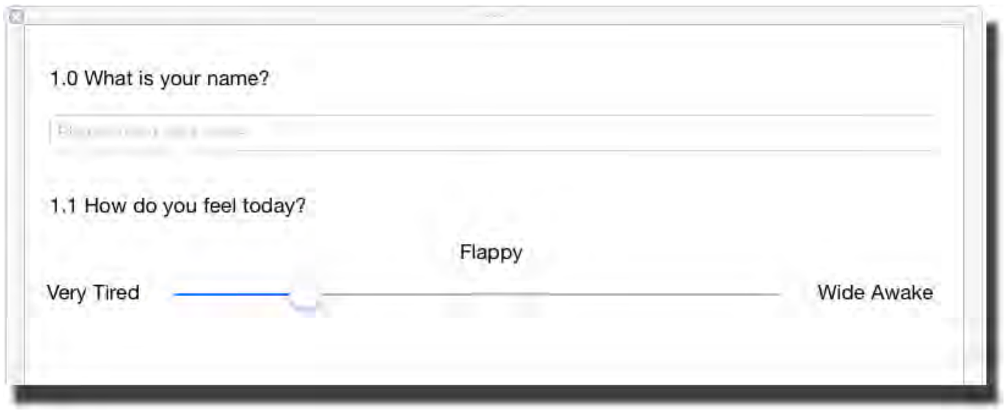


Figure 4.25.: Graphical User Interface in Interface Builder

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.iPad.XIB" version="3.0"
toolsVersion="5053" systemVersion="13C64" targetRuntime="iOS.CocoaTouch.iPad" propertyAccessControl="none"
useAutolayout="YES">
   <dependencies>
      <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="3733"/>
   </dependencies>
   <objects>
      <placeholder placeholderIdentifier="IBFilesOwner" id="-1" userLabel="File's Owner"/>
      <placeholder placeholderIdentifier="IBFirstResponder" id="-2" customClass="UIResponder"/>
      <view contentMode="scaleToFill" id="YWz-SI-hhp">
         <rect key="frame" x="0.0" y="0.0" width="768" height="900"/>
         <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
         <subviews>
            <label opaque="NO" clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
                        horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
                        text="1.0 What is your name?" lineBreakMode="tailTruncation"
                        baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
                        translatesAutoresizingMaskIntoConstraints="NO" id="eEt-HL-i3M">
               <rect key="frame" x="20" y="33" width="728" height="21"/>
               <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
               <fontDescription key="fontDescription" type="system" pointSize="17"/>
               <nil key="highlightedColor"/>
            </label>
            <textField opaque="NO" clipsSubviews="YES" contentMode="scaleToFill" fixedFrame="YES"
                        contentHorizontalAlignment="left" contentVerticalAlignment="center"
                        borderStyle="roundedRect" placeholder="Please insert your name" minimumFontSize="17"
                        translatesAutoresizingMaskIntoConstraints="NO" id="Lyj-HZ-kFa">
               <rect key="frame" x="20" y="73" width="728" height="30"/>
               <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
               <fontDescription key="fontDescription" type="system" pointSize="14"/>
               <textInputTraits key="textInputTraits"/>
            </textField>
            <label opaque="NO" clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
                        horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
                        text="1.1 How do you feel today?" lineBreakMode="tailTruncation"
                        baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
                        translatesAutoresizingMaskIntoConstraints="NO" id="WIz-uM-MVk">
               <rect key="frame" x="20" y="137" width="728" height="21"/>
               <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
               <fontDescription key="fontDescription" type="system" pointSize="17"/>
               <nil key="highlightedColor"/>
            </label>
            <slider opaque="NO" contentMode="scaleToFill" fixedFrame="YES"
                        contentHorizontalAlignment="center" contentVerticalAlignment="center"
                        value="0.20000000000000001" minValue="0.0" maxValue="1"
                        translatesAutoresizingMaskIntoConstraints="NO" id="vrk-p3-x2Y">
               <rect key="frame" x="120" y="204" width="501" height="31"/>
               <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
            </slider>
            <label opaque="NO" clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
                        horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
                        text="Very Tired" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines"
                        adjustsFontSizeToFit="NO" translatesAutoresizingMaskIntoConstraints="NO"
                        id="Usl-Cz-MHd">
               <rect key="frame" x="18" y="208" width="84" height="21"/>
               <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
               <fontDescription key="fontDescription" type="system" pointSize="17"/>
               <nil key="highlightedColor"/>
            </label>
            <label opaque="NO" clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
                        horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
                        text="Wide Awake" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines"
                        adjustsFontSizeToFit="NO" translatesAutoresizingMaskIntoConstraints="NO"
                        id="xcT-Z3-e7a">
               <rect key="frame" x="649" y="208" width="103" height="21"/>
               <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
               <fontDescription key="fontDescription" type="system" pointSize="17"/>
               <nil key="highlightedColor"/>
            </label>
            <label opaque="NO" clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
```

```
                        horizontalHuggingPriority="251" verticalHuggingPriority="251" fixedFrame="YES"
                        text="Flappy" lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines"
                        adjustsFontSizeToFit="NO" translatesAutoresizingMaskIntoConstraints="NO"
                        id="wGA-Ge-zxd">
                    <rect key="frame" x="356" y="175" width="56" height="21"/>
                    <autoresizingMask key="autoresizingMask" flexibleMaxX="YES" flexibleMaxY="YES"/>
                    <fontDescription key="fontDescription" type="system" pointSize="17"/>
                    <nil key="highlightedColor"/>
                </label>
            </subviews>
            <color key="backgroundColor" white="1" alpha="1" colorSpace="custom" customColorSpace="calibratedWhite"/>
        </view>
    </objects>
</document>
```

Listing 4.1: XIB-File for a Graphical User Interface

A big disadvantage of using this approach is that the *XIB* file is very technical and overloaded even for small interfaces. Furthermore, it is possible the markup changes if the iOS version changes. So the structure and attributes can differ and expensive adaptions need to be done. Through the transformation into a binary *NIB* file it cannot be understood what the application is processing during runtime. Considering these factors, another approach for the development of a generic questionnaire application is needed.

4.3.2. Second Approach: *Intelligent Parsing*

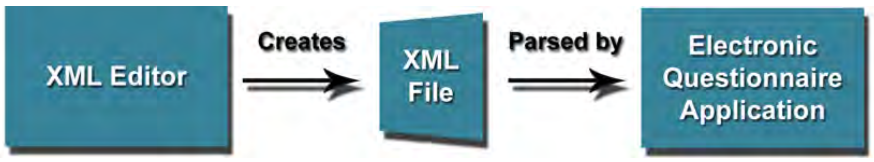


Figure 4.26.: General Process

Another approach which can allow for an automated generation of the user interface is called *Intelligent Parsing*. Using this approach, the disadvantages from the aforementioned approach can be eliminated. In contrast to *XIB & NIB* files the application will be determined by the user and not the other way around. That means an XML document will be defined which is automatically *parsed* by the application (see figure 4.26). Based upon the markup, the correct representation for the *UI* elements is chosen. In this XML document, predefined *tags* provide different questionnaire elements. The code sample in listing 4.1 shows the definition of the same example from figure 4.25.

```
1  <page>
2      <questionGroup>
3          <question>
4              <text>What is your name?</text>
5              <answer type="free-text" mode="text"
6                  placeholder="Please insert your name"/>
7          </question>
8          <question>
9              <text>How do you feel today?</text>
10             <answer type="scale">
11                 <option value="0">
12                     <optionText>Very Tired</optionText>
13                 </option>
14                 <option value="1">
15                     <optionText>Flappy</optionText>
16                 </option>
17                 <option value="2">
18                     <optionText>Wide Awake</optionText>
19                 </option>
20             </answer>
21         </question>
22     </questionGroup>
23 </page>
```

Listing 4.2: XML Document of Figure 4.25

Listing 4.1 is a possible definition of the *graphical user interface*, whereby the lines 3-7 defines the question `"What is your name?"`. Moreover, the `answer`-tag with the `type="free-text"` is added. When parsing the XML document the logic of the application maps the markup to one corresponding element (e.g., an *textfield*-element with an *label* for the text of the question).
The same applies for the rest of the document, however, the parser recognizes the different `answer`-tag (`type="scale"`) resulting in a *slider* with different `options`. The

`questionGroup`-tag ensures the correct numbering of questions. Furthermore, the `page`-tag divides the questionnaire into different pages.

4.3.3. Comparison between the Different Approaches

In the previous sections, two different approaches to create dynamic *user interfaces* were introduced. In general, the introduced approaches are based on the use of an XML file which is basically independent of both approaches. Merely the processing and implementation of the XML file for the creation of the *graphical user interface* differs.
In the process of the first approach a *XIB* file which specifies a *graphical user interface* internally, needs to be implemented. However, there are a few difficulties which are not very comfortable in development. The *XIB* file is very complex with a lot of details such as the *position*, *size* or *unique identification code* for the creation of an element such as a *label* or *textfield*. Furthermore, this approach is not *version-independent*. So if an update for the *Interface Builder* is available, the structure or attributes of the *XIB* file may vary. In principle, this approach has its justification but is not the optimal way for a generic questionnaire application. The more efficient way is to *parse* the document. Thereby, *Intelligent Parsing* ensures a comfortable and user-friendly definition of questionnaires. Moreover, through its version-independent implementation, no further adaptions for newer versions needs to be done if Apple will not change the definition of *UI* elements otherwise the *Parser* needs to be adapted. Table 4.2 and 4.3 list pros and cons of both approaches.
A precise definition of an XML document for a correct interpretation by the application is given in chapter 5.

Pros	• Precise specification of UI-Elements (e.g., position or size) • Utilization of the whole repertoire of all iOS specific UI-elements
Cons	• Very complex • Not version-independent with regard to iOS

Table 4.2.: Code Integration for Apple's Interface Builder

Pros	• User-friendly • Adapted to the application domain • Definition of logical operations • Version-independent with regard to iOS
Cons	• No precise specification • Limited number of UI-Elements

Table 4.3.: Intelligent Parsing

4.4. Summary

In this chapter the *application design and concept* was described. At the beginning, *functional*, *non-functional*, *technical* and *design* requirements were determined when analyzing existing questionnaires. Moreover, different *styleguides* for the representation of the *user interfaces* were considered.
In addition, *graphical user interfaces* for the application were illustrated and explained. Last but not least, two different approaches for a generic implementation of such electronic questionnaires based upon an XML document representation of questionnaires. Chapter 5 describes technical approaches, difficulties and solutions during implementation of the application.

5 Technical Implementation

In this chapter different technical aspects with respect to the implementation of the electronic questionnaire application are described. Thereby, a *schema* for the structure of the XML document specified and explained. Furthermore, the *application architecture* with its components and classes is determined and highlighted with various code examples explaining interesting aspects of the implementation.

5.1. Schema for XML Document

The chapter starts with a *schema* for the XML document. This *schema* determines how the document for a questionnaire is structured and specified for the correct interpretation within the application. If this XML document is valid, it is parsed and translated into graphical elements including its logical sequences. In the following, the syntax of the

XML document is explained in detail. Thereby, figure A.12 as well as figure 5.1 illustrate an overview of the general structure. Furthermore, listing A.2 provides an example to all definitions.

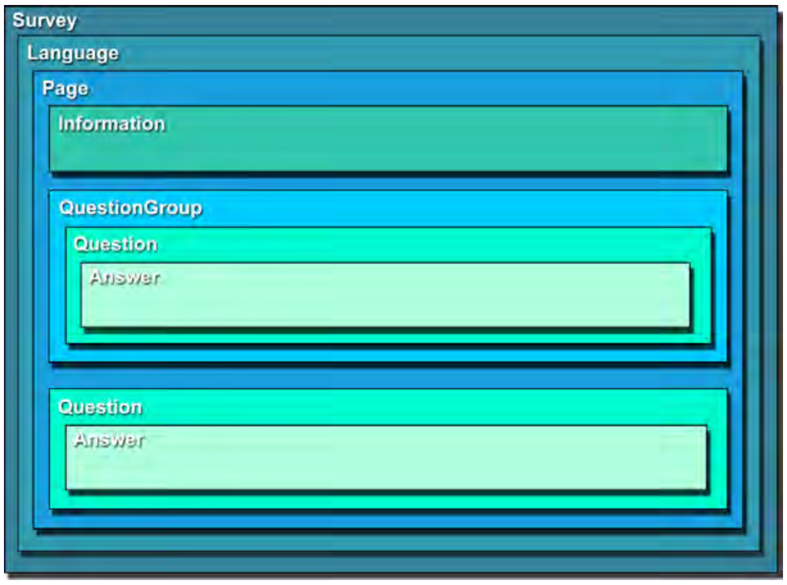


Figure 5.1.: Schema of the XML Document

Survey:
The root element for the XML document is the `survey` element. Within this element the whole questionnaire is defined.

Language:
The `language` element specifies the languages for this specific questionnaire. Everything within this element is assigned for this language. This enables the application to provide a questionnaire in different languages. However, it is also possible to provide a different structure for different languages. Note, that there can be multiple `language` elements.

Syntax:

```
<language lang="LANG" localized-survey-name="NAME">
 ...
</language>
```

`LANG` - language abbreviation (*EN, DE, ES, IT, ...*) – is used for the internal allocation of the language in the application

`NAME` - language name (*English, Deutsch, Español, Italiano, ...*) – is needed for the labeling of languages in the *language selection menu* before a questionnaire is performed

Page:
The `page` element defines a single page within the questionnaire for the selected language. This allows a better structuring and transparent presentation of the questionnaire. With a *Next Page* button which is added automatically, the page can be switched.

Information:
With an `information` element it is possible to define individual texts including images to provide additional information for the participant working with the application. For example, this element is used to layout the introduction text for a questionnaire. The content is defined using HTML which provides a huge variety of styling properties using CSS. It is important to define the HTML/CSS content within a `CDATA`, because HTML as well as XML contains markups which can not be clearly determined by the *parser* which results in possible errors.

Syntax:

```
<information>
      <text><![CDATA[HTML]]></text>
</information>
```

QuestionGroup:
A `QuestionGroup` element logically groups questions and is used for the correct numbering within the questionnaire. Using this element allows for a thematic grouping of questions (e.g., group all questions for a specific topic). However, a `QuestionGroup` element is optional.

Question:
A `Question` element defines a question of a questionnaire and contains a question text. In addition, the `answer`-tag determines the answer-element such as a *textfield, slider* or *checkbox*. In the following, different characteristics of a `Question` element are defined.

Syntax:

```
<question export-name="NAMEID" type="TYPE"
            event-id="EVENT_ID" event-decision="DECISION_ID">
      <text>QUESTION TEXT</text>
      <answer />
</question>
```

`NAMEID` - this ID used to export the data collected to a file. This acts as an identifier for the question.

`TYPE` - defines the type of the question. That type can be `required` or `intro`, if the question (with the `intro` type it is not really a question) is an introduction text, thereby, the included `answer`-tag is ignored by the electronic questionnaire application.

`QUESTION TEXT` - defines the text of the question.

`EVENT_ID` and `DECISION_ID` - the event-id is necessary to identify of the question within the questionnaire. In combination with the decision-id it is possible to link questions logically. That means, all questions with the same `EVENT_ID` have the same behavior if the `DECISION_ID` of an `option`-tag is the same like in the `question` element. The answer with the `option`-tag decides if the question is available or greyed out in the application. Both, `event-id` and `decision-id` are optional. Listing 5.1 illustrates an example of the utilization of logically linked questions. Thereby, the answer of the question `Q1` affects the possibility of answering question `Q2` and question `Q3`.

```
<questionGroup>
   <question export-name="Q1" event-id="0">
      <text>This is a one out of n-choice question with an event decision for following questions</text>
      <answer type="choice" maxAllowed="1">
         <option value="value1" event-decision="0">
            <optionText>Value 1</optionText>
         </option>
         <option value="value2" event-decision="1">
            <optionText>Value 2</optionText>
         </option>
      </answer>
   </question>

   <question export-name="Q2" event-id="0" event-decision="0">
         <text>This question is enabled if "Value 1" was selected in the question before</text>
         <answer type="free-text" mode="text" placeholder="placeholder name"/>
      </question>

```

```
  <question export-name="Q3" event-id="0" event-decision="1">
      <text>This question is enabled if "Value 2" was selected in the question before</text>
      <answer type="free-text" mode="text" placeholder="placeholder name"/>
    </question>
</questionGroup>
```

Listing 5.1: Example of logically linked questions

Answer:

An `answer` element is included in a `question` element and can be specified with different attributes. These attributes define the type and representation of the possible answers for the question.

Syntax: Textfield

```
<answer type="free-text" mode="MODE" placeholder="NAME"/>
```

Using the snippet *Textfield*, the `answer`-tag describes a *field* for a *text* or *number* input.

`MODE` - defines if a textfield (`text`) or numberfield (`number`) is needed which also determines the *keyboard* type.

`NAME` - defines the placeholder for the field.

Syntax: Multiple Choice

```
<answer type="choice" minAllowed="MIN" maxAllowed="MAX">
      <option value="VALUE">
            <optionText>VALUE NAME</optionText>
      </option>
      ...
</answer>
```

In snippet *Multiple Choice*, the `answer`-tag describes a *choice* between different possible *values*. With this syntax the application creates *checkboxes*. Furthermore, an unlimited amount of `option`-tags can be defined.

`MIN` - defines the minimum of choices, a participant can select for this question.

`MAX` - defines the maximum of choices, a participant can select for this question.

`VALUE` - describes the internal value for this option.

`VALUE NAME` - describes the name of the value shown in the *user interface*.

One out of n-Choice

In a *One out of n-Choice* question type, the `answer`-tag describes a *one out of n-choice* of different *answer values*. The syntax is the same like for a *Multiple Choice* question type, however, the `maxAllowed` value must be *1*. Thereby, the application creates *radiobuttons*. Furthermore, an unlimited amount of `option`-tags can be defined.

`MAX` - defines the maximum of choices that must be *1*.

`VALUE` - describes the value.

`VALUE NAME` - describes the name of the value shown in the *user interface*.

Syntax: Scale

```
<answer type="scale">
      <option value="VALUE">
            <optionText>VALUE NAME</optionText>
      </option>
      ...
</answer>
```

In snippet *Scale*, the `answer`-tag describes a *scale* which is interpreted by the application as *slider* element. Furthermore, an unlimited amount of `option`-tags can be defined.

`VALUE` - describes the internal value for this option.

`VALUE NAME` - describes the name of the value shown in the *user interface*.

With this *schema* it is possible to define individual questionnaires which are interpreted by the electronic questionnaire application for interactive use. The *schema* is kept simple for an easy understanding even for non IT-experts. Furthermore, this ensures the maintenance and enables the extensibility through its modularized implementation.

5.2. Application Architecture

This section describes the *technical architecture* of the application. Thereby, an overview of *classes* and *components* is provided, in which different techniques are highlighted. The following figure in 5.2 illustrates the most important *components* of the electronic questionnaire application. These *components* are described in detail in the further course of this thesis.

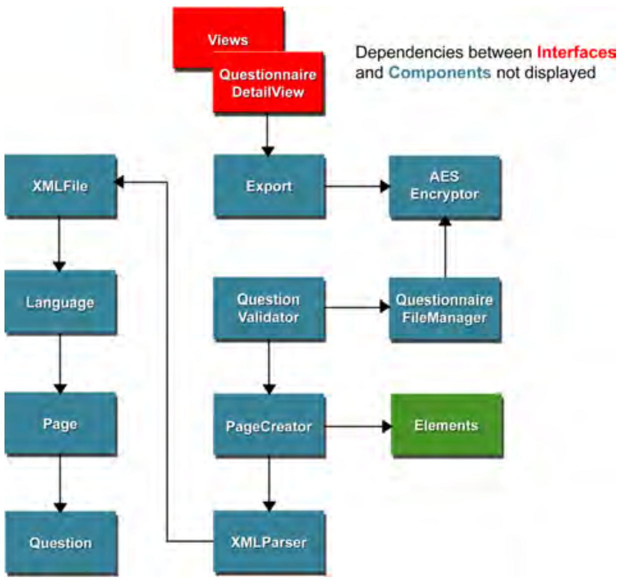


Figure 5.2.: Application Architecture

Question:
A *Question* component is an *instance* of an XML element and consists out of *ID, type, question text* and *answer options*.

Page:
A *Page* component is an *instance* of an XML element and consists out of *Question* components and is defined by an unique *PageID*.

Language:
A *Language* component is an *instance* of an XML element and consists out of *Page* components and is defined by an unique *LanguageID*.

XMLFile:

All previously mentioned components are hierarchically grouped in an *XMLFile* component. As the name suggests, an XML document is mapped into a data structure which can be read by the application. Using this structure it is possible to get access to every *language*, *page*, and *question* component. Moreover, the XML file represents a questionnaire.

XMLParser:

This component *parses* a given XML document. Its task is to filter and process all elements defined by the XML schema. The *parser* operates on a *key & value* principle, whereby, an element found is represented as the *key* whereas the *content* is found between the *opening* and *closing* tag. Each time, the *parser* finds a respective key (i.e., a specific element within the XML document), it maps the content (value) to the previously mentioned components (*question, page* and *language*).
Listing 5.2 shows how a `question` element is identified by the *parser* and converted into an application readable `question component`.

```
1   //get question from XML document
2   if ([tag isEqualToString:@"question"]) {
3      [question setObject:questionID forKey:@"questionID"];
4      [question setObject:questionText forKey:@"text"];
5      [question setObject:currentQuestionType forKey:@"questionType"];
6
7      //save question to an object
8      questionObject.questionID = [question objectForKey:@"questionID"];
9      questionObject.questionType = [question objectForKey:@"questionType"];
10     questionObject.questionText = [question objectForKey:@"text"];
11     questionObject.questionAnswer = [question objectForKey:@"answer"];
12     questionObject.questionOption = [question objectForKey:@"option"];
13
14     //add questionObject to a pageObject
15     [pageObject addQuestion:questionObject];
16  }
```

Listing 5.2: XMLParser for Parsing XML Document

If the key at the current position of the XML document is equal `"question"` (line 2) then the *parser* maps all *attributes* of the tag into a temporary `question` *array* (lines 3-5). Afterwards, the *array* with all *attributes* is mapped into a `question object` (lines 8-12). Finally, the `question object` is added to a `page object` which represents a single *questionnaire page* (line 15).

PageCreator:

The *PageCreator* is one of the main components of the application. It is responsible for the correct and automatic layouting of the generated elements (e.g., *textfields, checkboxes or radiobuttons*) of a *questionnaire page*. During runtime the *PageCreator Component* is successively run through and checks the current *question object* for creating the correct *UI*-element. In this context, an example for creating a *textfield* is shown in listing 5.3

```
1  //question is of type free-text. For this generate a textfield
2  if ([[questionAnswer objectForKey:@"type"] isEqual: @"free-text"]) {
3     UITextField *textField = [[UITextField alloc] initWithFrame:CGRectMake(20, containerHeight+10,
4                                        questionContainer.frame.size.width-40, 30)];
5  
6        [textField setPlaceholder:[questionAnswer objectForKey:@"placeholder"]];
7  
8        //check type of keyboard
9        if ([[questionAnswer objectForKey:@"mode"] isEqual: @"text"]) {
10             textField.keyboardType = UIKeyboardTypeDefault;
11       }
12       if ([[questionAnswer objectForKey:@"mode"] isEqual: @"number"]) {
13             textField.keyboardType = UIKeyboardTypeNumberPad;
14       }
15 
16       [questionContainer addSubview:textField];
17 }
```

Listing 5.3: PageCreator Component for Adding a Textfield

If the currently processed `answer type` is a `free-text` field, the application creates one (line 3) and positions it correctly on the current page. Then a placeholder (as it was defined in the XML document for the question) is set for the *textfield* (line 6). Afterwards, it is checked if the *textfield* needs a `text` or `number` input (lines 9-14). This is necessary, because the *keyboard layout* needs to be adapted to the respective input. Finally, the *textfield* element is add to the current *user interface* of the page (line 16).

Elements:

The *Elements* component contains all questionnaire elements for the *user interface*. These are *textfields, radiobuttons, checkboxs, sliders and labels*, however, it can be expanded with other questionnaire elements.

QuestionValidator:

If the participant of the questionnaire selects the *Next Page button* the current *page* is validated using the *QuestionValidator*. This component checks if all required *questions* are filled in. If not, the *QuestionValidator* throws an *alert* notifying that specific answers are missing. Thereby, every kind of element (e.g., *textfield*, *radiobutton* or *checkbox*) is validated. Listing 5.4 illustrates how a *textfield* is checked if an input occurred by the user.

```
1  //validate textfields
2  if ([element isKindOfClass:[UITextField class]]) {
3     //check if textField contains no text
4     if ([[element text] isEqualToString:@""]) {
5        if ([[[element superview] param3] isEqual:@"required"]) {
6           //generate and show alert
7           UIAlertView *alertMessage = [[UIAlertView alloc]
8                                initWithTitle:[[Settings getLanguage] valueForKey:@"cancelQuestionnaireTitle"]
9                                message:description delegate:self cancelButtonTitle:@"OK" otherButtonTitles:nil, nil];
10          [alertMessage show];
11       }
12    }
13 }
```

Listing 5.4: QuestionValidator Component for Validation

First, it has to be checked if the element type is a *textfield* (line 2). Afterwards, it is checked if the *textfield* is marked as *required* but no data was entered (lines 4-5). If so, an alert with the current *language settings* is created and shown to the user (lines 7-10).

QuestionnaireFileManager:

The *QuestionnaireFileManager* component manages all *results* of the questionnaire. If a *page* in the electronic questionnaire application is turned over the current page is validated by the *QuestionValidator*. If this validation is successful, all *results* are stored encrypted in a database.

AESEncryptor:

This component is responsible for the encryption of the data collected. The used encryption technique is called *AES*, which was already described in section 2.8. In the electronic questionnaire application an encryption is essential due to collect sensitive user data.

Export:

The *Export* component is responsible for creating a *CSV*-file containing all data collected. Moreover, it is responsible for uploading to a web-server. For that it accesses the local database and decrypts the stored *questionnaire results*.

Views:
The electronic questionnaire application contains several *user interfaces* as defined in section 4.2 which are created by *View* components. For example, these are used for visualizing the *Administration Interface* or different *Questionnaire Setting Menus*.

5.3. Summary

This chapter covers different aspects of the *technical implementation* for the electronic questionnaire application. At the beginning an *XML schema* is defined to create valid XML documents. Thereby, the structure and all possible elements recognized by the application are described. Afterwards, the *application architecture* with its core components and classes are discussed. Code examples highlight interesting implementation aspects of these components.
The following chapter 6 shows a practical example which acts as a possible *application scenario* for electronic questionnaires running on a smart mobile device.

6 Application Scenario

The previous chapters illustrated the concept and implementation of an electronic questionnaire application running on a smart mobile device. In this chapter, an application scenario for using such electronic questionnaires from the automotive domain is described. Thereby, the transformation from paper-based questionnaires to an electronic questionnaire is shown. In addition, the enactment with the smart mobile device is illustrated.

6.1. Scenario

In this section a possible application scenario is described. For this, a man named John wants to check his three year old Audi A5 about potential defects to avoid unpleasant surprises at the next *general inspection*. Because John does not know how the *general* *inspection* is structured and what has to be checked, he wants to use a structured and easy to follow *preliminary check*. John had heard from a mobile software application which is offered by his service garage for this purpose.

In this scenario, an electronic questionnaire should be used for the enactment of a *preliminary check* for the *general inspection*. This questionnaire serves as a *checklist* which checks parts of the car for possible defects. A nice side effect may be that the detected defects using this check can be transmitted automatically to a garage nearby. In the further sections the whole process (see Figure 6.1) from defining a *preliminary check* up to the enactment is described. Thereby, relevant questions are defined which were subsequently transformed in an XML document. Afterwards, the XML document will be integrated in the electronic questionnaire application where it can be enacted and evaluated.

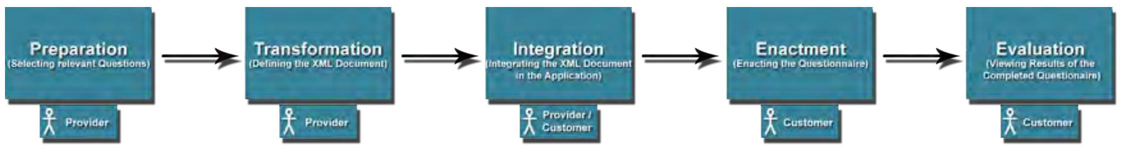
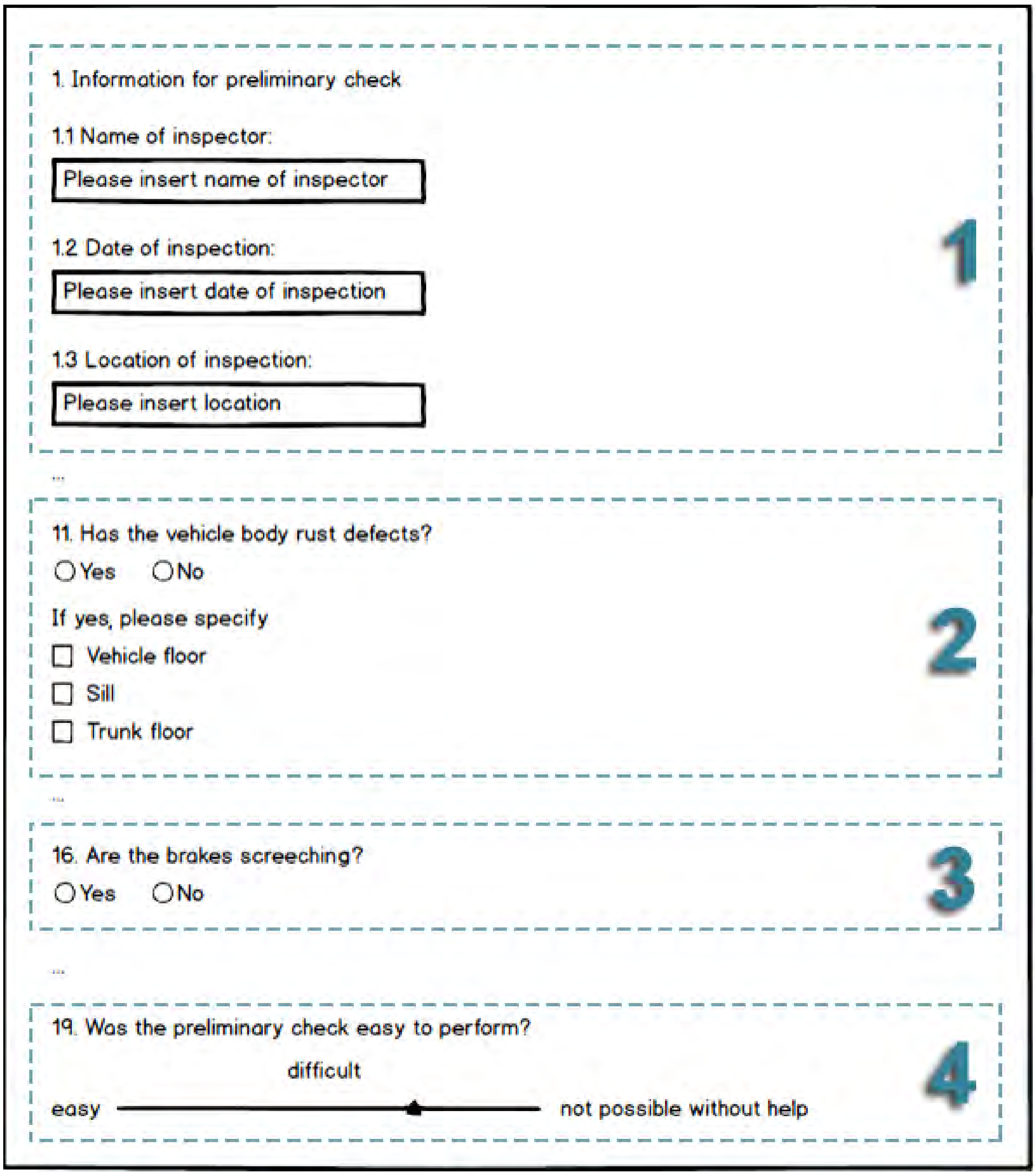

Preparation
(Selecting relevant Questions)
Provider
Transformation
(Defining the XML Document)
Provider
Integration
(Integrating the XML Document in the Application)
Provider / Customer
Enactment
(Enacting the Questionnaire)
Customer
Evaluation
(Viewing Results of the Completed Questionaire)
Customer

Figure 6.1.: General Process

6.2. Preparation and Transformation

Creating a *preliminary check* for the *general inspection* of a car it is necessary to define specific questions. Thereby, questions based on a PDF document for a *preliminary check* by *DEKRA* [Dek14] (see appendix) are used (it is may be possible that the *DEKRA* offers such electronic questionnaire applications). Afterwards, these are linked with questionnaire typically answer possibilities such as *checkboxes* or *textfields*. Figure 6.2 illustrates parts of a possible layout of the questionnaire. The next step is to transform the paper-based questionnaire into a technical representation which can be interpreted by the application running on the smart mobile device. For this, the questionnaire elements will be syntactically described using an XML document. This XML document is checked against a valid schema definition, allowing the application to check the validity of the structure defined (as seen in figure 5.1). Figure 6.2 illustrates a possible *preliminary check* with a subsequently XML definition.


1. Information for preliminary check

1.1 Name of inspector:

Please insert name of inspector

1.2 Date of inspection:

Please insert date of inspection

1.3 Location of inspection:

Please insert location

1

...

11. Has the vehicle body rust defects?

○ Yes ○ No

If yes, please specify

☐ Vehicle floor

☐ Sill

☐ Trunk floor

2

...

16. Are the brakes screeching?

○ Yes ○ No

3

...

19. Was the preliminary check easy to perform?

easy — difficult — not possible without help

4

Figure 6.2.: Conception of a Questionnaire Design [Bal14]

```
1  <questionGroup>
2     <question export-name="general_nformationen" type="intro">
3        <text>Information for preliminary check</text>
4     </question>
5     <question export-name="name" type="required">
6        <text>Name of inspector:</text>
7        <answer type="free-text" mode="text"
8        placeholder="Please insert the name of the inspector"/>
9     </question>
10    <question export-name="date" type="required">
11       <text>Date of inspection:</text>
12       <answer type="free-text" mode="text"
13       placeholder="Please insert date of inspection"/>
14    </question>
15    <question export-name="location" type="required">
16       <text>Location of inspection:</text>
17       <answer type="free-text" mode="text"
18       placeholder="Please insert location"/>
19    </question>
20 </questionGroup>
```

Listing 6.1: Code Example for Figure 6.2 (1)

```
<question export-name="car_body_rust" event-id="9">
   <text>Has the vehicle body rust defects?</text>
   <answer type="choice" maxAllowed="1">
      <option value="car_body_rust_yes"
            event-decision="0">
         <optionText>Yes</optionText>
      </option>
      <option value="car_body_rust_no"
            event-decision="1">
         <optionText>No</optionText>
      </option>
   </answer>
</question>
<question export-name="car_body_rust_option"
                  event-id="9" event-decision="0">
   <text>If yes, please specify</text>
   <answer type="choice" minAllowed="0"
      maxAllowed="4">
      <option value="vehicle_ground">
         <optionText>Vehicle ground</optionText>
      </option>
      <option value="vehicle_door">
         <optionText>Vehicle door</optionText>
      </option>
      <option value="trunk_ground">
         <optionText>Trunk floor</optionText>
      </option>
      <option value="wheel_case">
         <optionText>Wheel case</optionText>
      </option>
   </answer>
</question>
```

Listing 6.2: Code Example for Figure 6.2 (2)

```
<question export-name="break_squeak">
   <text>Are the brakes screeching?</text>
   <answer type="choice" maxAllowed="1">
      <option value="break_squeak_yes">
         <optionText>Yes</optionText>
      </option>
      <option value="break_squeak_no">
         <optionText>No</optionText>
      </option>
   </answer>
</question>
```

Listing 6.3: Code Example for Figure 6.2 (3)

```
<question export-name="feasibility">
   <text>Was the preliminary check easy
      to perfom?</text>
   <answer type="scale">
      <option value="0">
         <optionText>easy</optionText>
      </option>
      <option value="1">
         <optionText>feasible</optionText>
      </option>
      <option value="2">
         <optionText>is ok</optionText>
      </option>
      <option value="3">
         <optionText>difficult</optionText>
      </option>
      <option value="4">
         <optionText>not possible without
                  help</optionText>
      </option>
   </answer>
</question>
```

Listing 6.4: Code Example for Figure 6.2 (4)

In listing 6.1 different answer possibilities are collected using *textfields* (defined in lines 7, 12 and 17). The questions are enclosed using a `questionGroup` tag (lines 1 and 20) and introduced with a general information text (lines 2-4). The attribute `type = "required"` indicates that the question must be answered by the participant and may not be skipped.

In listing 6.2 a *one-out-of-n choice* question with *radiobuttons* is defined (line 3). The next question (line 13) is only available if the preceding question was answered with *Yes*. The dependency between this questions is realized by an `event-id` and the corresponding `event-decision` (lines 1, 5 and 9). If the dependent question fulfills all conditions it is available to the question that implements a multiple selection with the help of *checkboxes*.

In listing 6.3 a simple *one-out-of-n choice* question with *radiobuttons* is defined (line 3). Note, that there are no further dependencies to answers given to other questions.

In listing 6.4 a *scale* is defined using a *slider* element. Thereby, possible answers are listed using option tags. Note, that there exist a human-readable *text* as well as a *numerical value* for the subsequent evaluation (lines 5-20).

6.3. Integration, Enactment and Evaluation

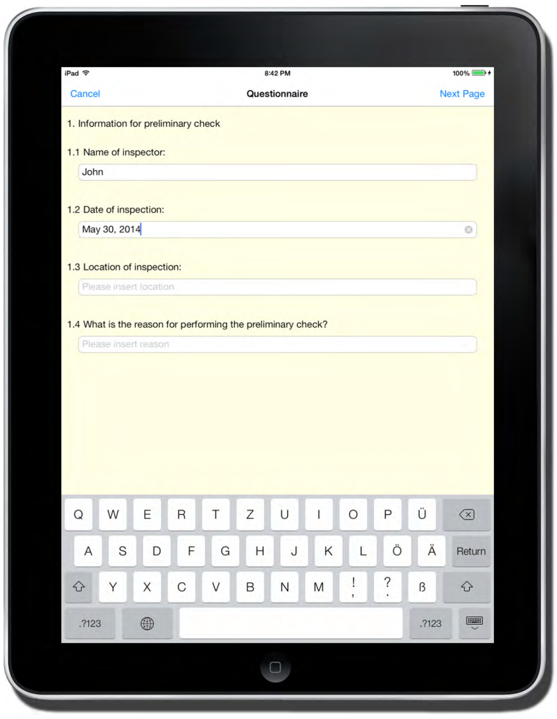
In the following, the general process is described which includes the integration, enactment and the evaluation of collected data of the electronic questionnaire.

6.3.1. Uploading a Questionnaire and Activating it

After specifying the XML document of the *preliminary check* for the *general investigation* of a car, the document must be uploaded to the smart mobile device which runs the electronic questionnaire application. This upload is done by connecting the device to a computer with the *iTunes* software on it. After the upload has finished, the questionnaire has to be *activated*. Therefore, a login as *administrator* is necessary to *select* and *activate* the uploaded questionnaire.

6.3.2. Enactment of the Questionnaire

After the questionnaire was *activated* it is ready for use. In the presented application scenario, the XML document for the *preliminary check* implements two languages (english and german). When the user starts the questionnaire, he can select the language which suits him best. Afterwards, the XML document is transformed into a *graphical user interface* with iOS specific elements by the application using the approach presented in chapter 4. The next figures illustrate excerpts of the *preliminary check*.



1. Information for preliminary check

1.1 Name of inspector:

John

1.2 Date of inspection:

May 30, 2014

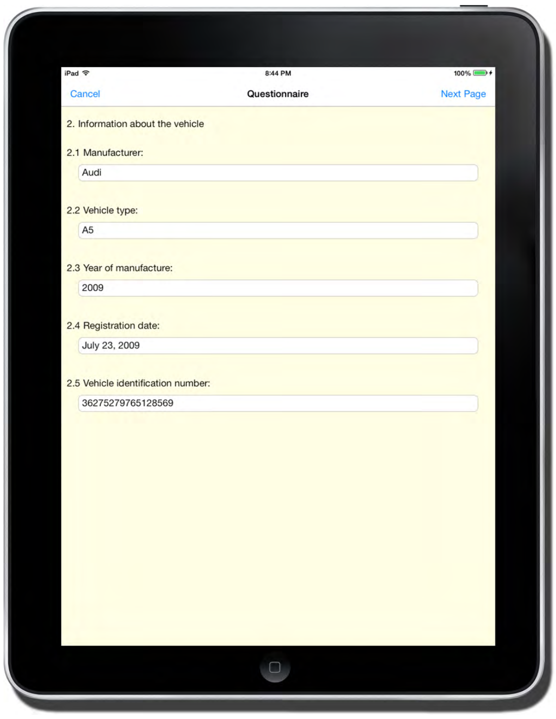
1.3 Location of inspection:

Please insert location

1.4 What is the reason for performing the preliminary check?

Please insert reason

Figure 6.3.: General Information



2. Information about the vehicle

2.1 Manufacturer:

Audi

2.2 Vehicle type:

A5

2.3 Year of manufacture:

2009

2.4 Registration date:

July 23, 2009

2.5 Vehicle identification number:

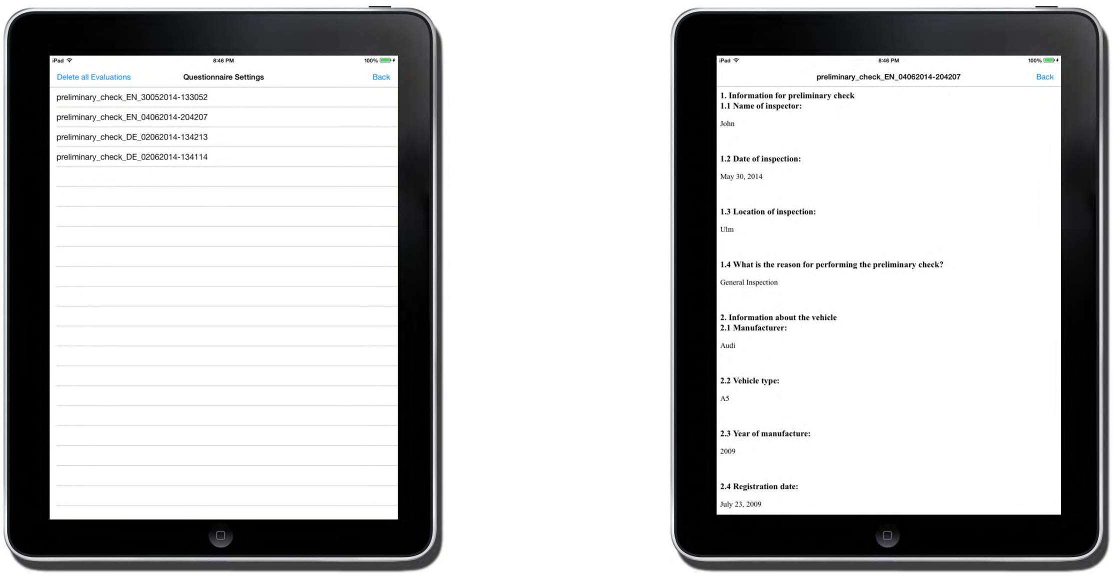
36275279765128569

Figure 6.4.: Information about the Vehicle

Figure 6.3 and 6.4 illustrate two pages with general information needed for the *preliminary check* and the *vehicle*. Using the *Next Page* button in the header of the page, the next page of the questionnaire is selected and displayed.

6.3.3. Results of a Completed Questionnaire

After a *preliminary check* was enacted the *results* for a specific questionnaire can be reviewed by a mechanic. Using an administrator account, the electronic questionnaire application offers the possibility to see all *results* for a completed questionnaire (see figure 6.5).



Delete all Evaluations | **Questionnaire Settings** | Back

preliminary_check_EN_30052014-133052

preliminary_check_EN_04062014-204207

preliminary_check_DE_02062014-134213

preliminary_check_DE_02062014-134114

preliminary_check_EN_04062014-204207 | Back

1. Information for preliminary check
1.1 Name of inspector:

John

1.2 Date of inspection:

May 30, 2014

1.3 Location of inspection:

Ulm

1.4 What is the reason for performing the preliminary check?

General Inspection

2. Information about the vehicle
2.1 Manufacturer:

Audi

2.2 Vehicle type:

A5

2.3 Year of manufacture:

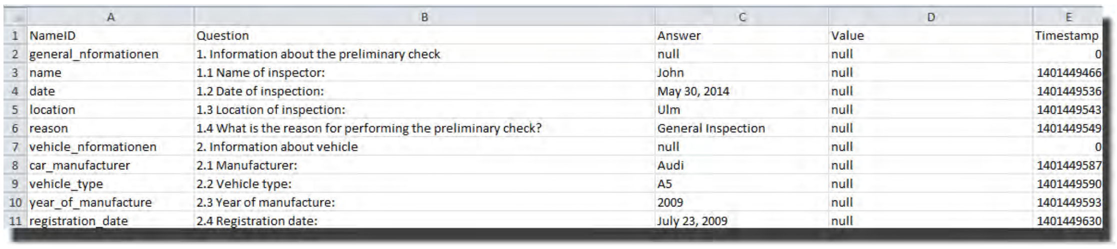
2009

2.4 Registration date:

July 23, 2009

Figure 6.5.: Selecting a specific Questionnaire and displaying the Results

Furthermore, by using such an administrator account, it is possible to *export* these *results* to a *CSV* file (see figure 6.6) or upload them to a server for further analysis. The *CSV* file offers all data collected for a completed questionnaire. Thereby, the `NameID` is the internal identification for the question. `Question` is the label of the question in the questionnaire. The `Answer` column shows the answer entered by the participant. The `Value` column is necessary for multiple choice or single choice answers. Thereby, the selected value is stored. The `Timestamp` column stores the current time as soon as a question is answered by the participant. In this way, it is possible to calculate the answering time of questions. If the timestamp is equal to zero it is just an information text and no question in the proper meaning.



	A	B	C	D	E
1	NameID	Question	Answer	Value	Timestamp
2	general_nformationen	1. Information about the preliminary check	null	null	0
3	name	1.1 Name of inspector:	John	null	1401449466
4	date	1.2 Date of inspection:	May 30, 2014	null	1401449536
5	location	1.3 Location of inspection:	Ulm	null	1401449543
6	reason	1.4 What is the reason for performing the preliminary check?	General Inspection	null	1401449549
7	vehicle_nformationen	2. Information about vehicle	null	null	0
8	car_manufacturer	2.1 Manufacturer:	Audi	null	1401449587
9	vehicle_type	2.2 Vehicle type:	A5	null	1401449590
10	year_of_manufacture	2.3 Year of manufacture:	2009	null	1401449593
11	registration_date	2.4 Registration date:	July 23, 2009	null	1401449630

Figure 6.6.: CSV File for the Results of the *Preliminary Check*

6.4. Summary

In this chapter an example of the use of an electronic questionnaire application is presented. This scenario describes the enactment of a questionnaire for a *preliminary check* for the *general inspection* of a car. In the process of using such electronic questionnaires all necessary steps are explained. This starts by transforming a paper-based questionnaire concept to an electronic questionnaire, uploading it to the smart mobile device and enacting the questionnaire. Further, it is explained, how the data collected can be reviewed. In addition, the application provides the functionality to *export* the results to a server or transform them locally using a *CSV* file.

7 Discussion

This chapter recalls the results achieved while developing an electronic questionnaire application. In addition, these results are compared with the *problem statement* in chapter 1. Thereby, it is checked if the results and scientific outcome solve the requirements for a significant improvement of the process of creating and enacting such electronic questionnaires. Furthermore, interesting aspects which were introduced in this thesis are treated.

The *problem statement* described the need to improve the creation of electronic questionnaires in order to save time and costs. Moreover, it should allow an easier and more comfortable development as well as for a much faster adaption to changes and new requirements. For this reason, the approach presented here provides a generic schema for a fast development of electronic questionnaires. Thereby, two generic approaches were considered and compared. In the chosen *Intelligent Parsing* approach a questionnaire is represented as an easy understandable and human-readable XML document. Using

this approach, even non-programmers are able to define an electronic questionnaire. Thereby, a lot of time, can be saved resulting in less costs. This is due to that it is no longer necessary that all user interfaces of an electronic questionnaire must be created individually. Only a structure of the questionnaire that is easy to realize has to be defined in a textual and human-readable form. Because of the definition of questionnaires as an XML document, every single questionnaire can be easily integrated via an upload to extend the application.
Another benefit is the user support of the enactment of questionnaires, so the application can react on user inputs and is able to do logical conclusions. This is particularly expressed if a question is depending from already given answers. Furthermore, the application reminds the participant if not all questions are answered.
Despite the simplification of the creation of electronic questionnaires defined as XML document the question raises if there are possibilities for further improvements. Thereby, it could be conceivable to implement a drag & drop *questionnaire editor*. This allows a simple and user-friendly designing of electronic questionnaires with an automatically generation of an XML document.

In this thesis, a concept for a native development on the iOS platform is described. The use of a cross-platform approach like *PhoneGap* basing on web technologies was deliberately avoided. There are several reasons for this decision. To ensure equal conditions, studies using unified tablet devices (e.g., the iPad) for collecting data by enacting electronic questionnaires. For this reason, it will make sense to offer an application, fully supported by the device which is not necessarily given by a cross-platform approach. Moreover, a native development allows a platform specific user interface design, in contrast to the other approach. There, the user has to get accustomed to the platform atypical design which could increase the error rate of data collected.

During the development of the electronic questionnaire application, similar electronic questionnaire applications were considered. For example, *Snap* [sna14] offers beside standard questionnaire elements (e.g., *textfields*, *checkboxes* or *sliders*) interactive media elements like *videos* and *selectable pictures*. Furthermore, this allows for additional questions (e.g., *"which emotions are triggered by this movie?"*). In contrast, other

products like *SurveyPocket* [sur] are characterized by a huge variety of more than 50 question types.
Two studies consider the use of paper-based vs. electronic questionnaires [DGF+95, BO+08]. They describe that participants highly prefer electronic questionnaires because of its interactive and easy usability. This results in more consistent answers, less missing values and a higher rate of completed questionnaires.

The architecture of the application is structured in interchangeable components which enables to enrich the application in features and functionality. For example, this is helpful if the encryption component should be switched from the currently implemented *Advanced Encryption Standard* to other algorithms like *RSA*.

The introduced generic approach for the structure of an electronic questionnaire, result in an application which makes it possible to develop and specify questionnaires very easily and quickly. The questionnaire is defined on a higher level of abstraction (an XML document following a specific schema) which even allows non-programmers to develop and maintain their own questionnaires.

8 Conclusion & Outlook

This chapter summarizes the introduced concepts and approaches for the development of an electronic questionnaire application. Furthermore, an outlook for additional features and other ways to extend the application developed is given.

8.1. Conclusion

This master's thesis deals with the concept and development of an electronic questionnaire application running on smart mobile devices. Usually, questionnaires are developed individually and often in a paper-based form. However, this approach is very time consuming and not very comfortable. Using such questionnaires the data collected has to be transferred (often manually) to digital worksheets to carry out evaluations. For this reason, a generic solution should be developed to solve the mentioned problems.

Therefore, two approaches are considered and compared with each other. Finally, a XML schema is defined, which describes a questionnaire syntactically. The application for such an electronic questionnaire then parses a document, based upon this provided schema and generates interface elements (e.g., textfields, checkboxes, radiobuttons or sliders) for the user interaction. This approach provides the possibility to generate electronic questionnaires in a much shorter time without hard coded user interfaces. After filling in such questionnaires, the results are stored encrypted on the device. These results can then be exported using a CSV file or uploaded to a server for a subsequent analysis.

This thesis describes the concept of the implemented application using different software development models. In the beginning, related work and comparable projects are investigated. This is important to get a deeper understanding and knowledge about questionnaire design and mobile software development in general. Further, functional and non-functional requirements are defined describing actions available in the mobile application. The non-functional requirements determine technical guidelines and styleguides for the user interface which guarantee a qualitative and user-friendly application. Afterwards, a graphical user interface is designed and illustrated.
In addition, a general architecture including its components is presented. Important components like the data export and an encryption technology are introduced in detail.

Last but not least, an application scenario describes the use of the electronic questionnaire application. This section provides a use case explaining the development of an electronic questionnaire using the concept developed. Furthermore, the application provides the functionality to review the results of enacted questionnaires. All these aspects result in a more comfortable and less time consuming development of questionnaires.

8.2. Outlook

The presented approach and implemented electronic questionnaire application offers a lot of possibilities. Nevertheless, further improvements and optimizations are possible.

In the current concept it is necessary to have knowledge about the structure of the XML schema. Thereby, it would be conceivable to offer an application which allows the generation of such a questionnaire using a repository of elements (e.g., textfields, checkboxes, radiobuttons or sliders). In this way, a simplification for the user is made by a drag & drop technique to design an individual questionnaire. In the background, the XML document is automatically generated on the basis of the graphical design of the questionnaire on the user interface.
Currently, all automated created questionnaire elements are arranged one below each other in the user interface. Perhaps this approach is not always the best so a possible optimization could be to implement a logical arrangement of elements (e.g. a *yes* and *no* decision placed side by side) for an optimal and space-saving look.
Furthermore, it may be desired to offer a *real-time* evaluation depending on the answers given by the participant. One example of this could be the preliminary check for a general inspection described in chapter 6. In this way, the results are evaluated by the application and transmitted immediately to a garage. As it can be seen a great potential and versatile use for electronic questionnaires is possible.

A
Appendix

1	Introduce a walkthrough, security audit review or a formal security review in every phase of the software life cycle development.
2	Establish security metrics during the software life cycle and a trace matrix for security requirements.
3	Determine stakeholders, and elicit and specify associated security requirements for each stakeholder
4	Determine context and potential usage of software product along with the operating environment and specify requisite security requirements.
5	Make available to programmers, developers, reviewers and test teams the vulnerabilities and potential exposures associated with programming languages and operating systems before the architectural design phase.
6	Set up security parameters for access to services such as ftp service where anonymous ftp is allowed but with write only and no read or list to the incoming directory and read only for outgoing directory
7	Check for sources of software security risks such as inconsistencies in requirements and in design, reusable programs and other shrink-wrap software. Use of requirements tools, modeling tools, etc. can aid in this area.
8	Avoid the use of unsafe routines such as sprintf(), strcpy/cat(), gets and fgets in coding.
9	Check the security of any middleware in the program.
10	Check for architectural-specific vulnerabilities and how data flows through the code.
11	Check for implementation-specific vulnerabilities such as Race Conditions, randomness problems and buffer overflows.
12	DO NOT allow programmer backdoors or unauthorized access paths that bypass security mechanisms.
13	Avoid storing secrets like passwords in the code or use weak encryption schemes
14	Identify all points in the source code where the program takes input from users.
15	Identify all points in the source code where the program takes input from another program or un-trusted source.
16	Investigate all sources from which input can enter the program such as GUI, network reads, etc.
17	Check API (Application Program Interfaces) calls to security modules or interfaces.
18	Investigate secure connections. Verify that they actually are secure and connect as indicated to the systems to which they are intended to connect.
19	Investigate software built-in extensible features.
20	Review software complexity and look for alternatives to reduce the complexity.

21	Investigate the security of the data when passed from application servers to databases.
22	Avoid default or other improper configurations that may open the door to attackers.
23	Default to “highest security” needed, and require validation and approval for deviations.
24	Establish tools to be used for various stages of the life cycle that will be used for assessing security.
25	Perform security testing for unit and system integration.
26	Potentially, establish a security risk rating criteria and document the rating of the software product within the organization. Using a risk assessment tool can benefit this area.

Table A.1.: Software Security Checklist (SSC) [GWSB03] in Chapter 3.2.2

Context Questions for Use Cases (Chapter 4.1.1):

Name of Use Case	Create User
Description	User can be created to assign them with questionnaires. Necessary to get an access to results of assigned questionnaires only.
Precondition	An administrator account exists.
Postcondition	User is created in application.
Regular Expiration	1. Administrator switches in user management area. 2. Entering of personal data for creating user account. 3. User will be stored in application database.
Exceptional Cases	Entered data are not complete.

Table A.2.: Context Question for Use Case *Create User*

Name of Use Case	Delete User
Description	User account can be delete from application by administrator.
Precondition	User to be deleted is existing.
Postcondition	User is deleted in application.
Regular Expiration	1. Administrator switches in user management area. 2. Administrator select user to be deleted. 3. User will be deleted from application database.
Exceptional Cases	-

Table A.3.: Context Question for Use Case *Delete User*

Name of Use Case	Activate Questionnaire
Description	Administrator is able to activate questionnaires which can be selected for enacting.
Precondition	Questionnaire which wants to be activated exists.
Postcondition	Questionnaire can be selected for enacting.
Regular Expiration	1. Administrator switches in questionnaire management area. 2. Selection of questionnaire which should be activated.
Exceptional Cases	Selected questionnaire is activated yet.

Table A.4.: Context Question for Use Case *Activate Questionnaire*

Name of Use Case	Deactivate Questionnaire
Description	Administrator is able to deactivate questionnaires. If a questionnaire is deactivated it is not available for enacting.
Precondition	Questionnaire which wants to be deactivated is existing and currently activated.
Postcondition	Questionnaire is not available for enacting in application anymore.
Regular Expiration	1. Administrator switches in questionnaire management area. 2. Selection of questionnaire which should be deactivated.
Exceptional Cases	Selected questionnaire is activated yet.

Table A.5.: Context Question for Use Case *Deactivate Questionnaire*

Name of Use Case	Login
Description	By enacting a questionnaire or getting access to questionnaire management area an user must be logged in.
Precondition	User exists in application.
Postcondition	User is logged into the application and can enact additional features.
Regular Expiration	1. User touches a login button 2. User logs in with username and password.
Exceptional Cases	Entering of wrong access data.

Table A.6.: Context Question for Use Case *Login*

Name of Use Case	Integrate Questionnaire
Description	For processing of questionnaires in application, first a XML document has to been uploaded in the application memory.
Precondition	Device is connected with a computer for uploading XML document.
Postcondition	XML document which represents the syntax of an questionnaire is accessible for the application and able for processing.
Regular Expiration	1. User connects device to a computer. 2. With a tool (e.g. iTunes) user uploads XML document.
Exceptional Cases	Uploaded XML document is not valid for the application.

Table A.7.: Context Question for Use Case *Integrate Questionnaire*

Name of Use Case	Select Questionnaire for Enacting
Description	User selects a questionnaire which is set to active from a list for enacting.
Precondition	A questionnaire was set to active in questionnaire management by an administrator.
Postcondition	After selecting a questionnaire it is started.
Regular Expiration	1. User wants to enact a questionnaire. 2. A list with all activated questionnaires is displayed. 3. User selects one questionnaire from list. 4. Selected questionnaire is started.
Exceptional Cases	No questionnaire is activated in application.

Table A.8.: Context Question for Use Case *Select Questionnaire for Enacting*

Name of Use Case	Enacting Questionnaire
Description	A selected questionnaire can be enacted. Questions generated by XML document are displayed and can be filled out. It is the main use case for the application.
Precondition	A questionnaire was selected.
Postcondition	Results of questionnaire are stored in a database.
Regular Expiration	1. User selects a questionnaire. 2. User fills out questions of questionnaire. 3. After completing results are stored in database.
Exceptional Cases	XML document is not valid and thereby wrong interpreted by the application. User cancels questionnaire during enacting.

Table A.9.: Context Question for Use Case *Enact Questionnaire*

Name of Use Case	Select Language
Description	The application is accessible in different languages.
Precondition	Language package is installed on application.
Postcondition	Application conforms selected language.
Regular Expiration	1. User switches in settings menu. 2. User selects available language. 3. Application conforms selected language.
Exceptional Cases	-

Table A.10.: Context Question for Use Case *Select Language*

Name of Use Case	View Questionnaire Results
Description	If a questionnaire was enacted the results can be viewed in questionnaire management area.
Precondition	A questionnaire was enacted. User must be logged in.
Postcondition	Results of questionnaire can be viewed.
Regular Expiration	1. User logs into the application. 2. User switches to questionnaire management area. 3. User selects questionnaire. 4. All enacted results of questionnaire are listed.
Exceptional Cases	User who has not enacted questionnaire in which he wants to view results is not able to view the results.

Table A.11.: Context Question for Use Case *View Questionnaire Results*

Name of Use Case	Delete Questionnaire Results
Description	Results of completed questionnaires can be deleted from application database.
Precondition	Results for questionnaire are existing.
Postcondition	Results are deleted from database.
Regular Expiration	1. User logs into the application. 2. User switches to questionnaire management area. 3. User selects questionnaire. 4. All enacted results of questionnaire are listed. 5. Results can be deleted.
Exceptional Cases	-

Table A.12.: Context Question for Use Case *Delete Questionnaire Results*

Name of Use Case	Export Questionnaire Results
Description	Results of completed questionnaires can be export from application in CSV format. On the one hand for downloading to a computer and on the other hand to upload them into a server database.
Precondition	Results for questionnaire are existing. Device is connected with a computer if exported by local.
Postcondition	Results are exported.
Regular Expiration	1. User logs into the application. 2. User switches to questionnaire management area. 3. User selects questionnaire. 4. All enacted results of questionnaire are listed. 5. User logs in again for export. 6. Choosing of export variant. 7. Results are downloadable for a determined timeslot.
Exceptional Cases	Server is not available or device is not connected to a computer.

Table A.13.: Context Question for Use Case *Export Questionnaire Results*

User Interfaces (Chapter 4.2):

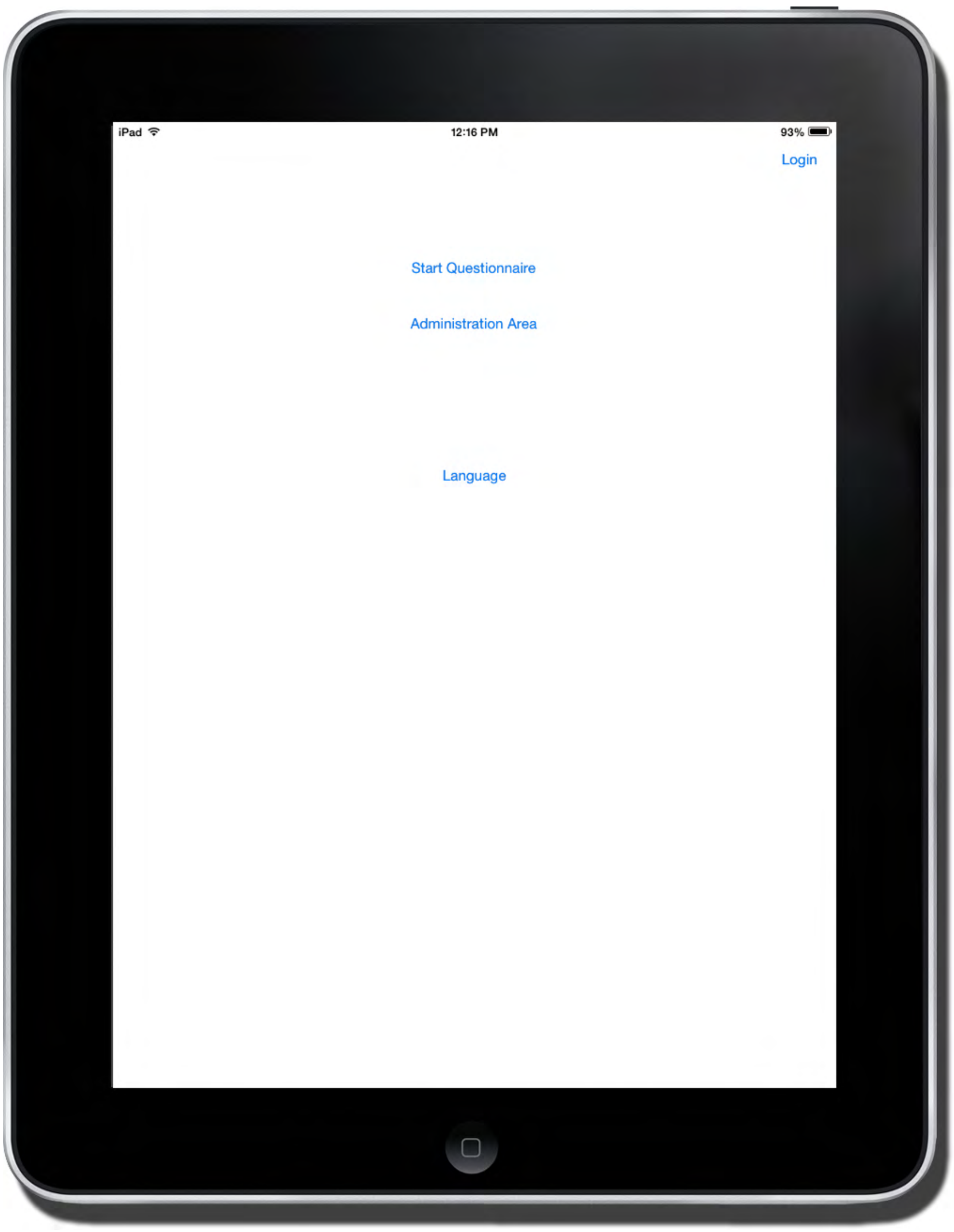


Figure A.1.: Main Menu – Overview

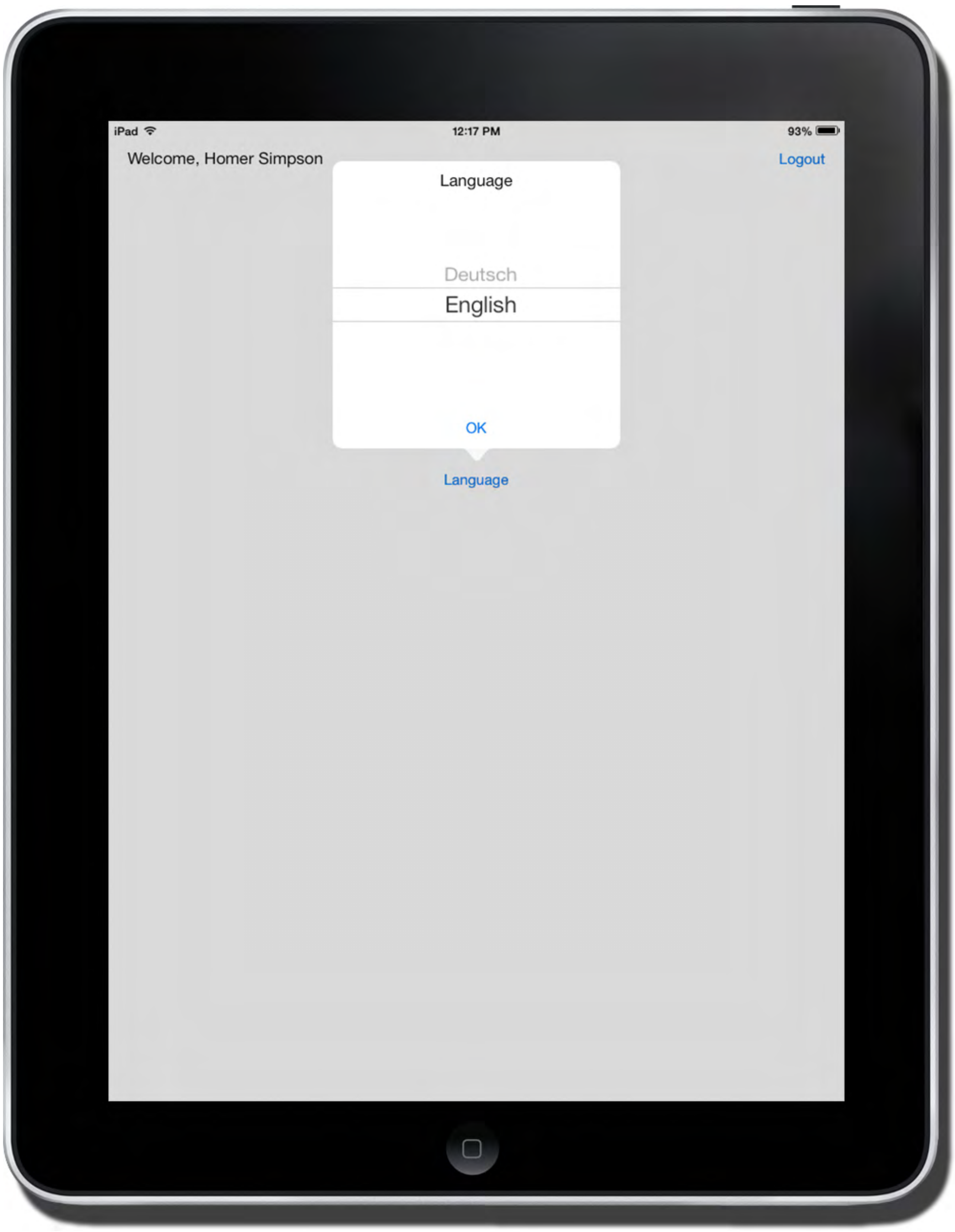


Figure A.2.: Main Menu – Language Selection

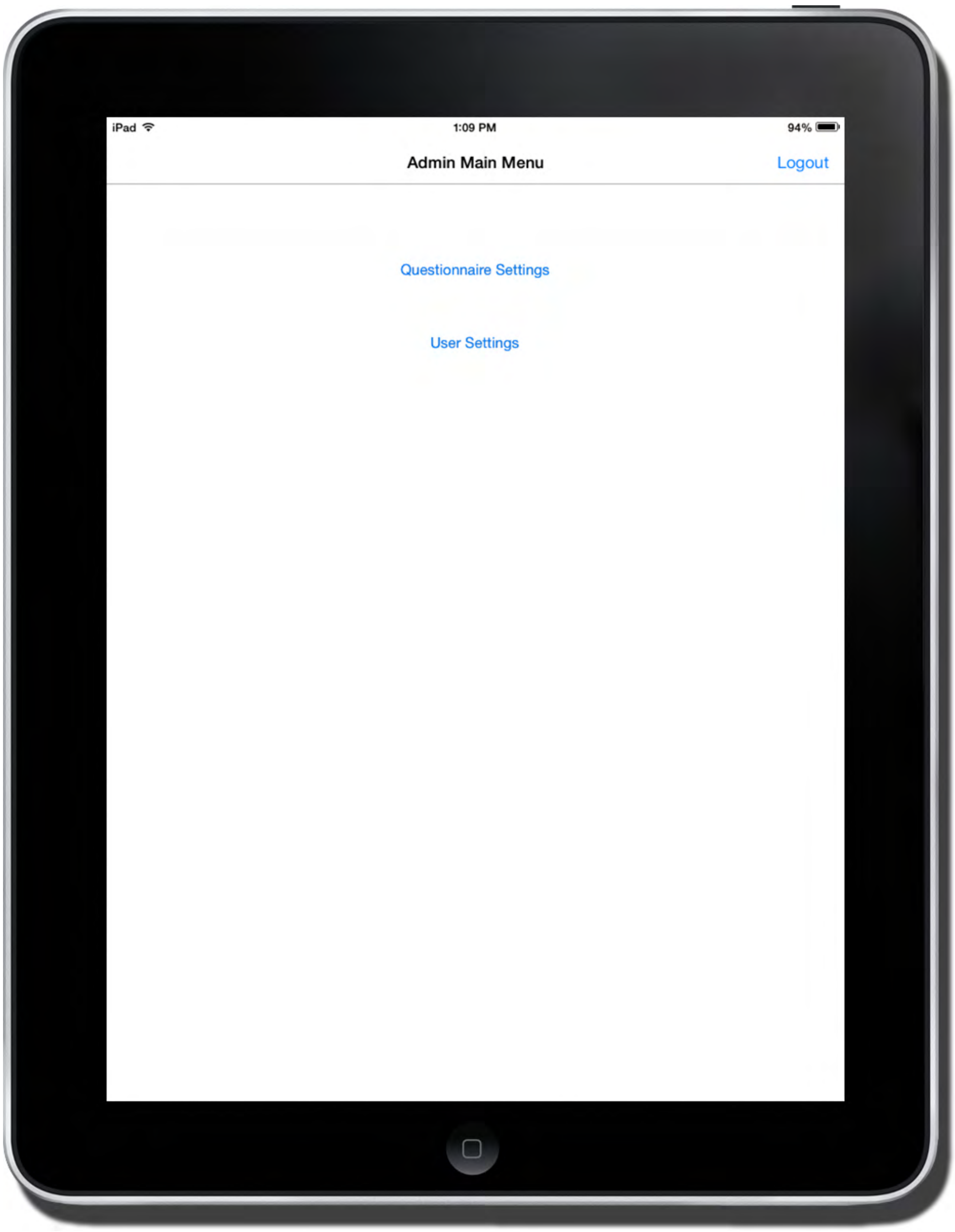
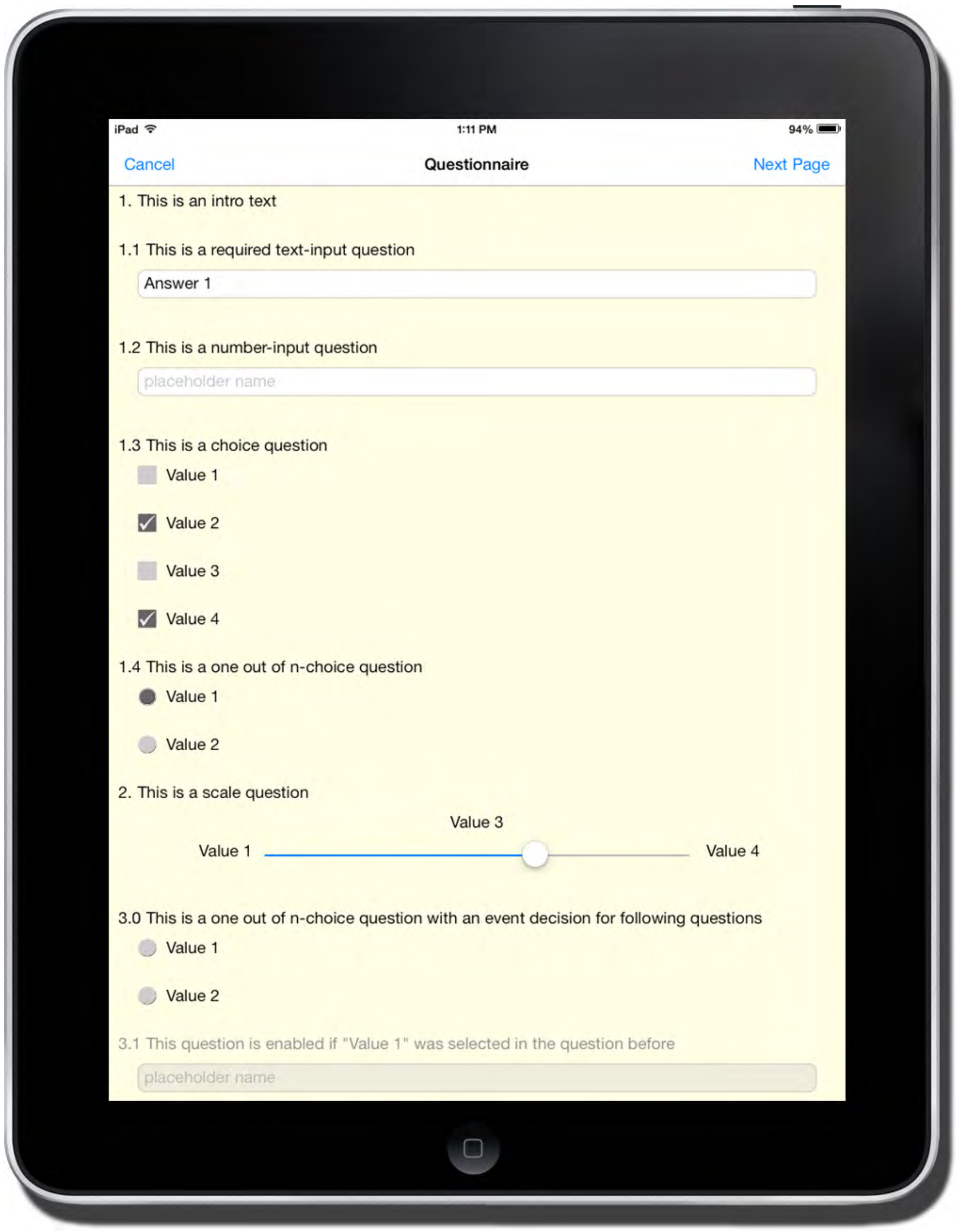


Figure A.3.: Administration Area – Overview



iPad 1:11 PM 94%

Cancel **Questionnaire** Next Page

1. This is an intro text

1.1 This is a required text-input question

Answer 1

1.2 This is a number-input question

placeholder name

1.3 This is a choice question

- [ ] Value 1
- [x] Value 2
- [ ] Value 3
- [x] Value 4

1.4 This is a one out of n-choice question

● Value 1

○ Value 2

2. This is a scale question

Value 3

Value 1 Value 4

3.0 This is a one out of n-choice question with an event decision for following questions

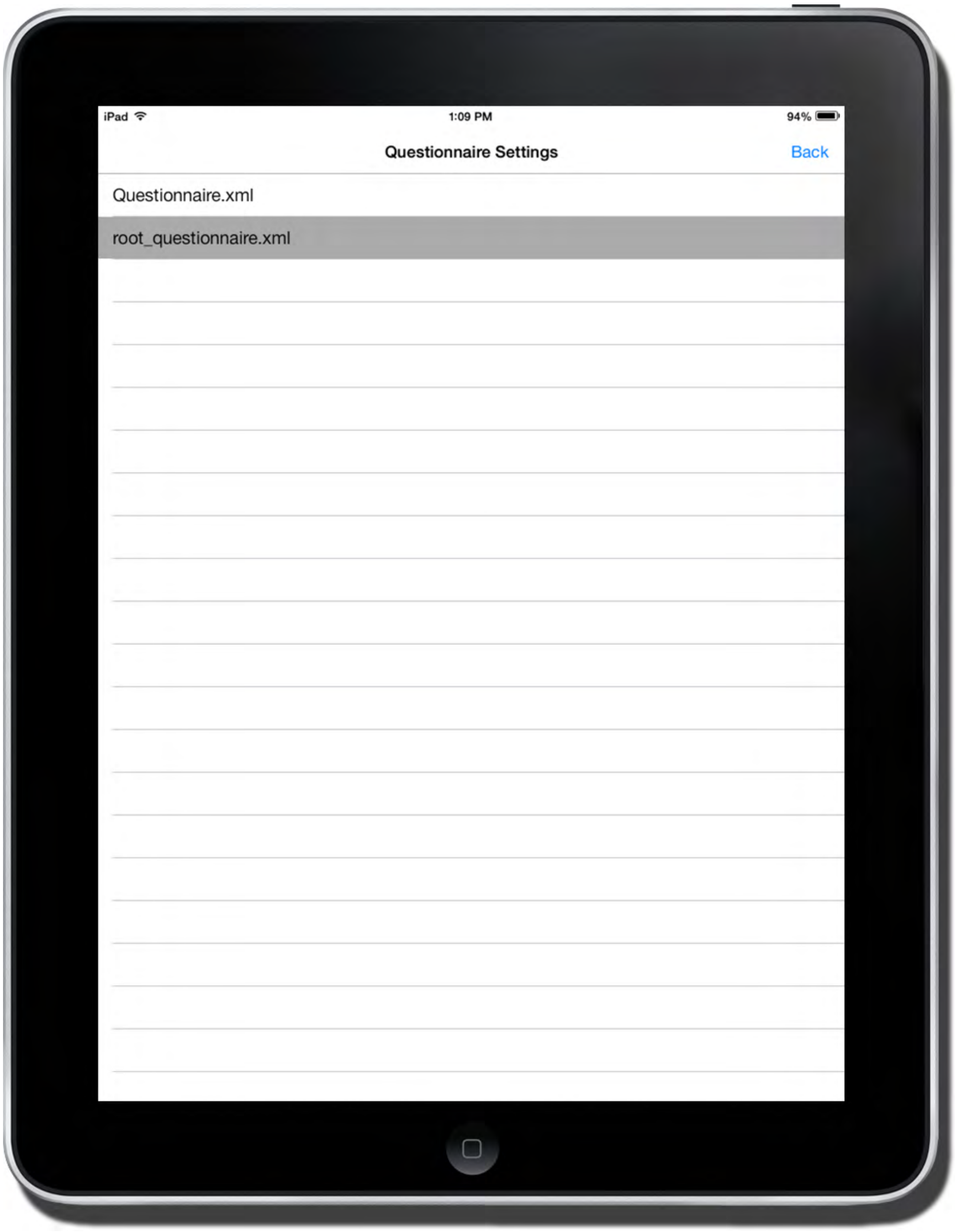
○ Value 1

○ Value 2

3.1 This question is enabled if "Value 1" was selected in the question before

placeholder name

Figure A.4.: Questionnaire



Questionnaire.xml
root_questionnaire.xml

Figure A.5.: Questionnaire Settings – Overview

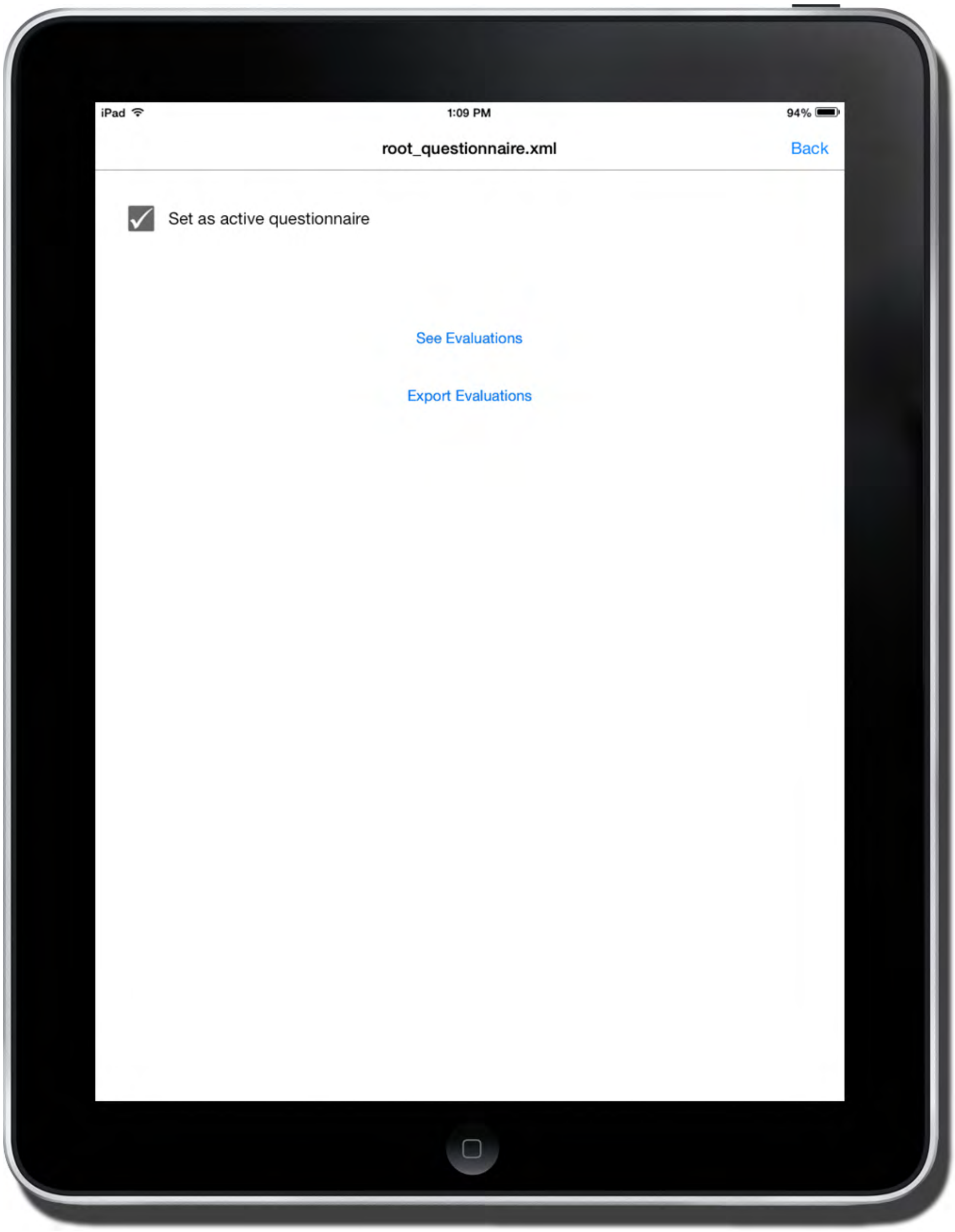
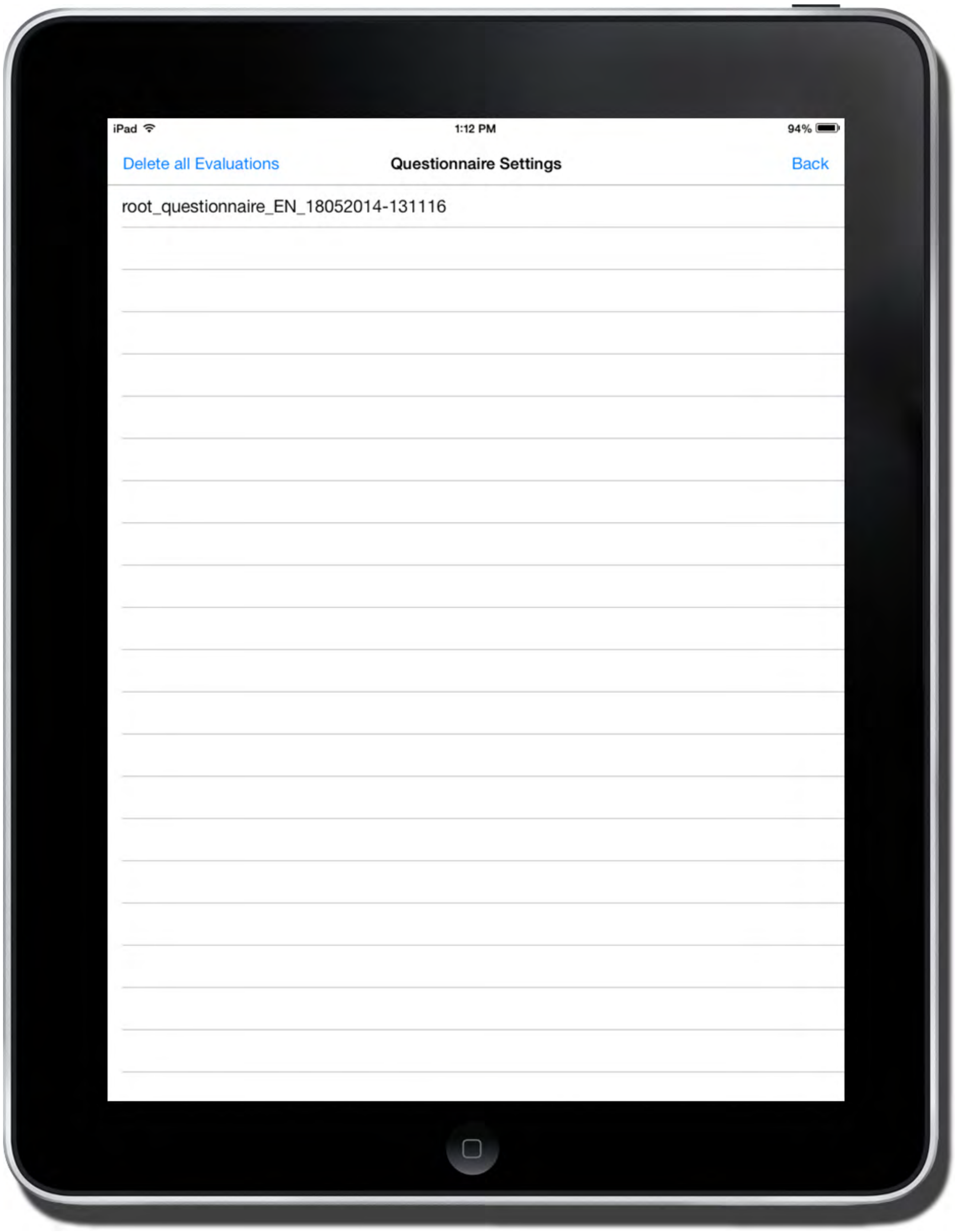
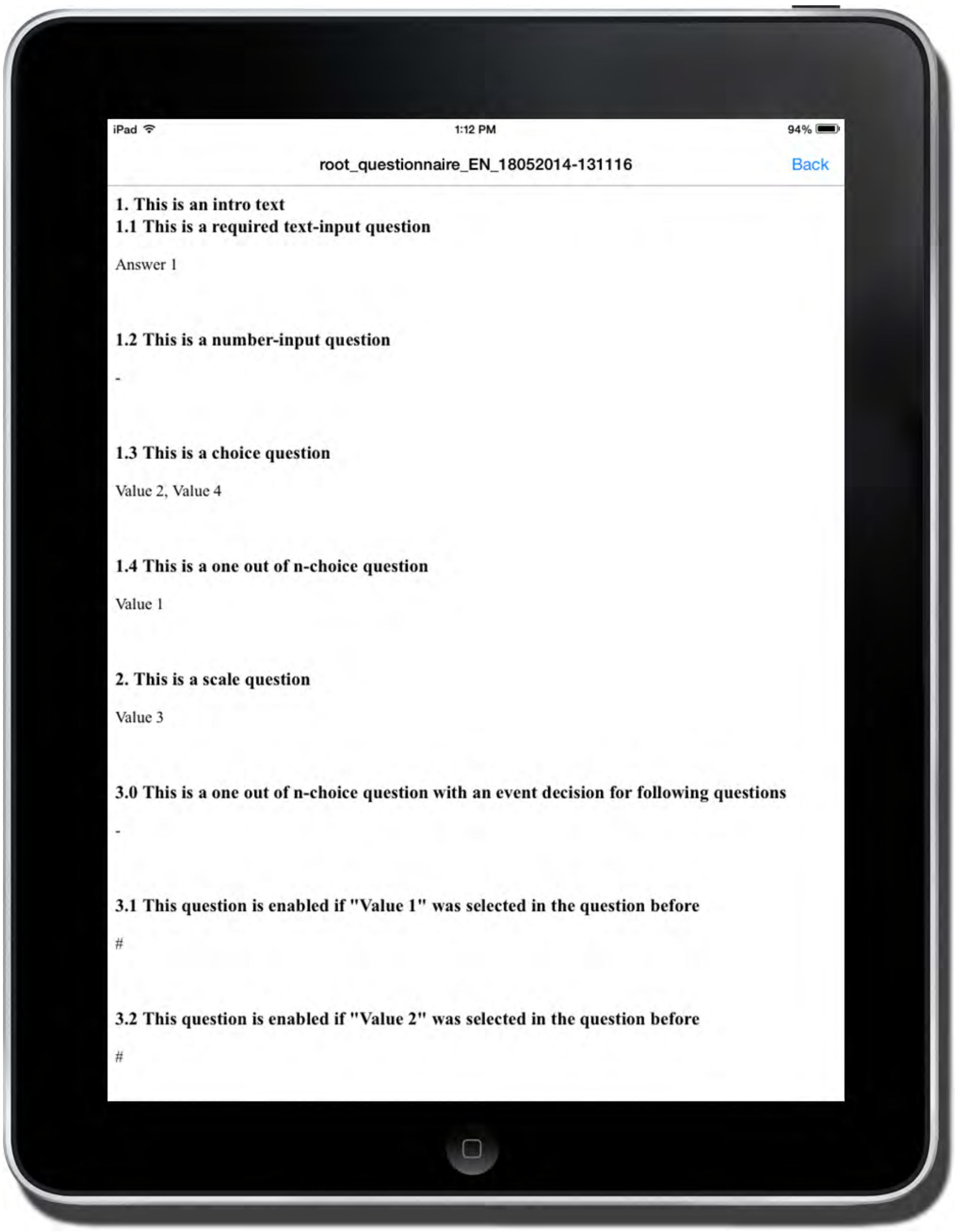


Figure A.6.: Questionnaire Settings – Detail



root_questionnaire_EN_18052014-131116

Figure A.7.: Questionnaire Results – Overview



root_questionnaire_EN_18052014-131116

1. This is an intro text
1.1 This is a required text-input question

Answer 1

1.2 This is a number-input question

-

1.3 This is a choice question

Value 2, Value 4

1.4 This is a one out of n-choice question

Value 1

2. This is a scale question

Value 3

3.0 This is a one out of n-choice question with an event decision for following questions

-

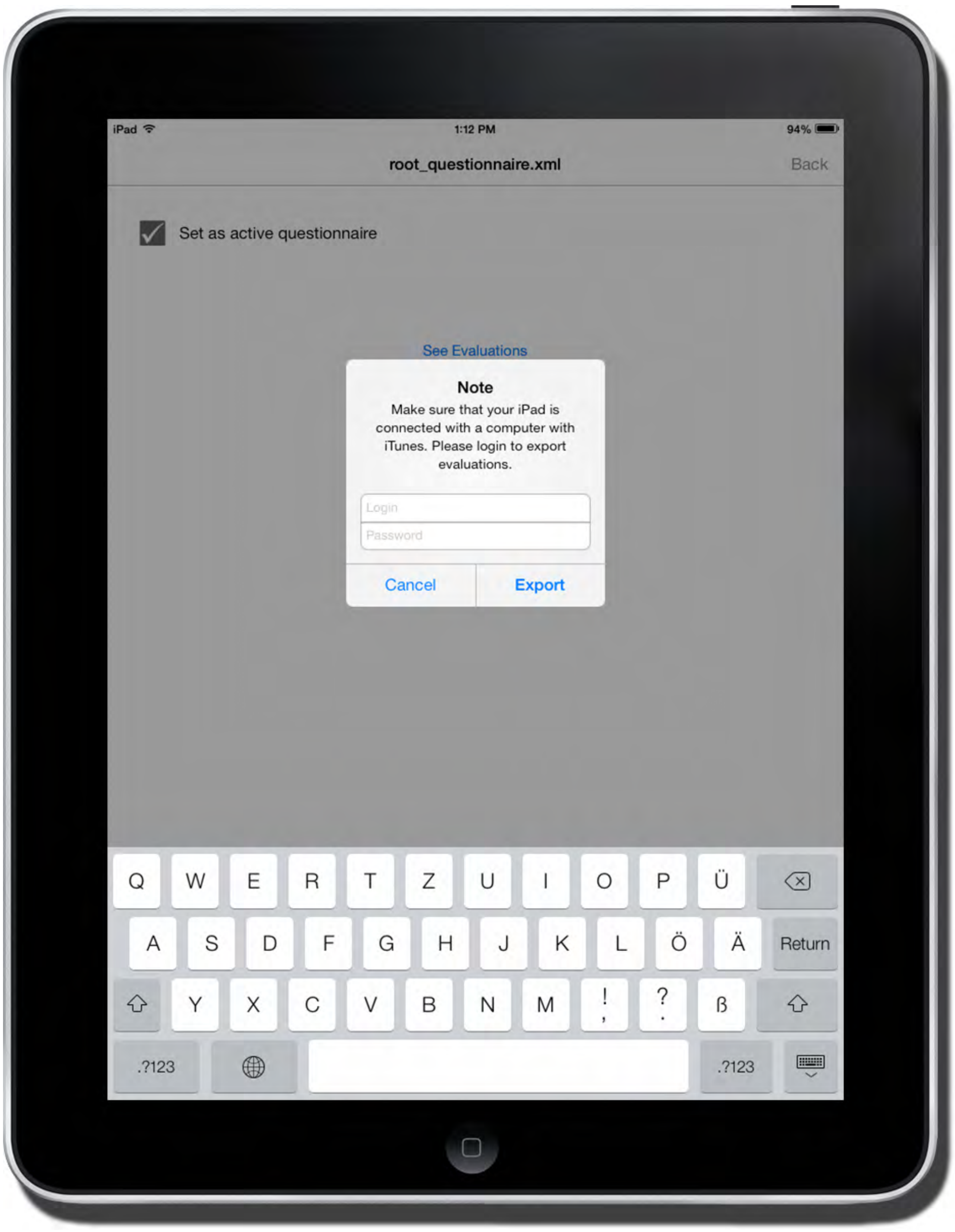
3.1 This question is enabled if "Value 1" was selected in the question before

#

3.2 This question is enabled if "Value 2" was selected in the question before

#

Figure A.8.: Questionnaire Results – Detail



Note

Make sure that your iPad is connected with a computer with iTunes. Please login to export evaluations.

Figure A.9.: Questionnaire Results – Export

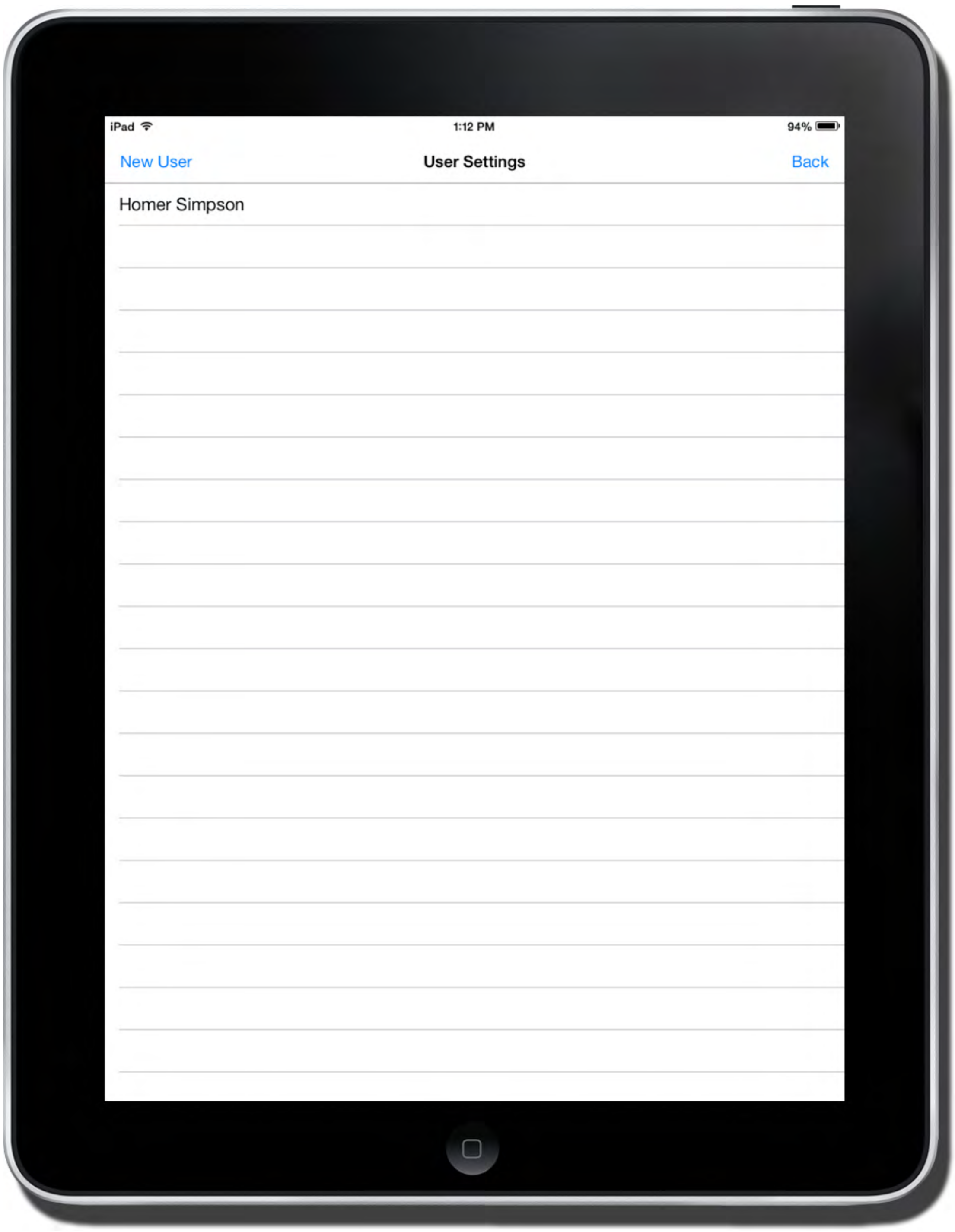


Figure A.10.: User Settings – Overview

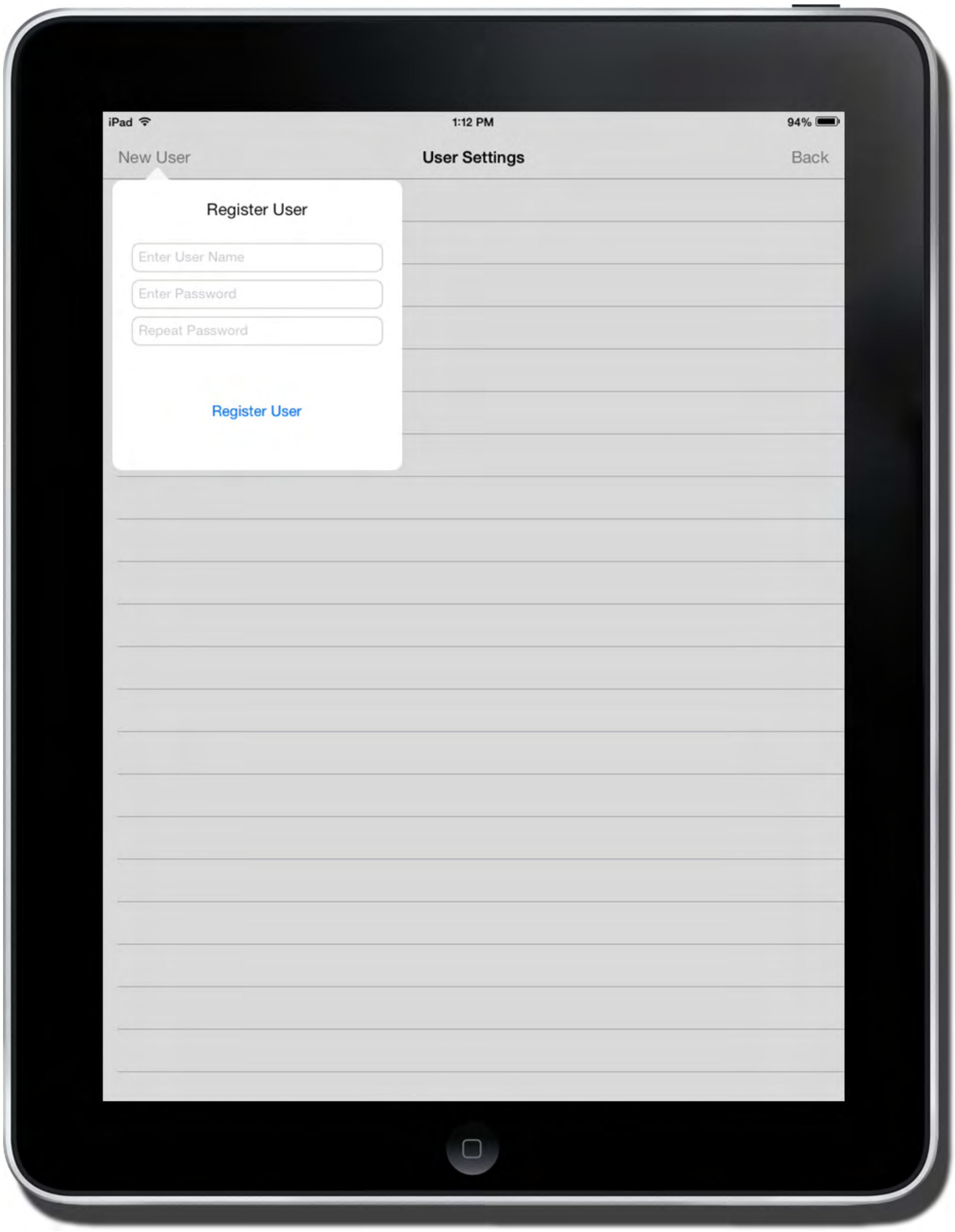
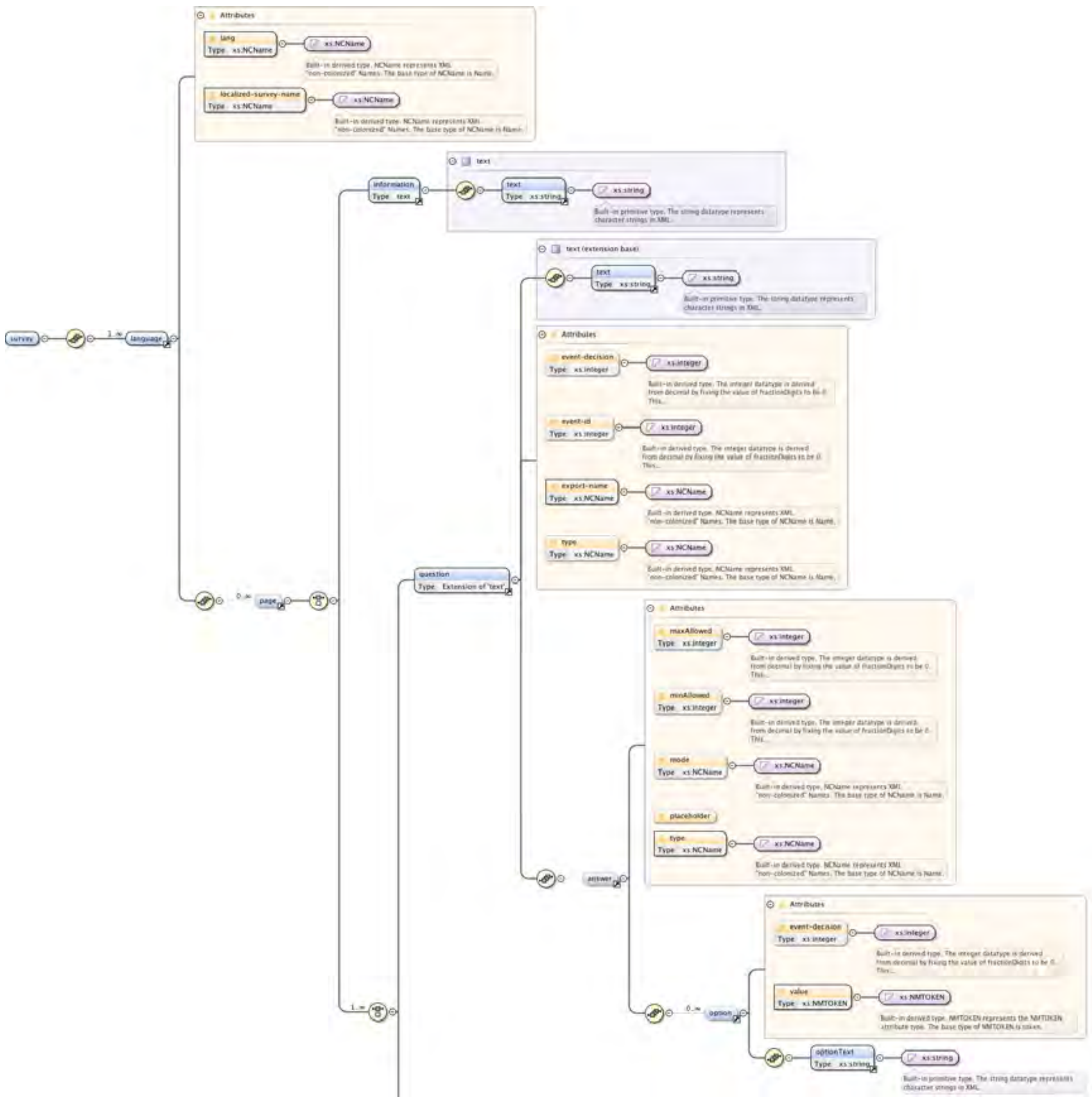


Figure A.11.: User Settings – User Registration


survey
1..∞
language
Attributes
lang
Type xs:NCName
xs:NCName
localized-survey-name
Type xs:NCName
0..∞
page
information
Type text
text
Type xs:string
xs:string
text (extension base)
Attributes
event-decision
Type xs:integer
xs:integer
event-id
Type xs:integer
export-name
Type xs:NCName
type
question
Type Extension of 'text'
answer
maxAllowed
minAllowed
mode
placeholder
option
value
Type xs:NMTOKEN
xs:NMTOKEN
optionText
Type xs:string
1..∞

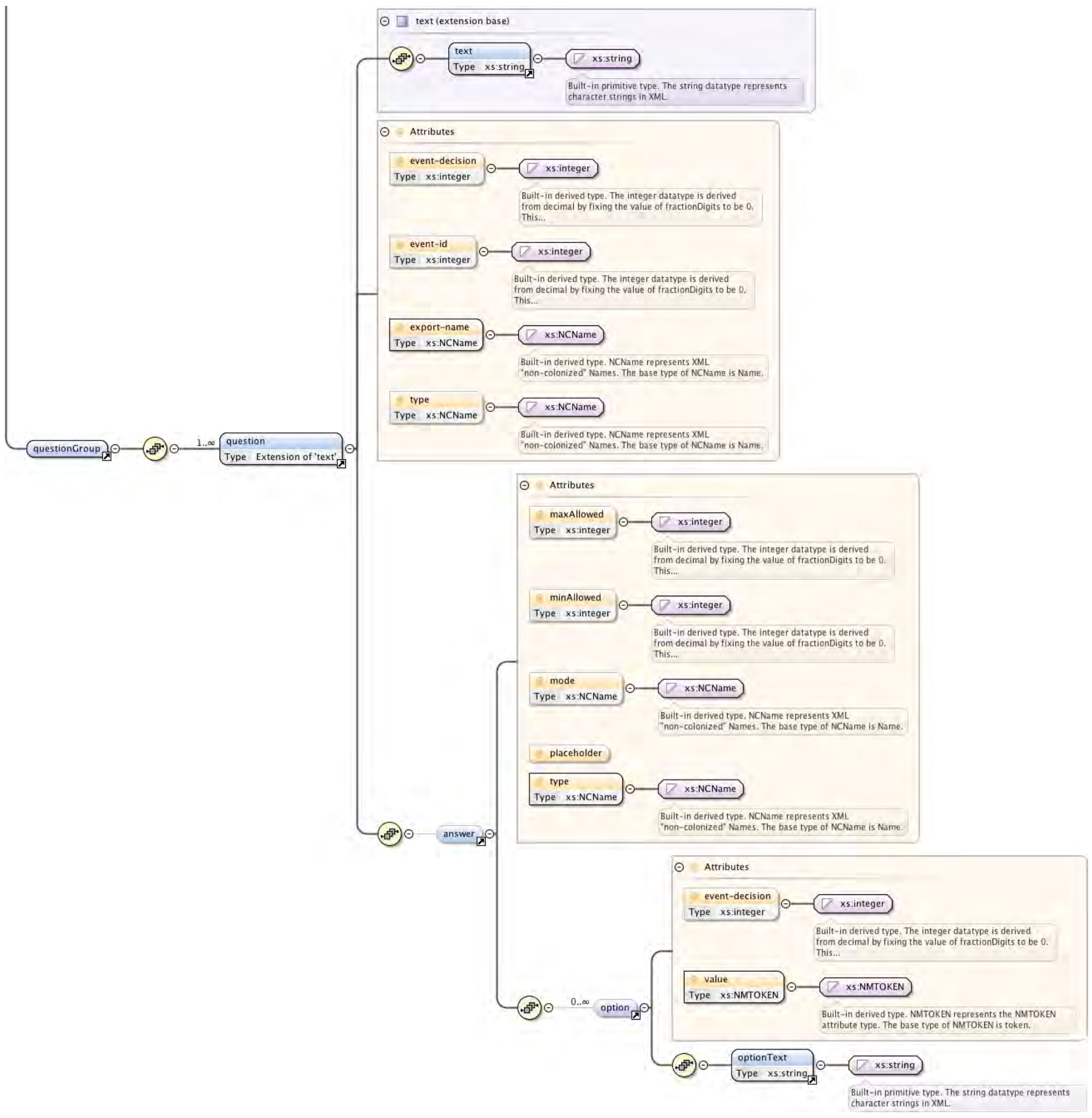


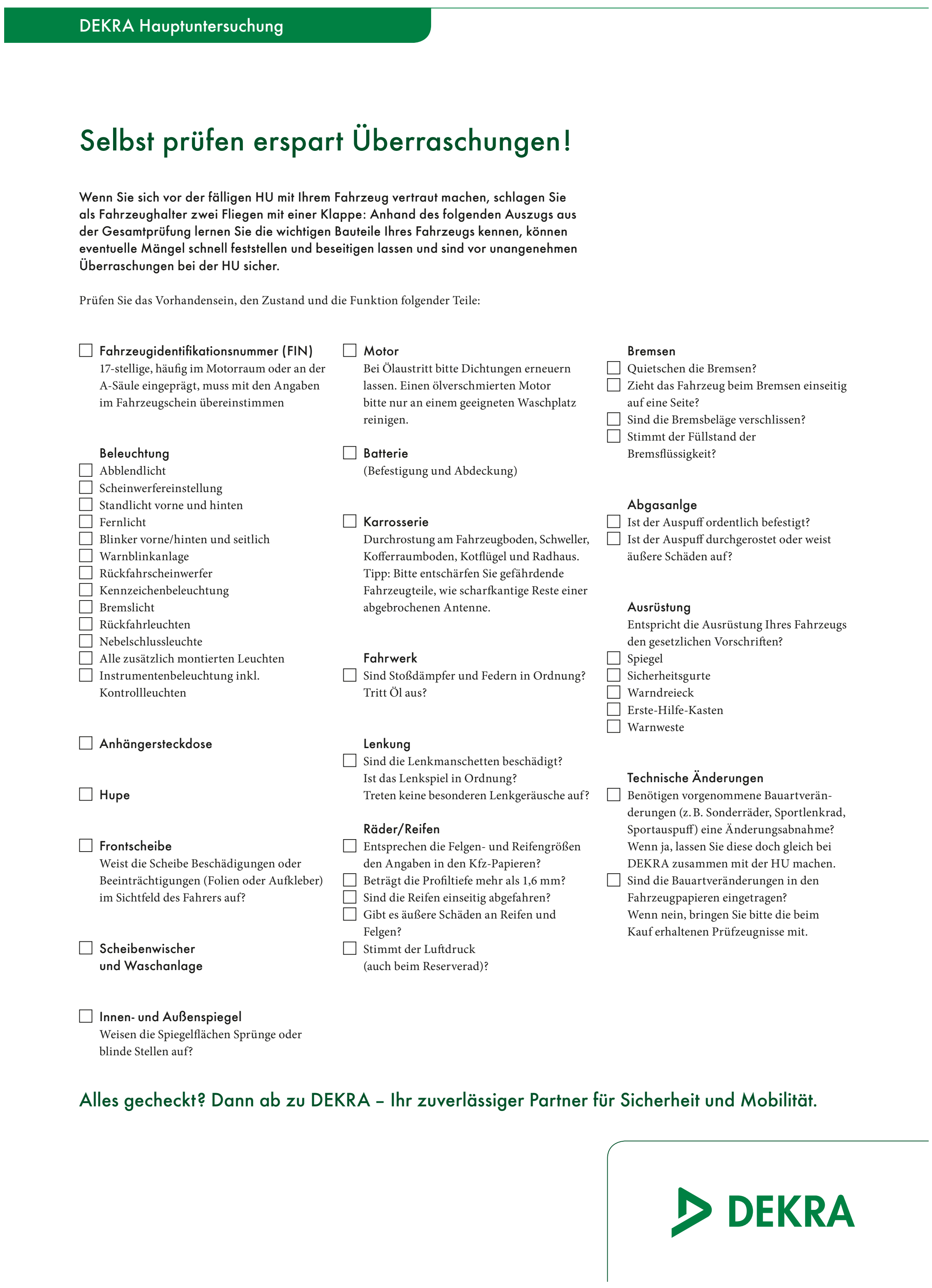
Figure A.12.: XML Document Syntax – Overview (Chapter 5.1)
Created with *oXygen XML* [oxy14]

Listing for XML Syntax:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<survey>
   <language lang="EN" localized-survey-name="English">
      <page>
         <information>
            <text><![CDATA[Information text as html]]></text>
         </information>
      </page>
      <page>
         <questionGroup>
            <question export-name="nameID_1" type="intro">
               <text>This is an intro text</text>
            </question>

            <question export-name="nameID_2" type="required">
               <text>This is a required text-input question</text>
               <answer type="free-text" mode="text" placeholder="placeholder name"/>
            </question>

            <question export-name="nameID_3">
               <text>This is a number-input question</text>
               <answer type="free-text" mode="number" placeholder="placeholder name"/>
            </question>

            <question export-name="nameID_4">
               <text>This is a choice question</text>
               <answer type="choice" minAllowed="1" maxAllowed="3">
                  <option value="value1">
                                    <optionText>Value 1</optionText>
                              </option>
                  <option value="value2">
                     <optionText>Value 2</optionText>
                  </option>
                  <option value="value3">
                     <optionText>Value 3</optionText>
                  </option>
                  <option value="value4">
                     <optionText>Value 4</optionText>
                  </option>
               </answer>
            </question>

            <question export-name="nameID_5">
               <text>This is a one out of n-choice question</text>
               <answer type="choice" maxAllowed="1">
                  <option value="value1">
                        <optionText>Value 1</optionText>
                  </option>
                  <option value="value2">
                        <optionText>Value 2</optionText>
                  </option>
               </answer>
            </question>
         </questionGroup>

         <question export-name="nameID_6">
            <text>This is a scale question</text>
            <answer type="scale">
               <option value="0">
```

```
                <optionText>Value 1</optionText>
             </option>
             <option value="1">
                <optionText>Value 2</optionText>
             </option>
             <option value="2">
                <optionText>Value 3</optionText>
             </option>
             <option value="3">
                <optionText>Value 4</optionText>
             </option>
          </answer>
       </question>

       <questionGroup>
          <question export-name="nameID_7" event-id="0">
             <text>This is a one out of n-choice question with an event decision for following questions</text>
             <answer type="choice" maxAllowed="1">
                <option value="value1" event-decision="0">
                   <optionText>Value 1</optionText>
                </option>
                <option value="value2" event-decision="1">
                   <optionText>Value 2</optionText>
                </option>
             </answer>
          </question>

          <question export-name="nameID_8" event-id="0" event-decision="0">
             <text>This question is enabled if "Value 1" was selected in the question before</text>
             <answer type="free-text" mode="text" placeholder="placeholder name"/>
          </question>

          <question export-name="nameID_9" event-id="0" event-decision="1">
             <text>This question is enabled if "Value 2" was selected in the question before</text>
             <answer type="free-text" mode="text" placeholder="placeholder name"/>
          </question>

       </questionGroup>
    </page>
 </language>
 <language lang="DE" localized-survey-name="Deutsch">
 </language>
</survey>
```

Listing A.1: Application-Specific Syntax of a XML Document (Chapter 5.1)



DEKRA Hauptuntersuchung

Selbst prüfen erspart Überraschungen!

Wenn Sie sich vor der fälligen HU mit Ihrem Fahrzeug vertraut machen, schlagen Sie als Fahrzeughalter zwei Fliegen mit einer Klappe: Anhand des folgenden Auszugs aus der Gesamtprüfung lernen Sie die wichtigen Bauteile Ihres Fahrzeugs kennen, können eventuelle Mängel schnell feststellen und beseitigen lassen und sind vor unangenehmen Überraschungen bei der HU sicher.

Prüfen Sie das Vorhandensein, den Zustand und die Funktion folgender Teile:

☐ **Fahrzeugidentifikationsnummer (FIN)**
17-stellige, häufig im Motorraum oder an der A-Säule eingeprägt, muss mit den Angaben im Fahrzeugschein übereinstimmen

Beleuchtung
- ☐ Abblendlicht
- ☐ Scheinwerfereinstellung
- ☐ Standlicht vorne und hinten
- ☐ Fernlicht
- ☐ Blinker vorne/hinten und seitlich
- ☐ Warnblinkanlage
- ☐ Rückfahrscheinwerfer
- ☐ Kennzeichenbeleuchtung
- ☐ Bremslicht
- ☐ Rückfahrleuchten
- ☐ Nebelschlussleuchte
- ☐ Alle zusätzlich montierten Leuchten
- ☐ Instrumentenbeleuchtung inkl. Kontrollleuchten

☐ **Anhängersteckdose**

☐ **Hupe**

☐ **Frontscheibe**
Weist die Scheibe Beschädigungen oder Beeinträchtigungen (Folien oder Aufkleber) im Sichtfeld des Fahrers auf?

☐ **Scheibenwischer und Waschanlage**

☐ **Innen- und Außenspiegel**
Weisen die Spiegelflächen Sprünge oder blinde Stellen auf?

☐ **Motor**
Bei Ölaustritt bitte Dichtungen erneuern lassen. Einen ölverschmierten Motor bitte nur an einem geeigneten Waschplatz reinigen.

☐ **Batterie**
(Befestigung und Abdeckung)

☐ **Karrosserie**
Durchrostung am Fahrzeugboden, Schweller, Kofferraumboden, Kotflügel und Radhaus. Tipp: Bitte entschärfen Sie gefährdende Fahrzeugteile, wie scharfkantige Reste einer abgebrochenen Antenne.

Fahrwerk
- ☐ Sind Stoßdämpfer und Federn in Ordnung? Tritt Öl aus?

Lenkung
- ☐ Sind die Lenkmanschetten beschädigt? Ist das Lenkspiel in Ordnung? Treten keine besonderen Lenkgeräusche auf?

Räder/Reifen
- ☐ Entsprechen die Felgen- und Reifengrößen den Angaben in den Kfz-Papieren?
- ☐ Beträgt die Profiltiefe mehr als 1,6 mm?
- ☐ Sind die Reifen einseitig abgefahren?
- ☐ Gibt es äußere Schäden an Reifen und Felgen?
- ☐ Stimmt der Luftdruck (auch beim Reserverad)?

Bremsen
- ☐ Quietschen die Bremsen?
- ☐ Zieht das Fahrzeug beim Bremsen einseitig auf eine Seite?
- ☐ Sind die Bremsbeläge verschlissen?
- ☐ Stimmt der Füllstand der Bremsflüssigkeit?

Abgasanlge
- ☐ Ist der Auspuff ordentlich befestigt?
- ☐ Ist der Auspuff durchgerostet oder weist äußere Schäden auf?

Ausrüstung
Entspricht die Ausrüstung Ihres Fahrzeugs den gesetzlichen Vorschriften?
- ☐ Spiegel
- ☐ Sicherheitsgurte
- ☐ Warndreieck
- ☐ Erste-Hilfe-Kasten
- ☐ Warnweste

Technische Änderungen
- ☐ Benötigen vorgenommene Bauartveränderungen (z. B. Sonderräder, Sportlenkrad, Sportauspuff) eine Änderungsabnahme? Wenn ja, lassen Sie diese doch gleich bei DEKRA zusammen mit der HU machen.
- ☐ Sind die Bauartveränderungen in den Fahrzeugpapieren eingetragen? Wenn nein, bringen Sie bitte die beim Kauf erhaltenen Prüfzeugnisse mit.

Alles gecheckt? Dann ab zu DEKRA – Ihr zuverlässiger Partner für Sicherheit und Mobilität.

DEKRA

Figure A.13.: Preliminary Checklist by DEKRA (Chapter 5) [Dek14]

XML document and XSD for Preliminary Check in Chapter 6

```xml
<?xml version="1.0" encoding="UTF-8"?>
<survey>
   <language lang="DE" localized-survey-name="Deutsch">
      <page>
         <information>
            <text><![CDATA[Herzlich willkommen zum Vorabchek fuer die anstehende
                        Hauptuntersuchung Ihres Fahrzeugs]]></text>
         </information>
      </page>
      <page>
         <questionGroup>
            <question export-name="general_nformationen" type="intro">
               <text>Informationen zum Vorabcheck</text>
            </question>
            <question export-name="name" type="required">
               <text>Name des Kontrolleurs:</text>
               <answer type="free-text" mode="text" placeholder="Bitte Name des Kontrolleurs eingeben"/>
            </question>
            <question export-name="date" type="required">
               <text>Datum der Kontrolle:</text>
               <answer type="free-text" mode="text" placeholder="Bitte heutiges Datum eingeben"/>
            </question>
            <question export-name="location" type="required">
               <text>Standort der Kontrolle:</text>
               <answer type="free-text" mode="text" placeholder="Bitte Standort eingeben"/>
            </question>
            <question export-name="reason" type="required">
               <text>Aus welchem Grund fuehren Sie diesen Vorabcheck durch?</text>
               <answer type="free-text" mode="text" placeholder="Bitte Grund eingeben"/>
            </question>
         </questionGroup>
      </page>
      <page>
         <questionGroup>
            <question export-name="vehicle_nformationen" type="intro">
               <text>Informationen zum Fahrzeug</text>
            </question>
            <question export-name="car_manufacturer" type="required">
               <text>Hersteller:</text>
               <answer type="free-text" mode="text" placeholder="Bitte Hersteller eingeben"/>
            </question>
            <question export-name="vehicle_type" type="required">
               <text>Fahrzeugtyp:</text>
               <answer type="free-text" mode="text" placeholder="Bitte Fahrzeugtyp eingeben"/>
            </question>
            <question export-name="year_of_manufacture" type="required">
               <text>Baujahr:</text>
               <answer type="free-text" mode="number" placeholder="Bitte Baujahr eingeben"/>
            </question>
            <question export-name="registration_date" type="required">
               <text>Erstzulassung:</text>
               <answer type="free-text" mode="number" placeholder="Bitte Erstzulassung eingeben"/>
            </question>
            <question export-name="vehicle_id" type="required">
               <text>Fahrzeugidentifikationsnummer:</text>
               <answer type="free-text" mode="number" placeholder="Bitte Fahrzeugidentifikationsnummer eingeben"/>
            </question>
         </questionGroup>
      </page>
      <page>
         <information>
            <text><![CDATA[<style type="text/css">body {background-color:#FFFEE5;}</style>Nachfolgend werden einige
                        Fragen zu Ihrem Fahrzeug gestellt.]]></text>
         </information>
      </page>
      <page>
         <questionGroup>
            <question export-name="check_illumination" type="intro">
               <text>Pruefung der Beleuchtung</text>
            </question>
            <question export-name="dimmed_headlights" event-id="0">
               <text>Funktioniert das Abblendlicht rechts und links?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="dimmed_headlights_yes" event-decision="0">
                     <optionText>Ja</optionText>
                  </option>
                  <option value="dimmed_headlights_no" event-decision="1">
                     <optionText>Nein</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="dimmed_headlights_option" event-id="0" event-decision="1">
               <text>Wenn nein, auf welcher Seite funktioniert es nicht?</text>
               <answer type="choice" minAllowed="1" maxAllowed="2">
                  <option value="dimmed_headlights_left">
                     <optionText>Links</optionText>
                  </option>
                  <option value="dimmed_headlights_right">
                     <optionText>Rechts</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="floodlight">
               <text>Sind die Scheinwerfer korrekt eingestellt?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="floodlight_yes">
                     <optionText>Ja</optionText>
                  </option>
                  <option value="floodlight_no">
                     <optionText>Nein</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="parking_light" event-id="1">
               <text>Funktioniert das Standlicht?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="parking_light_yes" event-decision="0">
                     <optionText>Ja</optionText>
                  </option>
                  <option value="parking_light_no" event-decision="1">
                     <optionText>Nein</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="parking_light_option" event-id="1" event-decision="1">
               <text>Wenn nein, wo funktioniert es nicht?</text>
               <answer type="choice" minAllowed="1" maxAllowed="2">
                  <option value="parking_light_ahead">
                     <optionText>Vorne</optionText>
```

```xml
                    </option>
                    <option value="parking_light_behind">
                        <optionText>Hinten</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="high_beam" event-id="3">
                <text>Funktioniert das Fernlicht rechts und links?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="high_beam_yes" event-decision="0">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="high_beam_no" event-decision="1">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="high_beam_option" event-id="3" event-decision="1">
                <text>Wenn nein, auf welcher Seite funktioniert es nicht?</text>
                <answer type="choice" minAllowed="1" maxAllowed="2">
                    <option value="high_beam_left">
                        <optionText>Links</optionText>
                    </option>
                    <option value="high_beam_right">
                        <optionText>Rechts</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="direction_indicator" event-id="4">
                <text>Funktionieren alle Blinker?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="direction_indicator_yes" event-decision="0">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="direction_indicator_no" event-decision="1">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="direction_indicator_option" type="required" event-id="4" event-decision="1">
                <text>Wenn nein, welche funktionieren nicht?</text>
                <answer type="free-text" mode="text" placeholder="Bitte geben Sie alle Blinker an,
                               die nicht funktionieren"/>
            </question>
            <question export-name="warning_lights">
                <text>Funktioniert die Warnblinkanlage?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="warning_lights_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="warning_lights_no">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="reversing_lights">
                <text>Funktioniert der Rueckfahrscheinwerfer?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="reversing_lights_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="reversing_lights_no">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="license_plate_lights">
                <text>Funktioniert die Kennzeichenleuchte?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="license_plate_lights_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="license_plate_lights_no">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="brake_light" event-id="5">
                <text>Funktioniert das Bremslicht rechts und links?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="brake_light_yes" event-decision="0">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="brake_light_no" event-decision="1">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="brake_light_option" event-id="5" event-decision="1">
                <text>Wenn nein, auf welcher Seite funktioniert es nicht?</text>
                <answer type="choice" minAllowed="1" maxAllowed="2">
                    <option value="brake_light_left">
                        <optionText>Links</optionText>
                    </option>
                    <option value="brake_light_right">
                        <optionText>Rechts</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="rear_fog_lamp">
                <text>Funktioniert die Nebelschlussleuchte?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="rear_fog_lamp_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="rear_fog_lamp_no">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="indicator_lamps" event-id="6">
                <text>Funktioniert die Instrumentenbeleuchtung inkl. Kontrollleuchten?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="indicator_lamps_yes" event-decision="0">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="indicator_lamps_no" event-decision="1">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="direction_indicator_option" type="required" event-id="6" event-decision="1">
               <text>Wenn nein, was funktioniert nicht?</text>
               <answer type="free-text" mode="text" placeholder="Bitte geben Sie an, was nicht funktioniert"/>
            </question>
         </questionGroup>
      </page>
      <page>
         <question export-name="horn">
            <text>Funktioniert die Hupe?</text>
            <answer type="choice" maxAllowed="1">
               <option value="horn_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="horn_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="windshield">
            <text>Weist die Frontscheibe Beschaedigungen oder Beeintraechtigungen im
                  Sichtfeld des Fahrers auf?</text>
            <answer type="choice" maxAllowed="1">
               <option value="windshield_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="windshield_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="wiper" event-id="7">
            <text>Funktionieren Scheibenwischer und Waschanlage?</text>
            <answer type="choice" maxAllowed="1">
               <option value="wiper_yes" event-decision="0">
                  <optionText>Ja</optionText>
               </option>
               <option value="wiper_no" event-decision="1">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="wiper_option" type="required" event-id="7" event-decision="1">
            <text>Wenn nein, was funktioniert nicht?</text>
            <answer type="free-text" mode="text" placeholder="Bitte geben Sie an, was nicht funktioniert"/>
         </question>
         <question export-name="mirror" event-id="8">
            <text>Weisen Innen- und Aussenspiegel Spruenge auf?</text>
            <answer type="choice" maxAllowed="1">
               <option value="mirror_yes" event-decision="0">
                  <optionText>Ja</optionText>
               </option>
               <option value="mirror_no" event-decision="1">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="mirror_option" event-id="8" event-decision="0">
            <text>Wenn ja, wo?</text>
            <answer type="choice" minAllowed="1" maxAllowed="2">
```

```xml
            <option value="mirror_inside">
               <optionText>Innenspiegel</optionText>
            </option>
            <option value="mirror_exterior">
               <optionText>Aussenspiegel</optionText>
            </option>
         </answer>
      </question>
      <question export-name="motor_oil">
         <text>Tritt aus dem Motor Oel aus?</text>
         <answer type="choice" maxAllowed="1">
            <option value="motor_oil_yes">
               <optionText>Ja</optionText>
            </option>
            <option value="motor_oil_no">
               <optionText>Nein</optionText>
            </option>
         </answer>
      </question>
      <question export-name="car_body_rust" event-id="9">
         <text>Weist die Karosserie Rostmaengel auf?</text>
         <answer type="choice" maxAllowed="1">
            <option value="car_body_rust_yes" event-decision="0">
               <optionText>Ja</optionText>
            </option>
            <option value="car_body_rust_no" event-decision="1">
               <optionText>Nein</optionText>
            </option>
         </answer>
      </question>
      <question export-name="car_body_rust_option" event-id="9" event-decision="0">
         <text>Wenn ja, bitte ankreuzen an welchen Stellen an Ihrem Fahrzeug</text>
         <answer type="choice" minAllowed="0" maxAllowed="4">
            <option value="vehicle_ground">
               <optionText>Fahrzeugboden</optionText>
            </option>
            <option value="vehicle_door">
               <optionText>Schweller</optionText>
            </option>
            <option value="trunk_ground">
               <optionText>Kofferraumboden</optionText>
            </option>
            <option value="wheel_case">
               <optionText>Radhaus</optionText>
            </option>
         </answer>
      </question>
      <question export-name="wiper_option" type="required" event-id="9" event-decision="0">
         <text>Andere Roststellen:</text>
         <answer type="free-text" mode="text" placeholder="Falls keine Auswahl von oben zutrifft,
                     bitte festgestellte Roststelle angeben"/>
      </question>
   </page>
   <page>
      <questionGroup>
         <question export-name="steering" type="intro">
            <text>Lenkung</text>
         </question>
         <question export-name="steering_test" event-id="10">
            <text>Ist das Lenkspiel in Ordnung?</text>
```

```
                <answer type="choice" maxAllowed="1">
                    <option value="steering_test_yes" event-decision="0">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="steering_test_no" event-decision="1">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="steering_test_option" type="required" event-id="10" event-decision="1">
                <text>Wenn nein, was ist nicht in Ordnung?</text>
                <answer type="free-text" mode="text" placeholder="Bitte geben Sie an, was nicht in Ordnung ist"/>
            </question>
            <question export-name="steering_sound">
                <text>Treten besondere Lenkgeraeusche auf?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="steering_sound_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="steering_sound_no">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
        </questionGroup>
        <questionGroup>
            <question export-name="wheels" type="intro">
                <text>Raeder/Reifen</text>
            </question>
            <question export-name="wheel_size">
                <text>Entsprechen die Felgen- und Reifengroessen den Angaben in den Kfz-Papieren?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="wheel_size_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="wheel_size_no">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="skid_depth">
                <text>Betraegt die Profiltiefe mehr als 1,6 mm?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="skid_depth_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="skid_depth_no">
                        <optionText>Nein</optionText>
                    </option>
                </answer>
            </question>
            <question export-name="wheel_abrasion">
                <text>Sind die Reifen einseitig abgefahren?</text>
                <answer type="choice" maxAllowed="1">
                    <option value="wheel_abrasion_yes">
                        <optionText>Ja</optionText>
                    </option>
                    <option value="wheel_abrasion_no">
                        <optionText>Nein</optionText>
                    </option>
```

```xml
            </answer>
         </question>
         <question export-name="wheel_defects" event-id="11">
            <text>Gibt es aeussere Schaeden an Reifen und Felgen?</text>
            <answer type="choice" maxAllowed="1">
               <option value="wheel_defects_yes" event-decision="0">
                  <optionText>Ja</optionText>
               </option>
               <option value="wheel_defects_no" event-decision="1">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="wheel_defects_option" type="required" event-id="11" event-decision="0">
            <text>Wenn ja, welche?</text>
            <answer type="free-text" mode="text" placeholder="Bitte geben Sie die Schaeden an"/>
         </question>
         <question export-name="air_pressure">
            <text>Ist der Luftdruck korrekt?</text>
            <answer type="choice" maxAllowed="1">
               <option value="air_pressure_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="air_pressure_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
      </questionGroup>
   </page>
   <page>
      <questionGroup>
         <question export-name="breaks" type="intro">
            <text>Bremsen</text>
         </question>
         <question export-name="break_squeak">
            <text>Quietschen die Bremsen?</text>
            <answer type="choice" maxAllowed="1">
               <option value="break_squeak_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="break_squeak_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="break_site">
            <text>Zieht das Fahrzeug beim Bremsen einseitig auf eine Seite?</text>
            <answer type="choice" maxAllowed="1">
               <option value="break_site_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="break_site_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="break_fluid">
            <text>Stimmt der Fuellstand der Bremsfluessigkeit?</text>
            <answer type="choice" maxAllowed="1">
               <option value="break_fluid_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="break_fluid_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
      </questionGroup>
      <questionGroup>
         <question export-name="flue_gas_system" type="intro">
            <text>Abgasanlage</text>
         </question>
         <question export-name="exhaust_attachement">
            <text>Ist der Auspuff ordentlich befestigt?</text>
            <answer type="choice" maxAllowed="1">
               <option value="exhaust_attachement_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="exhaust_attachement_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
         <question export-name="exhaust_sounds">
            <text>Sind laute Geraeusche am Auspuff zu hoeren?</text>
            <answer type="choice" maxAllowed="1">
               <option value="exhaust_sounds_yes">
                  <optionText>Ja</optionText>
               </option>
               <option value="exhaust_sounds_no">
                  <optionText>Nein</optionText>
               </option>
            </answer>
         </question>
      </questionGroup>
      <questionGroup>
         <question export-name="utensils" type="intro">
            <text>Ausruestung</text>
         </question>
         <question export-name="vehicle_utensils">
            <text>Befindet sich folgende Ausruestung im Fahrzeug? (Bitte ankreuzen)</text>
            <answer type="choice" minAllowed="0" maxAllowed="3">
               <option value="warning_triangle">
                  <optionText>Warndreieck</optionText>
               </option>
               <option value="first_aid_kit">
                  <optionText>Erste-Hilfe-Kasten</optionText>
               </option>
               <option value="safety_vest">
                  <optionText>Warnweste</optionText>
               </option>
            </answer>
         </question>
      </questionGroup>
      <question export-name="feasibility">
         <text>War der Vorabcheck fuer Sie leicht durchfuehrbar?</text>
         <answer type="scale">
            <option value="0">
               <optionText>leicht</optionText>
```

```
                </option>
                <option value="1">
                   <optionText>gut durchfuehrbar</optionText>
                </option>
                <option value="2">
                   <optionText>geht so</optionText>
                </option>
                <option value="3">
                   <optionText>schwer</optionText>
                </option>
                <option value="4">
                   <optionText>ohne Hilfe nicht durchfuehrbar</optionText>
                </option>
             </answer>
          </question>
       </page>
       <page>
          <information>
             <text><![CDATA[<style type="text/css">body {background-color:#FFFEE5;}</style>Vielen Dank fuer die
                         Durchfuehrung dieses Vorabchecks zur Hauptuntersuchung Ihres Fahrzeuges.]]></text>
          </information>
       </page>
    </language>
    <language lang="EN" localized-survey-name="English">
        <page>
          <information>
             <text><![CDATA[<style type="text/css">body {background-color:#FFFEE5;}</style>Welcome to the
                         preliminary check for the general inspection of your vehicle]]></text>
          </information>
       </page>
       <page>
          <questionGroup>
             <question export-name="general_nformationen" type="intro">
                <text>Information about the preliminary check</text>
             </question>
             <question export-name="name" type="required">
                <text>Name of inspector:</text>
                <answer type="free-text" mode="text" placeholder="Please insert the name of the inspector"/>
             </question>
             <question export-name="date" type="required">
                <text>Date of inspection:</text>
                <answer type="free-text" mode="text" placeholder="Please insert today’s date"/>
             </question>
             <question export-name="location" type="required">
                <text>Location of inspection:</text>
                <answer type="free-text" mode="text" placeholder="Please insert location"/>
             </question>
             <question export-name="reason" type="required">
                <text>What is the reason for performing the preliminary check?</text>
                <answer type="free-text" mode="text" placeholder="Please insert reason"/>
             </question>
          </questionGroup>
       </page>
       <page>
          <questionGroup>
             <question export-name="vehicle_nformationen" type="intro">
                <text>Information about vehicle</text>
             </question>
             <question export-name="car_manufacturer" type="required">
                <text>Manufacturer:</text>
```

```xml
         <answer type="free-text" mode="text" placeholder="Please insert manufacturer"/>
      </question>
      <question export-name="vehicle_type" type="required">
         <text>Vehicle type:</text>
         <answer type="free-text" mode="text" placeholder="Please insert vehicle type"/>
      </question>
      <question export-name="year_of_manufacture" type="required">
         <text>Year of manufacture:</text>
         <answer type="free-text" mode="number" placeholder="Please insert year of manufacture"/>
      </question>
      <question export-name="registration_date" type="required">
         <text>Registration date:</text>
         <answer type="free-text" mode="number" placeholder="Please insert registration date"/>
      </question>
      <question export-name="vehicle_id" type="required">
         <text>Vehicle identification number:</text>
         <answer type="free-text" mode="number" placeholder="Please insert vehicle identification number"/>
      </question>
   </questionGroup>
</page>
<page>
   <information>
      <text><![CDATA[<style type="text/css">body {background-color:#FFFEE5;}</style>In the following,
                  vehicle specific quesions have to be answered to enacting the
                  actual preliminary check]]></text>
   </information>
</page>
<page>
   <questionGroup>
      <question export-name="check_illumination" type="intro">
         <text>Inspection of the illumination</text>
      </question>
      <question export-name="dimmed_headlights" event-id="0">
         <text>Works the dimmed headlight left and right?</text>
         <answer type="choice" maxAllowed="1">
            <option value="dimmed_headlights_yes" event-decision="0">
               <optionText>Yes</optionText>
            </option>
            <option value="dimmed_headlights_no" event-decision="1">
               <optionText>No</optionText>
            </option>
         </answer>
      </question>
      <question export-name="dimmed_headlights_option" event-id="0" event-decision="1">
         <text>If not, on which side it does not work?</text>
         <answer type="choice" minAllowed="1" maxAllowed="2">
            <option value="dimmed_headlights_left">
               <optionText>Left</optionText>
            </option>
            <option value="dimmed_headlights_right">
               <optionText>Right</optionText>
            </option>
         </answer>
      </question>
      <question export-name="floodlight">
         <text>Are the headlamps set correctly?</text>
         <answer type="choice" maxAllowed="1">
            <option value="floodlight_yes">
               <optionText>Yes</optionText>
            </option>
```

```xml
                  <option value="floodlight_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="parking_light" event-id="1">
               <text>Works the parking lights?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="parking_light_yes" event-decision="0">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="parking_light_no" event-decision="1">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="parking_light_option" event-id="1" event-decision="1">
               <text>If not, where it does not work?</text>
               <answer type="choice" minAllowed="1" maxAllowed="2">
                  <option value="parking_light_ahead">
                     <optionText>Ahead</optionText>
                  </option>
                  <option value="parking_light_behind">
                     <optionText>Behind</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="high_beam" event-id="3">
               <text>Works the high beam left and right?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="high_beam_yes" event-decision="0">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="high_beam_no" event-decision="1">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="high_beam_option" event-id="3" event-decision="1">
               <text>If not, on which side it does not work?</text>
               <answer type="choice" minAllowed="1" maxAllowed="2">
                  <option value="high_beam_left">
                     <optionText>Left</optionText>
                  </option>
                  <option value="high_beam_right">
                     <optionText>Right</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="direction_indicator" event-id="4">
               <text>Works the direction indicator?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="direction_indicator_yes" event-decision="0">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="direction_indicator_no" event-decision="1">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="direction_indicator_option" type="required" event-id="4" event-decision="1">
               <text>If not, which do not work?</text>
               <answer type="free-text" mode="text" placeholder="Please insert direction indicator
                              which does not work"/>
            </question>
            <question export-name="warning_lights">
               <text>Works the warning lights?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="warning_lights_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="warning_lights_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="reversing_lights">
               <text>Works the reversing lights?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="reversing_lights_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="reversing_lights_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="license_plate_lights">
               <text>Works the license plates light?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="license_plate_lights_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="license_plate_lights_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="brake_light" event-id="5">
               <text>Works the brake light left and right?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="brake_light_yes" event-decision="0">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="brake_light_no" event-decision="1">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="brake_light_option" event-id="5" event-decision="1">
               <text>If not, on which side it does not work?</text>
               <answer type="choice" minAllowed="1" maxAllowed="2">
                  <option value="brake_light_left">
                     <optionText>Left</optionText>
                  </option>
                  <option value="brake_light_right">
                     <optionText>Right</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="rear_fog_lamp">
               <text>Works the rear fog lamp?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="rear_fog_lamp_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="rear_fog_lamp_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="indicator_lamps" event-id="6">
               <text>Works the instrument lighting including warning lights?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="indicator_lamps_yes" event-decision="0">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="indicator_lamps_no" event-decision="1">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="direction_indicator_option" type="required" event-id="6" event-decision="1">
               <text>If not, what does not work?</text>
               <answer type="free-text" mode="text" placeholder="Please insert what is not working"/>
            </question>
         </questionGroup>
      </page>
      <page>
         <question export-name="horn">
            <text>Works the horn?</text>
            <answer type="choice" maxAllowed="1">
               <option value="horn_yes">
                  <optionText>Yes</optionText>
               </option>
               <option value="horn_no">
                  <optionText>No</optionText>
               </option>
            </answer>
         </question>
         <question export-name="windshield">
            <text>Indicates the windshield damages or impairments in the view of the driver?</text>
            <answer type="choice" maxAllowed="1">
               <option value="windshield_yes">
                  <optionText>Yes</optionText>
               </option>
               <option value="windshield_no">
                  <optionText>No</optionText>
               </option>
            </answer>
         </question>
         <question export-name="wiper" event-id="7">
            <text>Works the windshield wiper and wiper?</text>
            <answer type="choice" maxAllowed="1">
               <option value="wiper_yes" event-decision="0">
                  <optionText>Yes</optionText>
               </option>
               <option value="wiper_no" event-decision="1">
                  <optionText>No</optionText>
               </option>
            </answer>
         </question>
         <question export-name="wiper_option" type="required" event-id="7" event-decision="1">
            <text>If not, what does not work?</text>
            <answer type="free-text" mode="text" placeholder="Bitte geben Sie an, was nicht funktioniert"/>
         </question>
         <question export-name="mirror" event-id="8">
            <text>Are fissures in the inside and outside mirrors?</text>
            <answer type="choice" maxAllowed="1">
               <option value="mirror_yes" event-decision="0">
                  <optionText>Yes</optionText>
               </option>
               <option value="mirror_no" event-decision="1">
                  <optionText>No</optionText>
               </option>
            </answer>
         </question>
         <question export-name="mirror_option" event-id="8" event-decision="0">
            <text>If yes, where?</text>
            <answer type="choice" minAllowed="1" maxAllowed="2">
               <option value="mirror_inside">
                  <optionText>Inside mirror</optionText>
               </option>
               <option value="mirror_exterior">
                  <optionText>Outside mirror</optionText>
               </option>
            </answer>
         </question>
         <question export-name="motor_oil">
            <text>Emerges oil from the engine?</text>
            <answer type="choice" maxAllowed="1">
               <option value="motor_oil_yes">
                  <optionText>Yes</optionText>
               </option>
               <option value="motor_oil_no">
                  <optionText>No</optionText>
               </option>
            </answer>
         </question>
         <question export-name="car_body_rust" event-id="9">
            <text>Has the vehicle body rust defects?</text>
            <answer type="choice" maxAllowed="1">
               <option value="car_body_rust_yes" event-decision="0">
                  <optionText>Yes</optionText>
               </option>
               <option value="car_body_rust_no" event-decision="1">
                  <optionText>No</optionText>
               </option>
            </answer>
         </question>
         <question export-name="car_body_rust_option" event-id="9" event-decision="0">
            <text>If yes, please specify</text>
            <answer type="choice" minAllowed="0" maxAllowed="4">
               <option value="vehicle_ground">
                  <optionText>Vehicle ground</optionText>
               </option>
               <option value="vehicle_door">
                  <optionText>Vehicle door</optionText>
               </option>
               <option value="trunk_ground">
```

```xml
                  <optionText>Trunk floor</optionText>
               </option>
               <option value="wheel_case">
                  <optionText>Wheel case</optionText>
               </option>
            </answer>
         </question>
         <question export-name="wiper_option" type="required" event-id="9" event-decision="0">
            <text>Other rusty areas:</text>
            <answer type="free-text" mode="text" placeholder="If nothing from above applies,
                        please insert rusty areas"/>
         </question>
      </page>
      <page>
         <questionGroup>
            <question export-name="steering" type="intro">
               <text>Steering</text>
            </question>
            <question export-name="steering_test" event-id="10">
               <text>Is the steering play alright?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="steering_test_yes" event-decision="0">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="steering_test_no" event-decision="1">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="steering_test_option" type="required" event-id="10" event-decision="1">
               <text>If not, what is not alright?</text>
               <answer type="free-text" mode="text" placeholder="Please insert what is not alright"/>
            </question>
            <question export-name="steering_sound">
               <text>Occure special steering noises?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="steering_sound_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="steering_sound_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
         </questionGroup>
         <questionGroup>
            <question export-name="wheels" type="intro">
               <text>Wheels</text>
            </question>
            <question export-name="wheel_size">
               <text>Match rims and wheels sizes information to the vehicle papers?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="wheel_size_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="wheel_size_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="skid_depth">
               <text>Is the skid depth more then 1.6 mm?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="skid_depth_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="skid_depth_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="wheel_abrasion">
               <text>Are the wheels worn on one side</text>
               <answer type="choice" maxAllowed="1">
                  <option value="wheel_abrasion_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="wheel_abrasion_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="wheel_defects" event-id="11">
               <text>Are there any external damages to the wheels and rims?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="wheel_defects_yes" event-decision="0">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="wheel_defects_no" event-decision="1">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="wheel_defects_option" type="required" event-id="11" event-decision="0">
               <text>If yes, which one?</text>
               <answer type="free-text" mode="text" placeholder="Please insert damages"/>
            </question>
            <question export-name="air_pressure">
               <text>Is the air pressure correct?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="air_pressure_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="air_pressure_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
         </questionGroup>
      </page>
      <page>
         <questionGroup>
            <question export-name="breaks" type="intro">
               <text>Brakes</text>
            </question>
            <question export-name="break_squeak">
               <text>Screeching the brakes?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="break_squeak_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="break_squeak_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="break_site">
               <text>Does the vehicle moves on one side during braking?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="break_site_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="break_site_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="break_fluid">
               <text>Is the brake fluid correct?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="break_fluid_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="break_fluid_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
         </questionGroup>
         <questionGroup>
            <question export-name="flue_gas_system" type="intro">
               <text>Flue gas system</text>
            </question>
            <question export-name="exhaust_attachement">
               <text>Is the exhaust attachement correctly?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="exhaust_attachement_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="exhaust_attachement_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
            <question export-name="exhaust_sounds">
               <text>Are loud noises to hear at the exhaust?</text>
               <answer type="choice" maxAllowed="1">
                  <option value="exhaust_sounds_yes">
                     <optionText>Yes</optionText>
                  </option>
                  <option value="exhaust_sounds_no">
                     <optionText>No</optionText>
                  </option>
               </answer>
            </question>
         </questionGroup>
         <questionGroup>
            <question export-name="utensils" type="intro">
               <text>Utensils</text>
            </question>
            <question export-name="vehicle_utensils">
               <text>Is the following equipment located in the vehicle (Please mark)</text>
               <answer type="choice" minAllowed="0" maxAllowed="3">
                  <option value="warning_triangle">
                     <optionText>Warning triangle</optionText>
                  </option>
                  <option value="first_aid_kit">
                     <optionText>First aid kit</optionText>
                  </option>
                  <option value="safety_vest">
                     <optionText>Safety vest</optionText>
                  </option>
               </answer>
            </question>
         </questionGroup>
         <question export-name="feasibility">
            <text>Was the preliminary check easy to perfom?</text>
            <answer type="scale">
               <option value="0">
                  <optionText>easy</optionText>
               </option>
               <option value="1">
                  <optionText>feasible</optionText>
               </option>
               <option value="2">
                  <optionText>is ok</optionText>
               </option>
               <option value="3">
                  <optionText>difficult</optionText>
               </option>
               <option value="4">
                  <optionText>without help not feasible</optionText>
               </option>
            </answer>
         </question>
      </page>
      <page>
         <information>
            <text><![CDATA[<style type="text/css">body {background-color:#FFFEE5;}</style>Thank you for the enactment
                        of the preliminary check for the general inspection of you vehicle]]></text>
         </information>
      </page>
   </language>
</survey>
```

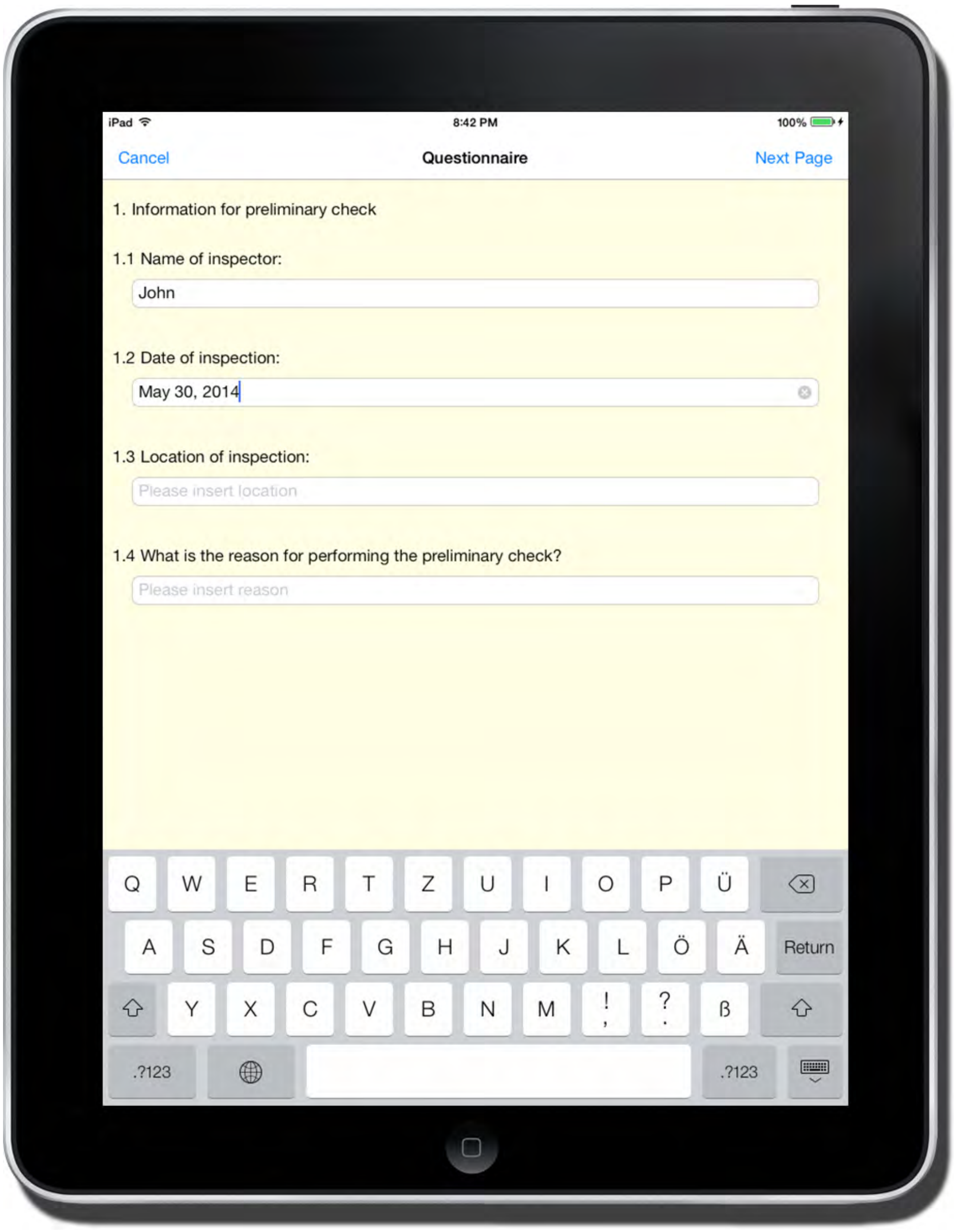
Listing A.2: XML Document for Preliminary Check

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="survey">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="language"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="language">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="page"/>
      </xs:sequence>
      <xs:attribute name="lang" use="required" type="xs:NCName"/>
      <xs:attribute name="localized-survey-name" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="page">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="information"/>
        <xs:choice maxOccurs="unbounded">
          <xs:element ref="question"/>
          <xs:element ref="questionGroup"/>
        </xs:choice>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="information" type="text"/>
  <xs:element name="questionGroup">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="question"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="text">
    <xs:sequence>
      <xs:element ref="text"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="text" type="xs:string"/>
  <xs:element name="question">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="text">
          <xs:sequence>
            <xs:element minOccurs="0" ref="answer"/>
          </xs:sequence>
          <xs:attribute name="event-decision" type="xs:integer"/>
          <xs:attribute name="event-id" type="xs:integer"/>
          <xs:attribute name="export-name" use="required" type="xs:NCName"/>
          <xs:attribute name="type" type="xs:NCName"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="answer">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="option"/>
      </xs:sequence>
      <xs:attribute name="maxAllowed" type="xs:integer"/>
      <xs:attribute name="minAllowed" type="xs:integer"/>
      <xs:attribute name="mode" type="xs:NCName"/>
      <xs:attribute name="placeholder"/>
      <xs:attribute name="type" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="option">
    <xs:complexType>
```

```
      <xs:sequence>
        <xs:element ref="optionText"/>
      </xs:sequence>
      <xs:attribute name="event-decision" type="xs:integer"/>
      <xs:attribute name="value" use="required" type="xs:NMTOKEN"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="optionText" type="xs:string"/>
</xs:schema>
```

Listing A.3: XSD for Validation of the XML Document

User Interfaces for Preliminary Check (Chapter 6):



1. Information for preliminary check

1.1 Name of inspector:

John

1.2 Date of inspection:

May 30, 2014

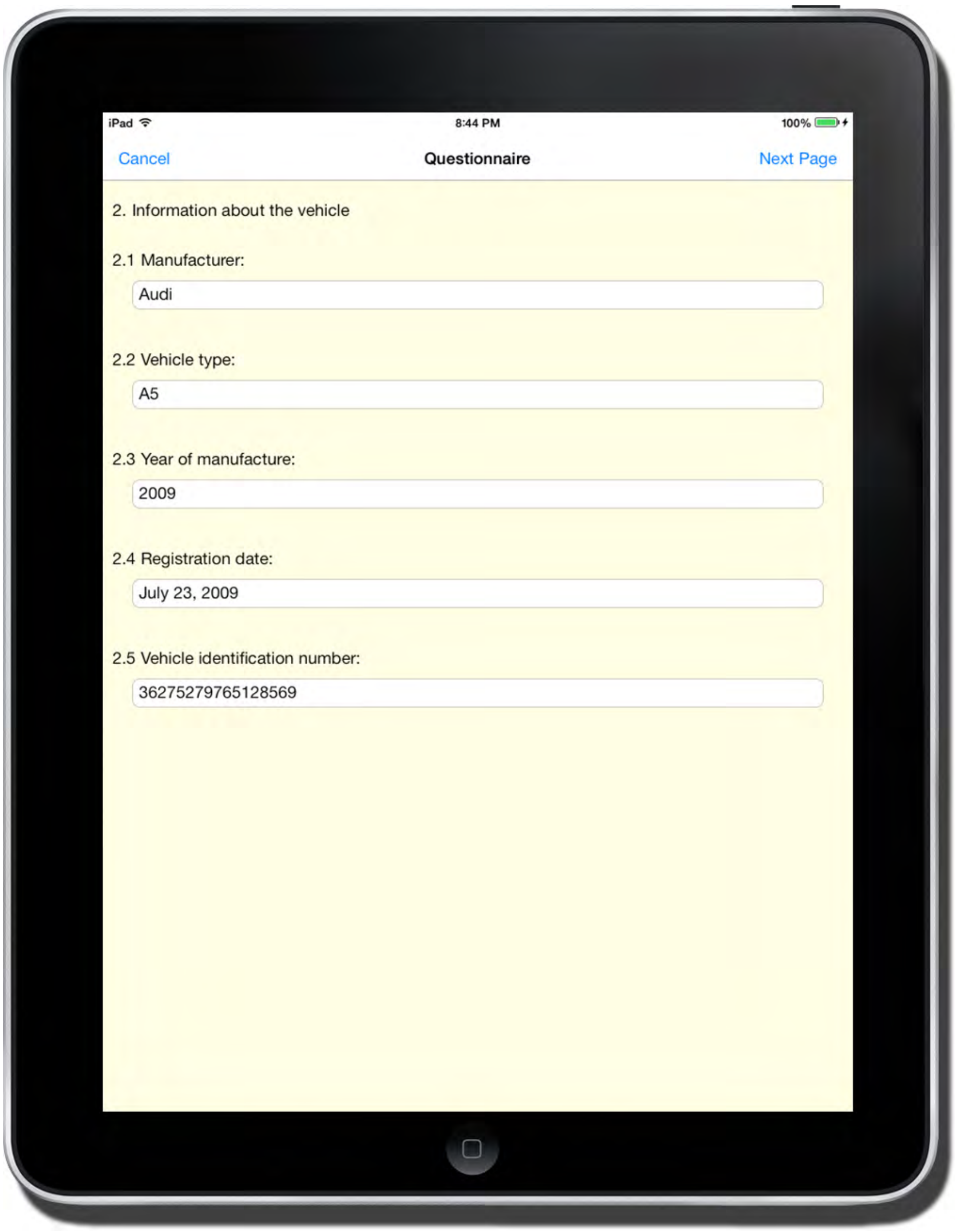
1.3 Location of inspection:

Please insert location

1.4 What is the reason for performing the preliminary check?

Please insert reason

Figure A.14.: General Information



2. Information about the vehicle

2.1 Manufacturer:

Audi

2.2 Vehicle type:

A5

2.3 Year of manufacture:

2009

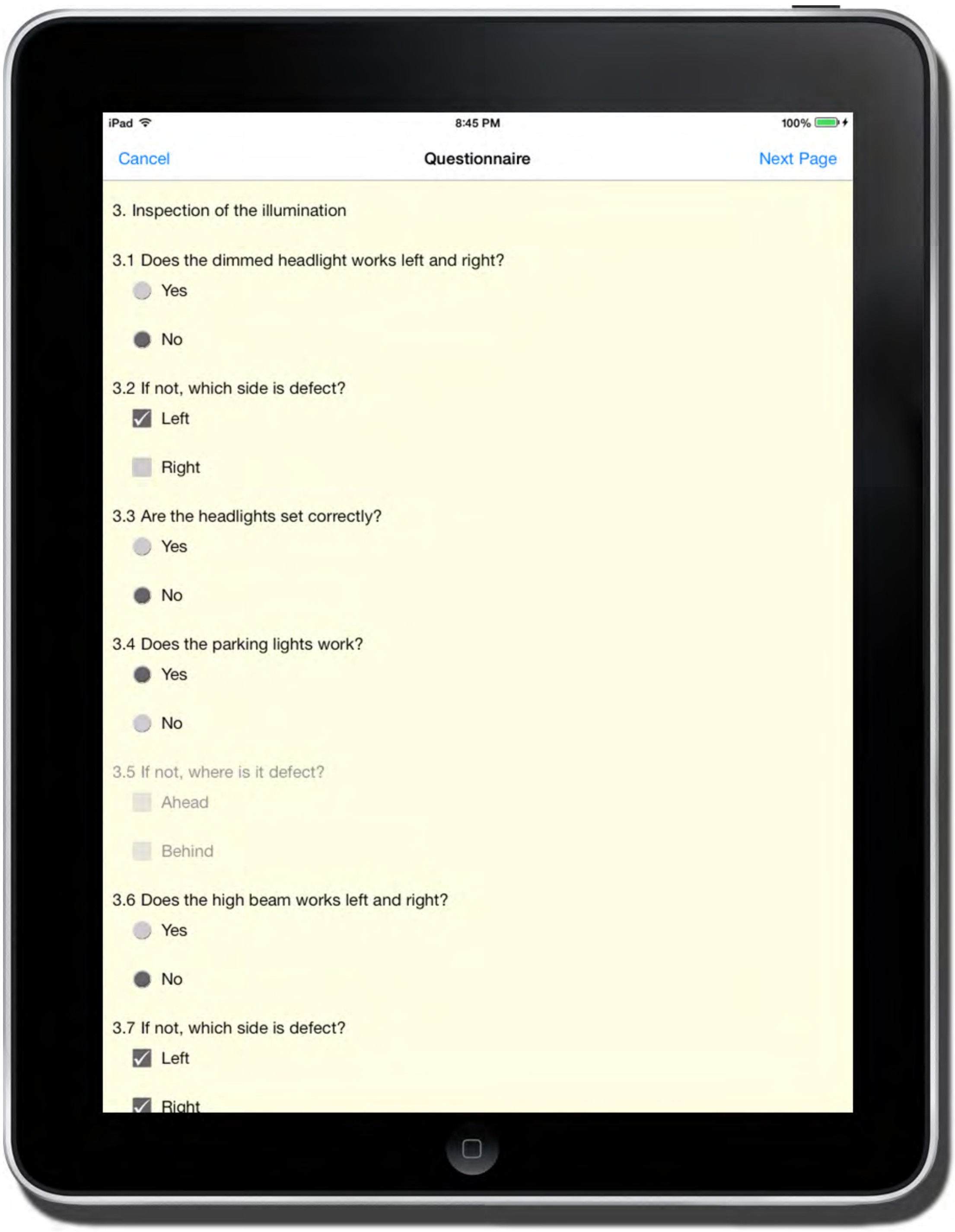
2.4 Registration date:

July 23, 2009

2.5 Vehicle identification number:

36275279765128569

Figure A.15.: Information about the Vehicle



iPad 8:45 PM 100%

Cancel **Questionnaire** Next Page

3. Inspection of the illumination

3.1 Does the dimmed headlight works left and right?

Yes [✓ No]

3.2 If not, which side is defect?

[✓ Left] Right

3.3 Are the headlights set correctly?

Yes [✓ No]

3.4 Does the parking lights work?

[✓ Yes] No

3.5 If not, where is it defect?

Ahead Behind

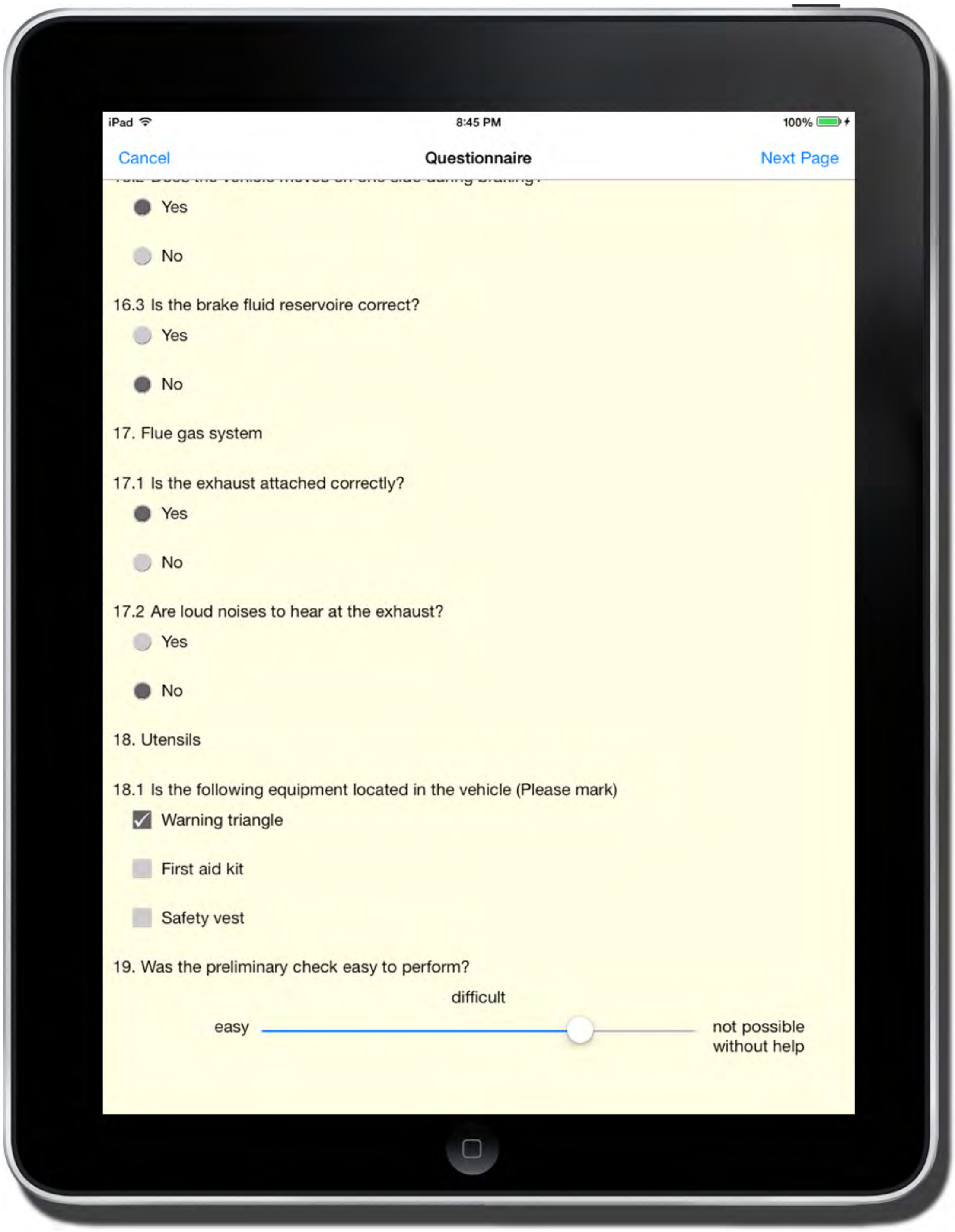
3.6 Does the high beam works left and right?

Yes [✓ No]

3.7 If not, which side is defect?

[✓ Left] [✓ Right]

Figure A.16.: Impressions of the Preliminary Check Performance (1)



[✓ Yes] No

16.3 Is the brake fluid reservoire correct?

Yes [✓ No]

17. Flue gas system

17.1 Is the exhaust attached correctly?

[✓ Yes] No

17.2 Are loud noises to hear at the exhaust?

Yes [✓ No]

18. Utensils

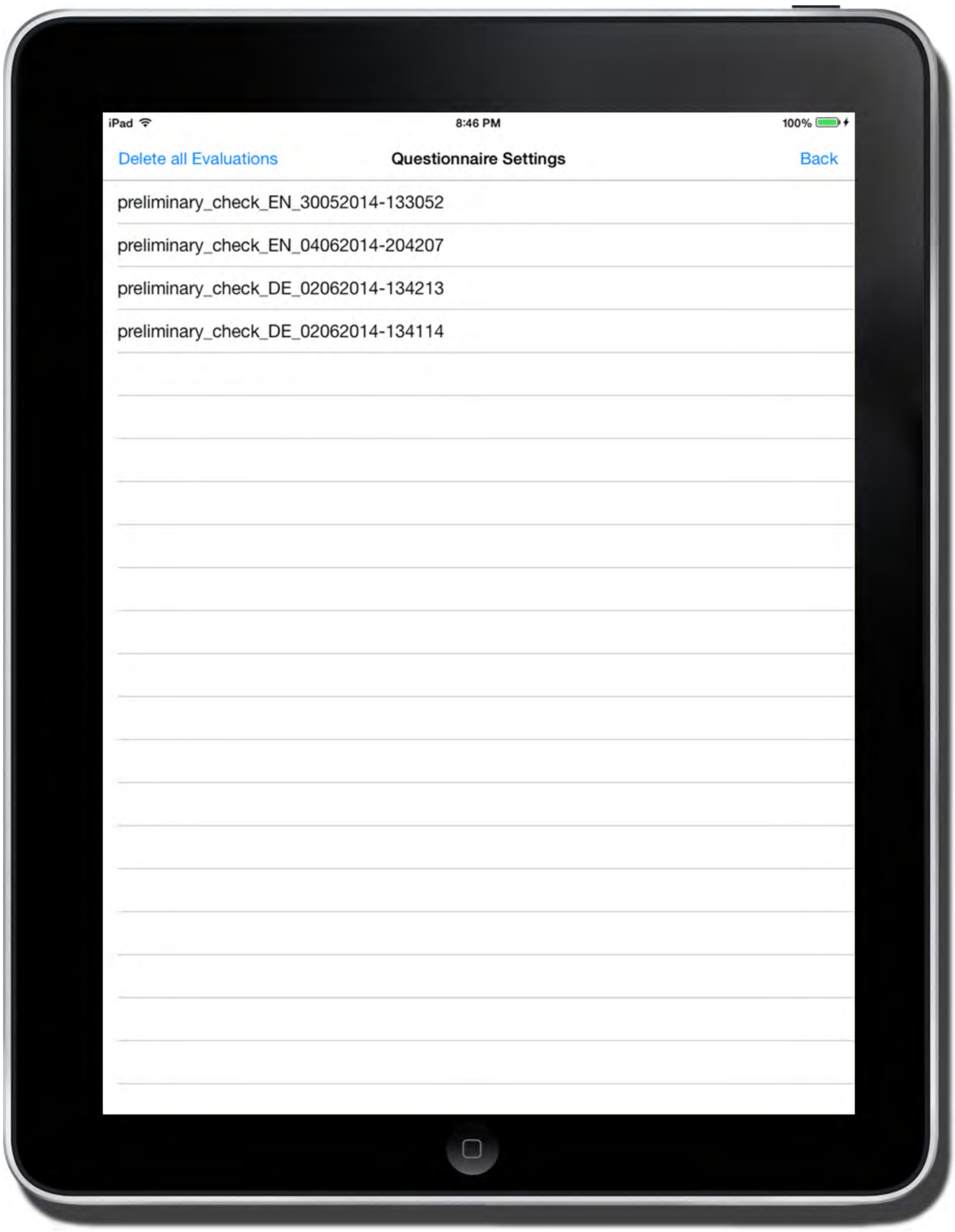
18.1 Is the following equipment located in the vehicle (Please mark)

- [✓ Warning triangle]
- First aid kit
- Safety vest

19. Was the preliminary check easy to perform?

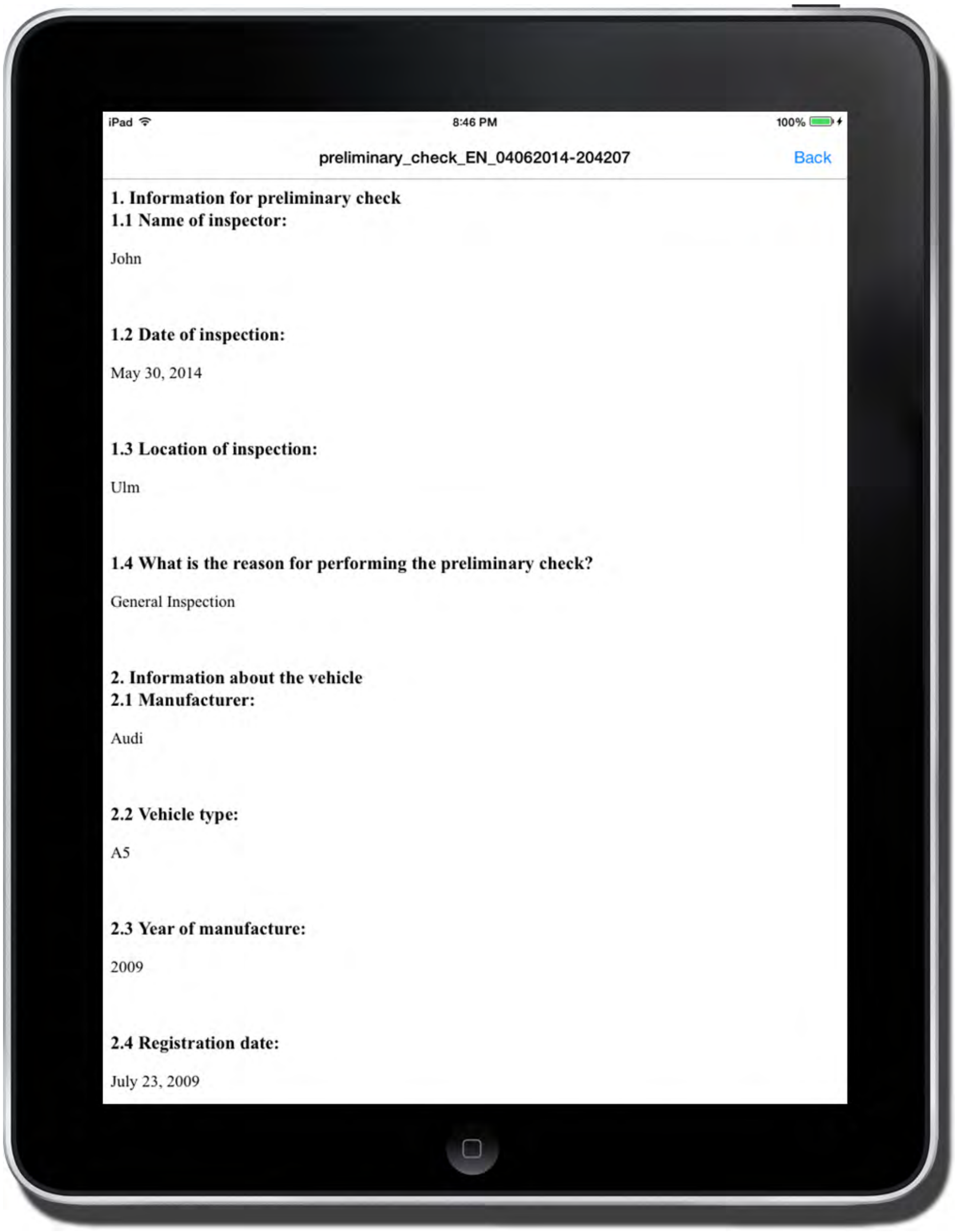
easy — difficult — not possible without help

Figure A.17.: Impressions of the Preliminary Check Performance (2)



preliminary_check_EN_30052014-133052
preliminary_check_EN_04062014-204207
preliminary_check_DE_02062014-134213
preliminary_check_DE_02062014-134114

Figure A.18.: Selection of Results



preliminary_check_EN_04062014-204207

1. Information for preliminary check
1.1 Name of inspector:

John

1.2 Date of inspection:

May 30, 2014

1.3 Location of inspection:

Ulm

1.4 What is the reason for performing the preliminary check?

General Inspection

2. Information about the vehicle
2.1 Manufacturer:

Audi

2.2 Vehicle type:

A5

2.3 Year of manufacture:

2009

2.4 Registration date:

July 23, 2009

Figure A.19.: Viewing of Results

List of Figures

List of Tables

Listings

Bibliography

[AP05] Axelle Apvrille and Makan Pourzandi. Secure Software Development by Example. *IEEE Security and Privacy*, 3(4):10–17, July 2005.

[App14] Apple. iOS Human Interface Guidelines. Technical report, Apple Inc., 2014. Accessed: 2014-05-16. URL: `https://developer.apple.com/library/iOs/documentation/UserExperience/Conceptual/MobileHIG/Principles.html`.

[Bal14] Balsamiq Mockups - Balsamiq Studios, 2014. Accessed: 2014-05-28. URL: `http://www.balsamiq.com/`.

[BO$^+$08] Antonio Bernabe-Ortiz et al. Handheld computers for self-administered sensitive data collection: A comparative study in Peru. *BMC Medical Informatics and Decision Making*, 8(1):11, 2008.

[box14] boxcryptor - AES Encryption, 2014. Accessed: 2014-05-24. URL: `https://www.boxcryptor.com/en/encryption`.

[Bru12] Simon Bruns. Tablet-based survey instrument. Bachelor Thesis, 2012. URL: `http://learntech.rwth-aachen.de/Bruns`.

[BS93] Victoria Bellotti and Abigail Sellen. Design for privacy in ubiquitous computing environments. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW'93*, pages 77–92. Springer, 1993.

[BSM96] T. Bray and C.M. Sperberg-McQueen. Extensible Markup Language (XML). World Wide Web Consortium - W3C Working Draft, 1996. Accessed: 2014-05-01. URL: `http://www.w3.org/TR/WD-xml-961114.html`.

[Coc12] Cocoa Touch Framework, 2012. Apple Inc. - Accessed: 2014-05-03. URL: `https://developer.apple.com/technologies/ios/cocoa-touch.html`.

[Dek14] Preliminary Checklist - DEKRA, 2014. Accessed: 2014-05-28. URL: `http://www.dekra.de/de/c/document_library/get_file?uuid=cdf485c2-f41d-4390-9a6d-401e62724530&groupId=10100`.

[DGF$^+$95] H.E. Drummond, S. Ghosh, A. Ferguson, D. Brackenridge, and B. Tiplady. Electronic quality of life questionnaires: a comparison of pen-based electronic questionnaires with conventional paper in a gastrointestinal study. *Quality of Life Research*, 4(1):21–26, 1995.

[DIN03] DIN. *Software engineering - Product quality*. DIN Deutsches Institut für Normung e. V., 2003.

[FAQ12] Joe Fawcett, Danny Ayers, and Liam R. E. Quin. *Beginning XML*. Wrox Press Ltd., Birmingham, UK, 5th edition, 2012. ISBN: 1-118-16213-7, 978-1-11816-213-2.

[For14] Formaestudio.com - AES Encryption, 2014. Accessed: 2014-05-24. URL: `http://www.formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng.swf`.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[Gar] Gartner. Statistic - Market Share of Sold Tablets (Worldwide). Source: Gartner - Accessed: 2014-05-09. URL: `http://www.gartner.com/newsroom/id/2674215`.

[GP08] Fosca Giannotti and Dino Pedreschi. *Mobility, data mining and privacy: Geographic knowledge discovery*. Springer, 2008. ISBN: 978-3-540-75177-9.

[GR11] Mark H. Goadrich and Michael P. Rogers. Smart Smartphone Development: iOS Versus Android. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, SIGCSE '11, pages 607–612, New York, NY, USA, 2011. ACM. ISBN: 978-1-4503-0500-6.

[GWSB03] David P. Gilliam, T.L. Wolfe, J.S. Sherif, and M. Bishop. Software security checklist for the software life cycle. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. 12th IEEE International Workshops on*, pages 243–248, June 2003.

[HK00] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000. ISBN: 1-55860-489-8.

[HKS02] Matthias Hansch, Stefan Kuhlins, and Martin Schader. XML-Schema. *Informatik-Spektrum*, 25(5):363–366, 2002.

[HR05] Simon Harris and James Ross. *Beginning Algorithms (Understanding Big O-Notation)*. John Wiley & Sons, 2005. ISBN: 978-0-7645-9674-2.

[HW79] John A Hartigan and Manchek A Wong. Algorithm AS 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[Joh67] Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[KA09] Atul Kahate and Kahate Atul. *XML & Related Technologies*. Pearson Education India, 2009. ISBN: 8-131-71865-4, 978-8-13171-865-0.

[Ker03] Sascha Kersten. *Kompendium der Informationstechnik,15.2 DTDs und XML Schema*. Galileo Press, Bonn, 2003. ISBN: 978-3-89842-355-7. URL: `http://www.galileocomputing.de/openbook/kit/itkomp07000.htm`.

[Kle11] S. Kleuker. *Grundkurs Software-Engineering mit UML - Der pragmatische Weg zu erfolgreichen Softwareprojekten*. Vieweg + Teubner Verlag, 2 edition, 2011. Chapter 4, ISBN: 978-3-8348-1417-3.

[Lan01] Marc Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In *Ubicomp 2001: Ubiquitous Computing*, pages 273–291. Springer, 2001.

[Man02] C. Mann. Why software is so bad... and what's being done to fix it. *MIT Rechnology Review*, 105:33–38, July 2002.

[Mar13] Marc. Working with user interface files in oxygene "nougat", 2013. Accessed: 2014-05-20. URL: `http://blogs.remobjects.com/blogs/mh/2013/01/02/p5365`.

[Nie93] Jakob Nielsen. *Usability Engineering:25-26*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN: 0-125-18405-0.

[Nie12] J. Nielsen. Usability 101: Introduction to Usability. *Jakob Nielsen's Alertbox*, 2012. URL: `http://www.nngroup.com/articles/usability-101-introduction-to-usability/`.

[Obj12] Objective-C Documentation, 2012. Apple Inc., Accessed: 2014-05-03. URL: `https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html`.

[Off12] M. Offergeld, 2011/2012. Reference Modell of Software Development in Usability Engineering - Lecture Script; Chapter Grundmodell, University of Ulm.

[oxy14] oxygen xml - syncro soft srl, 2014. Accessed: 2014-05-22. URL: `http://www.oxygenxml.com/`.

[pad] padcapi. Hopp und Partner, Accessed: 2014-05-21. URL: `http://www.hopp-und-partner.de`.

[PMGC81] John P. McIver and Edward G. Carmines. *Unidimensional Scaling*. Sage Publications, 1981. ISBN: 978-0-803-91736-1.

[PtJ14] Prototyper - justinmind, 2014. Accessed: 2014-05-17. URL: `http://www.justinmind.com/`.

[Ray03] Erik T. Ray. *Learning XML:51-58*. O'Reilly & Associates, Sebastopol, California, 2nd edition, September 2003. ISBN: 0-596-00420-6.

[RJ07] Janice Rattray and Martyn C Jones. Essential elements of questionnaire design and development. *Journal of clinical nursing*, 16(2):234–243, 2007.

[sbo14] S-box, 2014. Accessed: 2014-05-25. URL: `https://edipermadi.files.wordpress.com/2008/03/sbox.png`.

[SCCP02] Everett V. Smith, Karen M. Conrad, Karen Chang, and Jo Piazza. An introduction to rasch measurement for scale development and person assessment. *Journal of Nursing Measurement*, 10(3):189–206, 2002.

[Sch03] Jochen H Schiller. *Mobile communications*. Pearson Education, 2003. ISBN: 0-321-12381-6.

[Sch05] Mischa Schwartz. *Mobile wireless communications*. Cambridge University Press, 2005. ISBN: 0-521-84347-2.

[Sch14] Steffen Scherle. Konzeption und Evaluierung einer domänenspezifischen Modellierungsumgebung für prozessorientierte Fragebögen. Diploma Thesis. University of Ulm, 2014.

[Shi] J. Shirrell. XML: A deeper Understanding, Chapter 4.1 XML. Accessed: 2014-05-01. URL: `www.xmlbook.info`.

[Shn03] Ben Shneiderman. *Designing the user interface*. Pearson Education India, 2003.

[sna14] Snap Surveys, Your Perfect Survey Partner (PDF), 2014. Accessed: 2014-07-03. URL: `http://www.snapsurveys.com`.

[Som10] Ian Sommerville. *Software Engineering:295-296*. Addison-Wesley, Harlow, England, 9 edition, 2010. ISBN: 978-0-13-703515-1.

[SSP$^+$13] Johannes Schobel, Marc Schickler, Rüdiger Pryss, Hans Nienhaus, and Manfred Reichert. Using vital sensors in mobile healthcare business applications: Challenges, examples, lessons learned. In *9th Int'l Conf. Web Inf. Sys. and Techn. (WEBIST 2013)*, pages 509–518, May 2013.

[sur] Surveypocket. Accessed: 2014-05-21. URL: `http://www.surveypocket.com`.

[Tid10] Jenifer Tidwell. *Designing interfaces*. " O'Reilly Media, Inc.", 2010. ISBN: 978-0-59-600803-1.

[VS03] Wijbrandt H Van Schuur. Mokken scale analysis: Between the Guttman scale and parametric item response theory. *Political Analysis*, 11(2):139–163, 2003.

[Was] Shneiderman's eight golden rules of interface design. Accessed: 2014-05-16. URL: `http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html`.

[Web13] Michael Weber, 2013. Ubiquitous Computing - Lecture Script; Chapter 2 Mobile Communication, University of Ulm.

[Wie07] Gerhard Wiehler. *Mobility, security and Web services: technologies and service-oriented architectures for a new era of IT solutions: 9 - 10*. John Wiley & Sons, 2007. ISBN: 978-3-89-578228-2.

[Wil04] Christian Wilkin. Der Algorithmus des Advanced Encrypton Standard. Trier University of Applied Sciences, 2004.

[YS95] Yufei Yuan and Michael J Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and systems*, 69(2):125–139, 1995.

Name: Dominik Deuter Matriculation Number: 708686

Statutory declaration

I hereby declare that I wrote the master's thesis independently and used no other aids that those cited.

Ulm, ...

Dominik Deuter