



MoBiLe App: Conception and Realisation of Mobile Serious Games for learning support in biochemistry with the Android operating system

Bachelor's Thesis

Annalisa Degenhard

December 2019

Examiner: Prof.Dr. Manfred Reichert

Supervisor: Prof.Dr. Rüdiger Pryss

Supervisor: PD Dr. Susanne Kühl

Institute of Databases and Information Systems
Faculty of Engineering, Computer Science and Psychology
Ulm University

Declaration

I declare that I performed the presented research thesis

**MoBiLe App: Conception and Realisation of Mobile Serious Games for
learning support in biochemistry with the Android operating system**

myself and that I wrote the thesis myself. Any additional sources of information have been duly identified and referenced. Further, I have regarded the rules of scientific work of the Ulm University.

Ulm, Friday 13th December, 2019

Annalisa Degenhard

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem	1
1.3	Objective	1
1.4	Overview	2
2	Overall description	3
2.1	Definitions, acronyms and abbreviations	3
2.2	Product perspective	4
2.3	Product functions	5
2.3.1	Necessary requirements	5
2.3.2	Additional requirements	6
2.3.3	Optional requirements	6
2.4	Functionality overview	6
2.4.1	Application start	8
2.4.2	Testing	9
2.4.3	Abort and recover a test session	10
2.4.4	Statistics	11
2.4.5	Functionalities from the angle of gamification	12
2.5	Target group	13
2.6	User types	16
2.7	Context of use	16
2.8	Design and implementation constraints	17
2.9	User documentation	17
3	System requirements	19
3.1	Functional system requirements	19
3.1.1	Application: general	19
3.1.2	Application: Features	20
3.2	Non-functional System requirements	28
4	Software specifications	31
4.1	System interfaces	31
4.1.1	Dialog structure	31
4.1.2	Mockups	33
4.1.3	Code architecture	57
5	Conclusion	57

6 Vision	58
.1 Glossary	60
.2 Class diagrams	62
.3 Use case diagram	65
.4 Dialog structure	67
List of Figures	69
List of Tables	71
Bibliography	73

1 Introduction

1.1 Motivation

Productive learning is a difficult task as there are numerous factors, which can lead to a lack of efficiency or effectiveness. A not uncommon problem is to learn the wrong learning content. This effects not only a massive lack in effectiveness, but also causes frustration of the learner. The students of biochemistry at the university of Ulm have struggled increasingly with this problem. Their studies include three main exams, where each of them tests a certain part of a set of above five thousand questions. But the identification of the right set causes uncertainties. As a result, students have increasingly learned the whole set of questions to ensure having learned all relevant content. But this led to worse examination results and a high frustration level. The goal of this project is, to develop an application which helps the students to learn the right questions, by providing only the relevant ones for the pending exam. Furthermore, the students shall be motivated through gamification elements as achievements and experience points in combination with reminders, to learn in a more constant manner.

1.2 Problem

The challenge is, to find a mobile solution, which hasn't any restrictions in comparison to other methods and also provides functionalities that others cannot offer. Otherwise, the system won't be adapted by the students. Therefore, a detailed analysis of their needs and on the other hand a consideration and comparison of possible application components and features to fit these needs is necessary to achieve the purpose of the application. Furthermore, the design should please as many students as possible without having restrictions in functionality, as the users should be able to learn in an effective but comfortable way.

1.3 Objective

The goal is to develop an application which adds value to the students, facilitating the preparation by substituting existing learning methods. Therefore, necessary filters for the questions will be used, to fix the issue of selecting the right questions. To achieve the best possible substitute for other learning methods, targeted learning shall be enabled through different test types which allow the user to create individual tests. In addition to this, the integration of questions about basic knowledge of the different topics will be provided, to support a deeper understanding of the content. Furthermore, the user shall be able to track his progress of preparation at any time by using progress overviews and experience points. This will allow a more anticipatory learning manner. To ensure constant practicing, training goals in combination with rewards for keeping

on track and reminders will be used. Achievements will complete the range of gamification motivators by adding additional incentives to learn. If this succeeds, the usage of gamification components could increase the willingness to learn and therefore enhance the results of the exams.

1.4 Overview

This document is a detailed analysis of the application MoBiLe separated in three general parts. The first section is a detailed specification of the requirements and goals of the project. This includes the explanation of all related terminologies, the summary of all required functionalities separated into necessary, additional and optional functionalities as well as the quality requirements. The second part is the detailed functional description of the application including mock-ups, use case diagrams, the dialog structure diagram and the class diagram. The conclusion of this work is a analysis of possible extensions for the application in the future.

2 Overall description

The following chapter will give an overview of the requirements, defined from the user's perspective. This includes the analysis of the context of use, user and target groups and a summary of the systems requirements including a prioritization. For further understanding, necessary specialist knowledge will be explained and important terms defined. All services that have to be provided by the new system are precisely documented here.

2.1 Definitions, acronyms and abbreviations

In the following, the relevant terms of the application will be explained in an alphabetical order and will be illustrated by applications.

Term	API
Description	The application programming interface is an interface between client and server used to facilitate the creation of client side software. In this document the mentioned API is the minimum API Version, which is necessary to run the android application code.
Is a	Tool
May be	API 1-27
Aspect	Definition of minimum requirements of the system.
Example	Minimum android API Level is 16.
Term	Gamification
Description	Gamification is the usage of typical game elements in a non-game context.
Is a	Tool
May be	Motivators, avatars
Aspect	Facilitation of learning for exams
Example	Quizzes, rewards
Term	JSON (JavaScript Object Notation)
Description	JSON is a compact file format that uses text to transmit data objects consisting of attribute value pairs and array types.
Is a	File format
May be	Format of question data file.
Aspect	Allows the processing and saving of data through the app.
Example	user data file, question data file

Term	XLSX (Excel file format)
Description	Abbreviation of the format of an excel file.
Is a	File format
May be	Format of question data file.
Aspect	Is supported by the provided data converter for the question data file.
Example	question data file

2.2 Product perspective

MoBiLe helps bachelor students to learn the right content for their pending exam in the right way. By prompting the user to choose the exam he wants to learn for and only providing the corresponding questions in the subsequent use, the app helps the student to prevent from learning the wrong content. In addition to this, MoBiLe has the goal to motivate students to learn in a constant manner. To achieve this, MoBiLe is provided as an application for smartphones and tablets. This allows the user to learn spontaneous and location-independent. To reach as many students as possible, the application will be provided for both, iOS and android. To ensure the quality of both versions, they are developed separately, but have the same design and functional range. This document discusses only the android version. The iOS version is implemented by Nico Brenner [3].

The application was decided to be an offline application, to ensure location-independent availability. As the questions data will still be updated constantly, the application and data will be provided as two separate modules. On the one hand, this architecture facilitates frequent updates and minor correction and on the other hand, prompting the user to download the questions data, will prevent him from learning a deprecated set of questions by using an older version of MoBiLe.

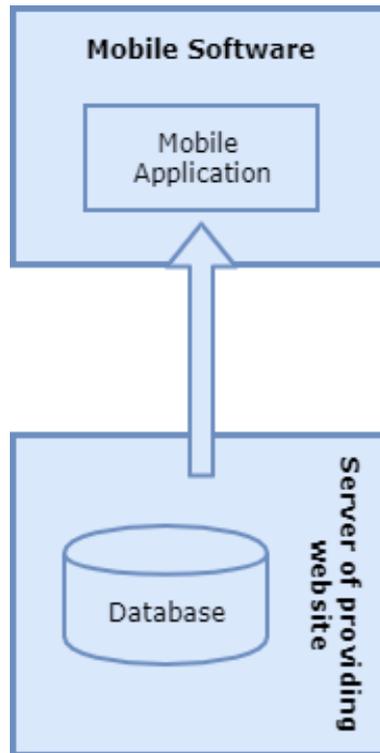


Fig. 2.1: Block architecture

2.3 Product functions

This chapter is a classification of the general requirements in three different types depending on their importance. A distinction is made between requirements which are necessary for fulfilling the purpose of the application, requirements that are certainly considered, but not necessary for the purpose of use, and optional requirements which can be omitted to meet the deadline. All of the requirements will also be explained in detail in chapter 3.

2.3.1 Necessary requirements

To allow flexible learning MoBiLe is realized as a mobile application. It is used to filter the set of all questions based on the pending exam of the user. To ensure this purpose, the set of the desired exam has to be necessary for using the application. Furthermore, MoBiLe has to be able to import a given question data file into the application and display the questions in a correct manner. The user has to be able to create a test of a configurable size, about a selectable topic and difficulty. Afterwards he can answer each question by selecting one of the available responses. MoBiLe then has to be able to give case sensitive feedback. In general, the system has to ensure, that each question will be asked at least once and that the user is able to repeat each question arbitrarily.

2.3.2 Additional requirements

To help the user keep on training in an efficient way, the application has to contain several motivators as well as individually configurable tests. Therefore the user can get achievements by reaching different given goals and view his statistics including the achievements in a corresponding overview. For flexible testing, the system provides four different test types. A classic test about a certain topic and difficulty, a test containing all wrong answered questions of a certain topic and difficulty, a training test consisting of the users marked questions and an exam test which asks as many questions as in a realistic exam with the same time limit. Furthermore, the user has to be able to cancel a test and continue it at a later time.

The user shall be able to view the explanation of a question after giving an answer. His results have to be saved and should be available at any time. In addition to the regular questions, the user can activate basic questions, which are not relevant for his exam, but support a deeper understanding of the learning content. These questions should be marked appropriately.

2.3.3 Optional requirements

An additional tablet version of the application would be desirable. Furthermore a feedback function about question content and using comfort could be integrated, to support a fast improvement of the application. Another useful settings option would be the reset of the user data to start from the beginning. A import support for the data file could also increase the comfort of use. Notifications could be used to remind the user to keep on training.

2.4 Functionality overview

This section gives an overview about the different use scenarios of MoBiLe. All simple interaction possibilities are summarized in use case diagrams. More complex using scenarios as the test procedure are also illustrated as a sequence diagram. The following illustration shows the full range of the application's functionalities. There is also a bigger version of the diagram in the attachments of this document.

2.4.1 Application start

The start of the application is case sensitive. The system checks, if there is already existing user data. If so, the user gets straight into the main view of the application. Otherwise, the application requests the user to set all necessary information for using the it, through the initial set up and allows him to import question data via the download dialog. The following illustration is a summary of the application start scenario.

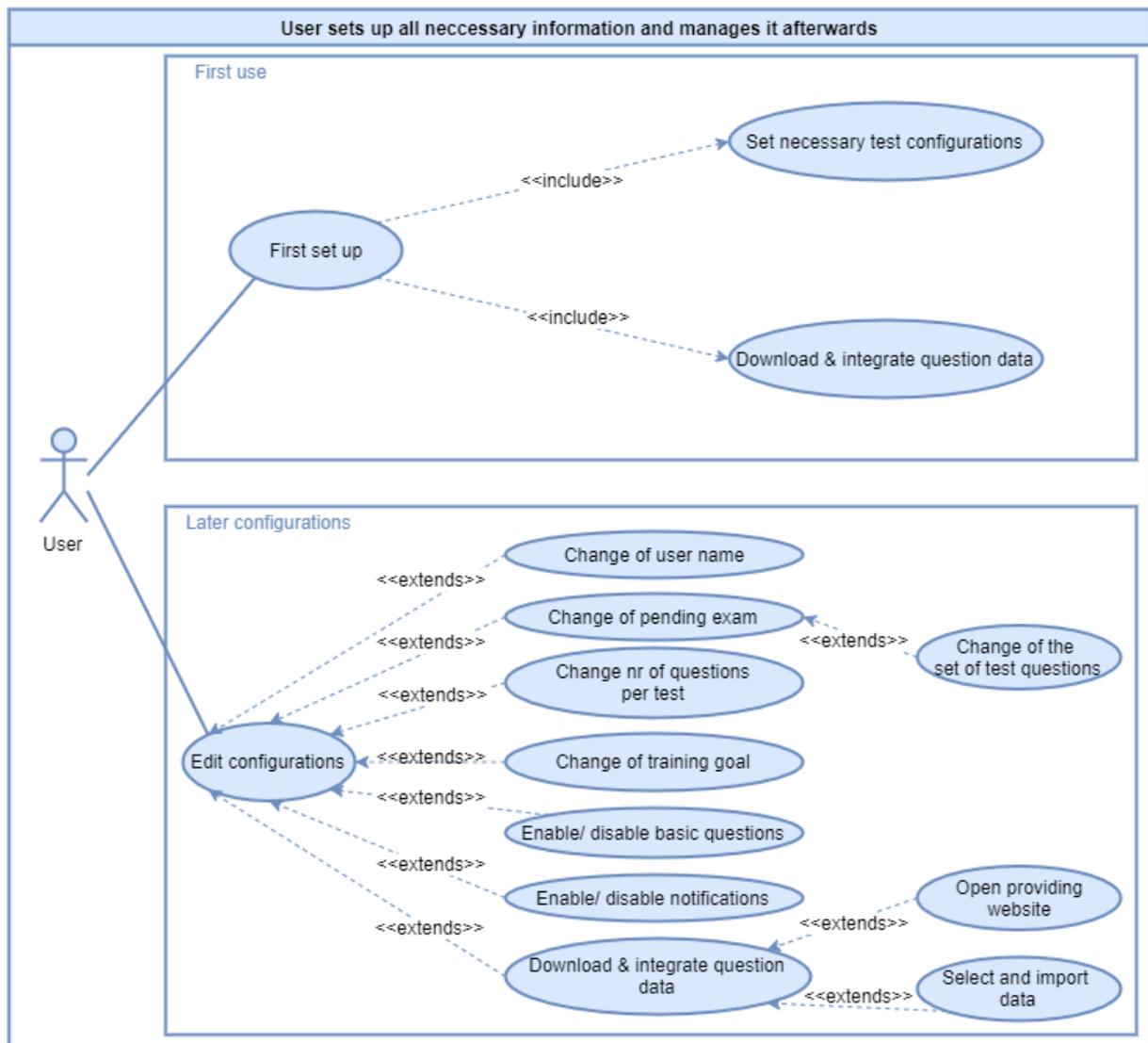


Fig. 2.4: Use case diagram of the application start

2.4.2 Testing

The application allows the user to train the available questions through tests of different kinds. Independent of the type, the underlying test procedure is always the same. The user gets the

question displayed and chooses one of the response possibilities. He can change his selection until he confirms his decision. Afterwards, the system gives the user feedback, based on the given answer. He has now the opportunity to view the explanation and references by clicking on the corresponding explanation button. Furthermore the question can be marked as a training question, so it can be trained at a later time. This procedure gets repeated for every question. After confirming the last question, the system calculates the new progress of the user and shows it in the test summary. If new achievements have been made, the corresponding information dialog will be shown after returning to the main view by clicking on the confirmation button. The following sequence diagram illustrates the described scenario of a test session.

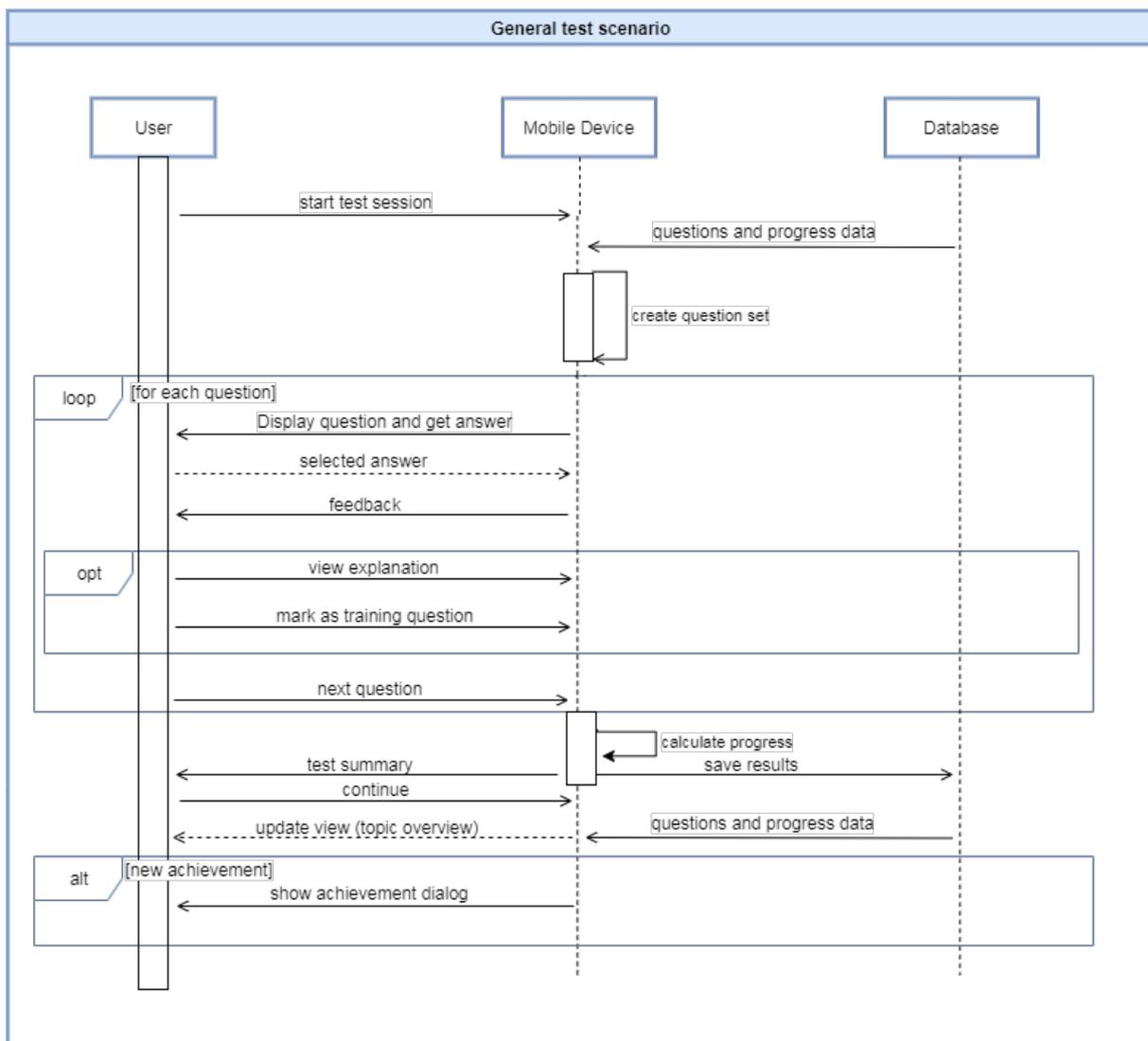


Fig. 2.5: Typical use sequence of a test session

2.4.3 Abort and recover a test session

The user is able to save up to one canceled test session and continue it at a later time. This feature is only supported for classic topic tests and result type tests, as training sessions don't get any reward and exams try to reflect realistic exam conditions as good as possible. The recovery is automatically created when the user tries to leave a test session by clicking on the back button and confirming the corresponding abort request. If the system recognizes an existing recovery file, the user gets the opportunity to choose between a new test and the existing test, every time he enters the main view.

If the user starts a recovery session, the test will automatically continue at the point of cancellation. From now on, the use case relates to the normal test scenario described in the previous section. The only difference is, that in addition to the normal scenario, the recovery file will get deleted after completing the test. The complete recovery scenario looks as follows:

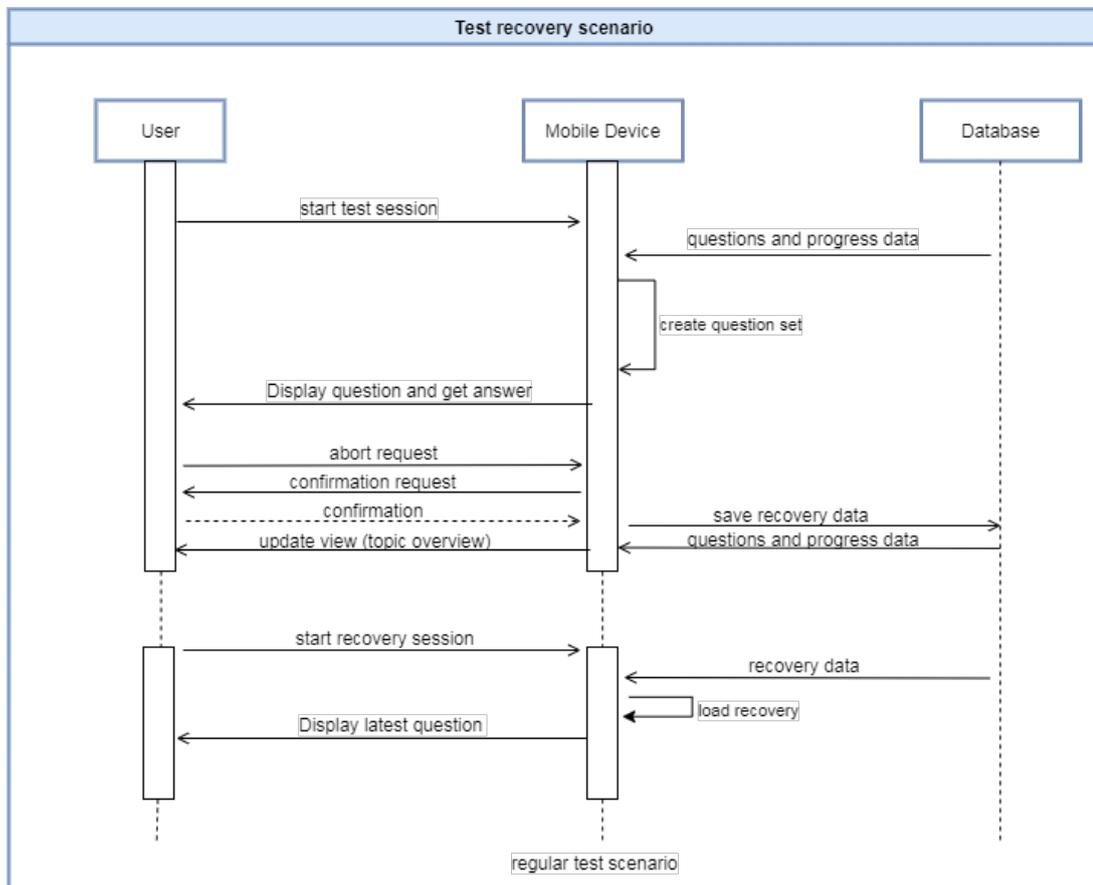


Fig. 2.6: Interaction possibilities with the statistics view.

2.4.4 Statistics

The user has the opportunity to follow his personal progress in the statistics view. It summarizes his training progress over a settable period of time, calculates the users experience and

for how many consecutive days he has reached his training goal. Furthermore, he can view his achievements in the trophy room dialog and check for new achievement goals he can reach. All interaction possibilities are depicted in the following use case diagram.

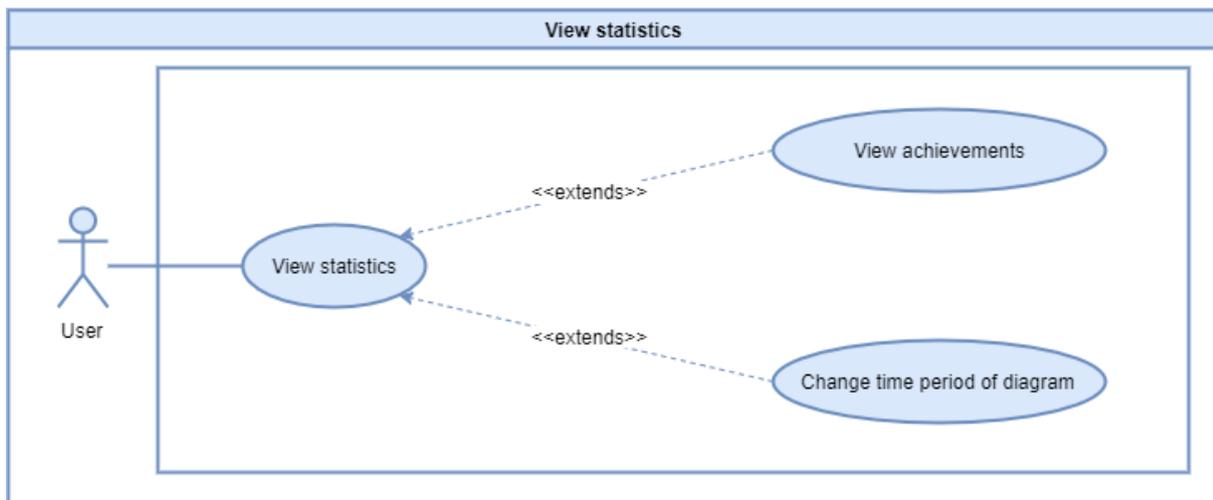


Fig. 2.7: Interaction possibilities with the statistics view.

2.4.5 Functionalities from the angle of gamification

As a result of several discussions with some students and the customers, a set of gamification elements has been chosen. The focus was on offline components which motivate to practice in a non intrusive manner. Furthermore, the students demanded to only integrate a few ones, as their priority was to keep the application clear and simple. To summarize the chosen set of gamification elements, the following classification table of Majuri, Koivisto and Hamari [1], shows which elements are integrated by coloring the corresponding field. The value indicates the frequency of integration in other existing gamification applications.

Achievement/progression		Immersion	
Points, score, XP	67	Avatar, character, virtual identity	15
Challenges, quests, missions, tasks, clear goals	53	Narrative, story telling	13
Badges, achievements, medals, trophies	47	Virtual world, simulation	9
Leaderboards, ranking	47	In-game-rewards	6
Levels	35	Role play	3
Quizzes, questions	25	Non-digital elements	
Progress, status bars, skill trees	19	Check-ins, location data	8
Performance stats/ feedback	18	Real world/financial reward	2
Timer, speed	13	Motion tracking	1
Increasing difficulty	8	Physical objects as game resources	1
Social		Miscellaneous	
Cooperation, teams	31	Assistance, virtual helpers	9
Social networking features	12	Retries, health	6
Peer-rating	10	Full game(also board games)	5
Customization, personalization	3	Adaptive difficulty	3
Multiplayer	2	Game rounds	2
		Onboarding (safe environment to practice)	2
		Reminders, cues, notifications, annotations	2
		Penalties	1

Tab. 2.1: Overview about all used gamification elements according to Majuri et al. [1]

2.5 Target group

The primary target group of the application are bachelor students of biochemistry at the university of Ulm, which are preparing for one of the three exams. Another possible target group are people, who are interested in studying biochemistry and want to get an impression of the study content. The primary target group can be specified through these characteristics:

- Bachelor biochemistry student
- On the first to tenth semester
- German
- Is preparing for a pending exam

In general, there are three possible motives of using the application: having difficulties to identify the right set of questions for the pending exam, having motivational problems or wanting to learn the exam content in a more playful manner. Regarding the second mentioned target

group, another possible motive is to get an overview of the study content.

According to a published paper about gamification [2], the understanding of the user's needs is essential for a successful use of gamification. Therefore, the project included a long concept phase with several meetings, where the requirements and wishes were discussed, as well as mockups and as a last design ideas accessed. The summary of the user needs analysis and the conclusions is illustrated in the following diagram:

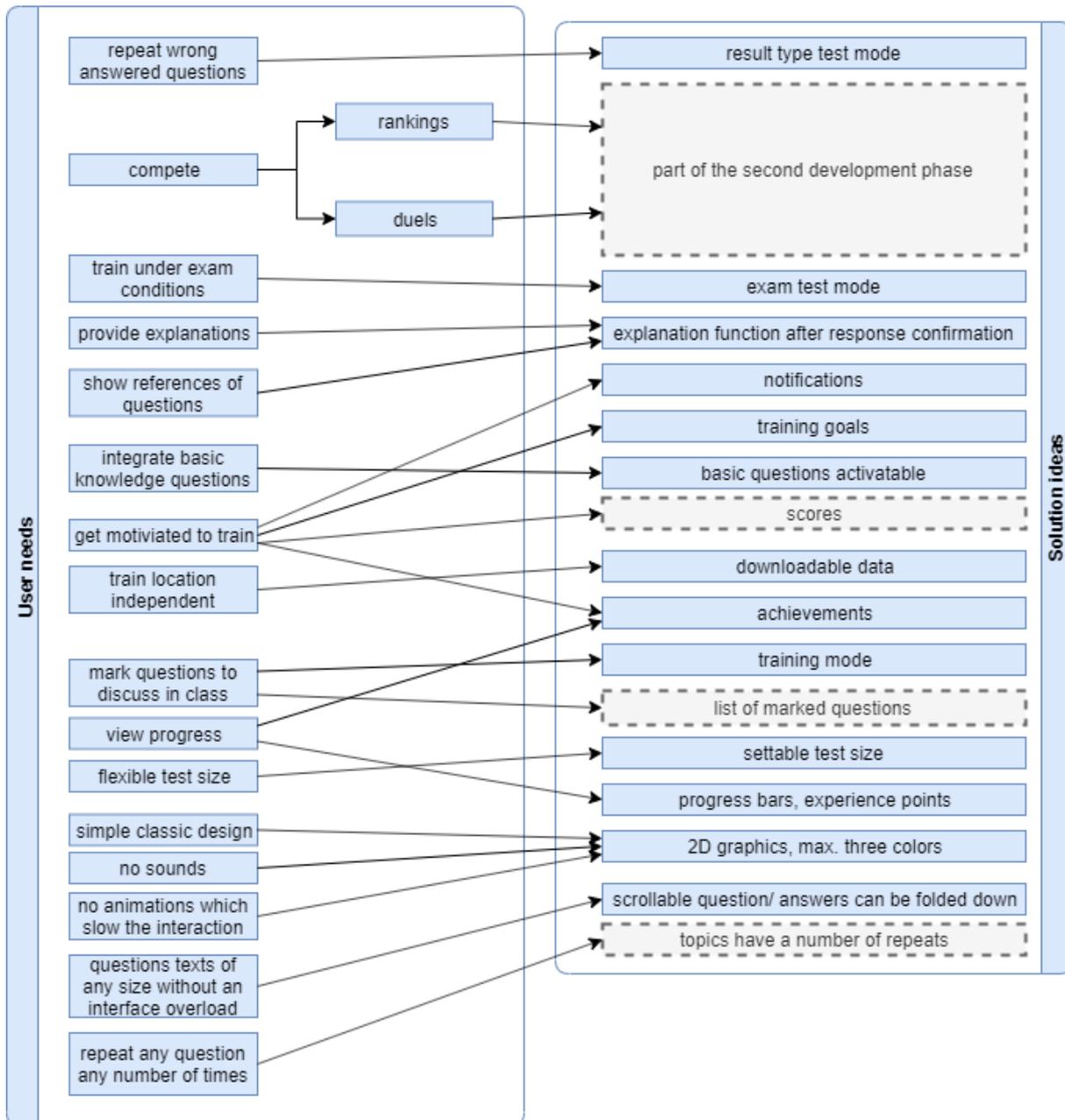


Fig. 2.8: Determined user needs and discussed solutions

The blue ones could be realized, whereas the gray ones have been rejected or postponed to the second development phase of the application, where a server integration will be implemented. For the repetition of questions has to be mentioned, that it has also been fulfilled, as each topic can be repeated any number of times although the number of repetitions hasn't been realized for reasons of interface clarity.

2.6 User types

There are three types of users that will interact with the system: users of the application, data administrators and the developers. Each of them interacts with a different part of the system.

The users use the application to learn the available set of questions provided by the administrators. They don't have any influence on the given data set, except for filtering it locally, based on their current needs.

Administrators manage and provide the set of questions used in the application. As question data and application code are two separated components, administrators don't need to interact with the program code or application. Their task is to convert the created data into a valid format and making it downloadable via a static link.

The developers are experienced programmers which interact with the application code. They are responsible for all functional requirements. Developers don't have any influence on the question content, except for providing the necessary integration tools as converter and import function.

The following table is a summary of all different user types and their corresponding context of use.

Actor	Description	Application context
User	Is a human student of the subject of biochemistry	Represents the later user of the application.
Customer	The customer specifies the rough requirements for the application	The representatives of biochemistry.
Administrator	The administrators will be able to upload the database including texts and images.	Employees of the biochemistry.
Developer	Person, which designs and implements the application.	Represented by Annalisa Degenhard and Nico Brenner[3].

Tab. 2.2: User type summary

2.7 Context of use

This project is about an android application called MoBiLe which is provided by the institute of biochemistry at the university of Ulm. Bachelor students of biochemistry have three exams and a corresponding set of about five thousand questions to learn during their studies, whereby each exam is about a certain part of this question set. The application is used to ensure that the correct set of questions is learned for each exam and helps the students during their whole studies to keep constantly training.

2.8 Design and implementation constraints

The application should be runnable on each android device with an API higher than 16 (Jelly Bean, released in June, 2012). Furthermore, the application is requested to work on android devices of any size. As the customer asked for a high level of comprehension support for the code and data module, the code shall have a comment coverage from over 90%. In addition to method and class descriptions, all graphical elements of the layouts, except for secondary layout containers, shall have a content description. All strings, arrays and content descriptions shall be saved in the corresponding value file of the project. This is required to facilitate quick changes as well as a possible later integration of an English version. The desired design of the application interfaces has a simple, bright appearance with rounded shapes and blue as primary color. In addition to this, it should reflect the study of biochemistry. Another functional and visual restriction is, that both versions, the iOS and the android version should have the same functional range as well as a similar appearance.

2.9 User documentation

As this documentation is mainly targeted on developers, there is another documentation in the attachments which is especially for administrators. The administrator manual consists of a detailed content description, a user and a style guide as well as instructions for all changes that can be made without the support of a developer. As the application has integrated instructions, there isn't a separate user manual, but all necessary information about using the application can be looked up in the administrator guide.

- Administrator guide: "moBiLe_administrator_guide.pdf", 37 pages

3 System requirements

This section defines all functional and non-functional requirements of the application. Every requirement has an ID, a description as well as its justification and a priority value. Possible priorities are listed below.

Priority	–	-	0	+	++
Relevance	very low	low	medium	high	very high

3.1 Functional system requirements

3.1.1 Application: general

FR 1 Language

The language of the application is German.

Reason: The corresponding lectures and exams are also in German.

++

FR 2 Offline Application

The whole application is offline. The questions can be downloaded from the internet and from will from then on be available independent of an available connection.

Reason: To enable the use of the application at any location.

++

FR 3 Start Application

The app is startable via the depending shortcut named “MoBiLe“ in the menu of the user’s device. Clicking on it leads either to the first set up interface or to the main view, depending on the existence of a user data file.

Reason: Familiar operation facilitates learnability.

+

FR 4 User presets

When starting the app, the system will check whether the required user settings (pending exam, number of questions per test and training goal) have already been set. If not, a set up display will be shown, which allows the user to set the required values and his preferences for the tests. Unless they haven't been deleted, this screen won't show up again.

Reason: To ensure that the right exam is selected which prevents the user from learning the wrong questions.

++

FR 5 Quit application

The application can be quit by the common way to close apps of the users device.

Reason: Familiar way of closing app supports learnability.

+

3.1.2 Application: Features

FR 6 Test modes

The user has the ability to choose between four different test modes. Topic test, result type test, training and exam mode.

Reason: The different test modes allow the user to learn targeted.

+

FR 7 Test recovery

The user is able to cancel a test session and recover it at a later time. The test session then continues at the point of termination. Recoverable is up to one session of type topic or result type test. The progress of a training or exam test are discarded if the user cancels.

Reason: Allows the user to continue a session at a later time.

+

Interfaces

FR 8 Topic overview

The topic overview consists of a list of all existing topics in the data set which correspond to the currently set exam. It allows the user to switch between topics as well as the desired difficulty level.

Reason: Necessary functionality to enable a targeted selection of questions.

++

FR 9 Statistics

The statistics allow the user to trace his personal progress. This includes a summary of his test results, the progress measuring values and achievements.

Reason: To support traceability and motivates the user.

+

FR 10 Settings

The settings are used to adjust the app according to the user's preferences. Settable are user name, pending exam, questions per test, streak goal, notifications and basic questions. Furthermore the settings include a user manual, download- and reset function.

Reason: Used to support the customizability.

+

FR 11 Question view

The question view allows any size of question and answer texts. The user has the opportunity to change his selected response until he confirms it. Basic questions are labeled as such.

Reason: Necessary for a high flexibility regarding possible input types.

+

FR 12 Question feedback

After confirmation, the user gets feedback to his answer. Furthermore he gets the opportunity to mark the question as a training question and view the questions explanation and continue with the next question.

Reason: Supports the user to understand the test content.

+

FR 13 Test summary

The application provides a test summary after a completed test session, which summarises his results as well as his rewards gained through the test completion.

Reason: Supports the traceability of the user's progress.

+

Data

FR 14 Question format

The application is able to import the question set as a JSON File.

Reason: The json format is an appropriate format for both, Android and IOS.

0

FR 15 Image format

The application supports images of questions which have the format png or jpg. The image's name is linked in the corresponding attribute of the question data.

Reason: The json format is an appropriate format for both, Android and IOS.

-

FR 16 Static download link

The user can download the question data set over a static download link. Doing this will automatically integrate the data into the application.

Reason: Simplification of the download for the user.

0

FR 17 XML structure

The structure of the xml file is determined by the administrators. The only constraints are, that each questions must have an unique ID, must be assigned to a unique subject area and the filename of the associated image has to be named in a corresponding column.

Reason: Ensures a high degree of adaptivity of the database.

+

FR 18 Question ID

Each question has an unique ID. The scheme of the question IDs were determined by the data administrators.

Reason: Ensures a correct identification of a certain question.

++

FR 19 Question topics

Each question is assigned to a certain theme. The application is able to identify the different topics and create specific topic tests. The themes are determined by the institute of biochemistry.

Reason: To categorize questions and offer tests based on a specific theme.

++

FR 20 Question difficulty

Each question is assigned to a certain difficulty level. The application is able to identify the difficulty and create specific topic tests of a certain difficulty. The difficulty levels are determined by the institute of biochemistry.

Reason: To enable a logical order in the theme based questioning.

++

Topic test mode

FR 21 Test mode: topic based

The topic based test mode is the main test mode. The topic based algorithm sorts the questions as follows: semester, topic, difficulty. Afterwards he creates a random set of the user's desired number of questions matching the requested topic and difficulty, regarding the questions priorities.

Reason: To enable purposeful learning of certain topics.

++

FR 22 Topic overview

The user can view his current progress for each topic and difficulty. The progress summary gives information about the number of right, wrong and not yet answered questions.

Reason: Facilitates the tracking of the personal progress.

+

FR 23 Rating of completed topic test sessions

If a user completes a test session successfully, he gets 1XP for each easy right answered question, 2XP for a moderate questions and 3XP for each right answered difficult question, and the results of the test will be settled with his current state. Furthermore, the result will be taken into account in determining, if the daily goal has been reached.

Reason: Motivation factors to keep on practicing.

0

FR 24 Rating of canceled topic test sessions

If a user cancels a test session, his previous state doesn't change. All questions answered until canceled don't get rated as answered. The test session is not taken into account in determining the reach of the daily goal.

Reason: To allow user to avoid offsetting his answer results.

0

Result type test mode

FR 25 Test mode: based on result type

The user is able to select wrong answered questions in each theme and level and has the opportunity to test them for it's own.

Reason: Allows the user to repeat wrong answered questions without the need of marking each of them.

0

FR 26 Rating of result type test sessions

If a user completes a result type session successfully, the results of the test will be settled with his current state and the user gets one, two or three XP, depending on the current test difficulty, for each answered question. The number of answered question will also be taken into account for reaching the streak goal. If the user doesn't complete the test, the results won't be settled with his progress.

Reason: Easy understandable offsetting of points.

0

Training mode

FR 27 Training mode: Marking questions

After answering a question, the user has the opportunity to mark or undo a marking of the question as a training question.

Reason: Allows the user to train specific questions or mark them for discussions with other users.

-

FR 28 Test mode: training

The training mode allows the user to train certain questions. To do so, he can mark the corresponding question as a training question after answering it in another test session. The training session will then consist of all marked training questions. The marking can be undone at any test session when answering the question.

Reason: Enables the user to train certain questions.

-

FR 29 Rating of training sessions

A training session doesn't have any impact on the progress, XP, daily goal or other statistics of the user. If a previously wrong answered question of a test session was correct answered in training mode, it won't change the response state to correct.

Reason: To clearly separate training and rating.

-

Exam mode

FR 30 Test mode: exam

The exam mode enables the user to test his knowledge under conditions similar to an exam. Each exam test consists of thirty questions of the current semester which are a random mix of all belonging topics and difficulties. Furthermore, the user has 45 minutes for the test which corresponds to the normal time limit of a real exam.

Reason: Allows the user to test his knowledge under exam conditions.

+

FR 31 Rating of exam sessions

If a user completes a exam session successfully, the results of the test will be settled with his current state. He also gets the sum of each questions difficulty as XP and the test gets taken into account for reaching the daily goal with 30 questions.

Reason: To separate training and exam clearly.

0

Statistics

FR 32 Statistics overview

The user is able to overview his progress in a corresponding diagram. It gives information about his progress about the time split up in two lines: answered questions and right answered questions. The time period of the overview is settable to this week, this month and all time. Only the questions of the currently set exam are considered.

Reason: To give the user an overview of his progress.

0

FR 33 Statistics: Streak

The streak gives the user information about how many days in a row he has been training. To get the streak, he has to answer a certain amount of questions, which he adjusted in advance in the user settings. If he doesn't answer enough questions, the streak will be reset to zero.

Reason: Used to motivate the user to keep training

0

FR 34 Statistics: XP

The XP value gives the user information about the amount of right answered questions in correlation with their difficulties of the current set exam. Easy questions get 1XP, medium 2XP and difficult questions 3XP, if right answered. Reason: Allows the user to get an overview about his progress and compare his efforts with his fellow students.

0

FR 35 Statistics: Achievement overview

The user can reach three different achievements. Their current state is shown in the achievement overview which can be opened via the statistics. Achievements are answering of all questions of an exam, number of answered question and holding a streak.

Reason: Motivate the user to practice.

-

FR 36 Statistics: Achievement: all questions answered

The user can get an achievement for answering all questions of the selected exam.

Reason: Motivates the user to practice.

-

FR 37 Statistics: Achievement: number of answered questions

The user can get an achievement for answering a certain number of questions. There are three reachable goals: 250, 500 and 1000 questions.

Reason: Motivates the user to practice.

-

FR 38 Statistics: Achievement: streak value

The user can get an achievement for reaching a certain streak level. There are three reachable goals: 7, 21 and 84 days.

Reason: Motivates the user to practice.

-

Settings

FR 39 Settings: User name

The user is able to set his name in the user settings. As long as the application will be offline, this setting won't have a specific use and is only used for personalization.

Reason: Personalization of the application usage.

0

FR 40 Settings: Exam

For using the test mode, it is necessary to set the right pending exam. This is necessary for the algorithm to only provide the belonging questions. By default, the exam will be set to the first exam. The user can change the current exam at any time.

Reason: Filtering of the questions depending on the current exam.

++

FR 41 Settings: Questions per test

The number of questions per test determines of how many questions each test session will consist. Settable values are all numbers from 10 to 100 in increments of ten.

Reason: To support the using comfort of the app.

+

FR 42 Settings: training goal

The user can set his personal training goal. It will be continually checked by the application and the user gets reminded, if he hasn't reached it yet. Therefore, he can choose between 10 question a day, 20 questions a day, 40 a week and 60 questions a week.

Reason: Motivation to keep on training adapted to the user.

+

FR 43 Settings: basic questions

The user is able to enable basic questions. They won't be in the exam, but help the user to get a deeper understanding of the learning content. Basic questions will be marked, to point out, that this isn't an exam question.

Reason: Helps the user to get a deeper understanding of the learning material.

+

FR 44 Settings: reset

The user can reset his progress through the settings. Doing so, his statistics as well as the current answer states of each question will be deleted. The data set won't get deleted.

Reason: Allows the user to start from the beginning.

-

FR 45 Settings: notifications

The user has the opportunity to enable notifications which will remind him to keep practicing if he has not reached his streak and time is running out.

Reason: Motivation factor to help the user staying on track.

-

FR 46 Settings: import tool

The user can import the question data through a download function in the settings. Doing so, processes and saves the data in the internal storage so that is not necessary, to import it again.

Reason: Facilitates the usage for the user.

-

FR 47 Settings: instructions

The user can view explanations about all features of the application via the settings interface.

Reason: To support the comfort of use.

-

3.2 Non-functional System requirements

QA 1 Robustness

Of one hundred completed test session, a maximum of one session may be aborted due to an error.

Reason: To ensure error robustness and thus the product quality.

0

QA 2 Coding standard

For the implementation of the application code, general coding standards have to be considered.

Reason: For a comprehensibly code which can be reused by following developers.

+

QA 3 Graphical representation

The application will exclusively consist of 2D design elements.

Reason: Most appropriate representation for this type of application.

-

QA 3 Operability

The user should be able to control and use the application within a maximum of 15 minutes without any help.

Reason: To ensure a high comfort of use.

-

QA 4 Animations

The duration of each animation should not last longer than three seconds.

Reason: To avoid unnecessary waiting of the user.

0

Product quality	very good	good	normal	not relevant
Functionality		X		
Adequacy		X		
Correctness			X	
Interoperability				X
Reliability			X	
Fault tolerance			X	
Recoverability		X		
Usability			X	
Comprehensibility			X	
Learnability	X			
Efficiency		X		
Changeability	X			
Modifiability	X			
Stability			X	
Adaptivity			X	

Tab. 3.1: Non functional requirement summary

4 Software specifications

4.1 System interfaces

This chapter goes into detail about the structure of the application's interfaces. This includes an general overview about the dialog structure for getting an idea about the interaction possibilities of the app. Furthermore, mockups provide an overview about the planned interface composition, including elements, interaction possibilities and planned animations. While the dialog structure diagram already shows the exact dialog structure of the later app, the mockups are only a schematic representation of the later application. Design details like icons or secondary colors of the final application may differ from the following mockups.

4.1.1 Dialog structure

The dialog structure is the totality of all interaction possibilities of the application including their effect. To get a quick overview, a general dialog structure diagram is shown in the following. Excluded from the diagram are the back operations in one of the main views (topics, statistics and settings) because the triggered effect depends on the previous view from which the user switched to the current view. Navigating back will lead to this previous view. Furthermore, there are two possible starting interfaces mentioned. Which one is be displayed depends on the existence of user data. This means that if the user hasn't already set his profile, the app will show up a set up interface (4.1.2). Otherwise the app checks, if there is a recovery session left. If so, it displays an interface which allows the user to choose between starting a new test session or continue the last session (4.1.2). If there is no recovery session, the application will directly lead in to the topic view (4.1.2).

4.1.2 Mockups

In the following, the interfaces named in the previous abstract will be explained from the point of interface composition, interaction elements and general design elements. This is not used as a exact preview of the later design. Placements, texts, secondary colors and other design aspects may differ from the following mockups. Only composition and primary coloring are an exact representation of the later application.

Start menu: first use

The first view of the application is case sensitive. If it's the user's first time of use, which means that, there doesn't exist any user data on the device, the application will show up a user set up interface right below the welcome area. It's schematic structure is shown in the following mockup.

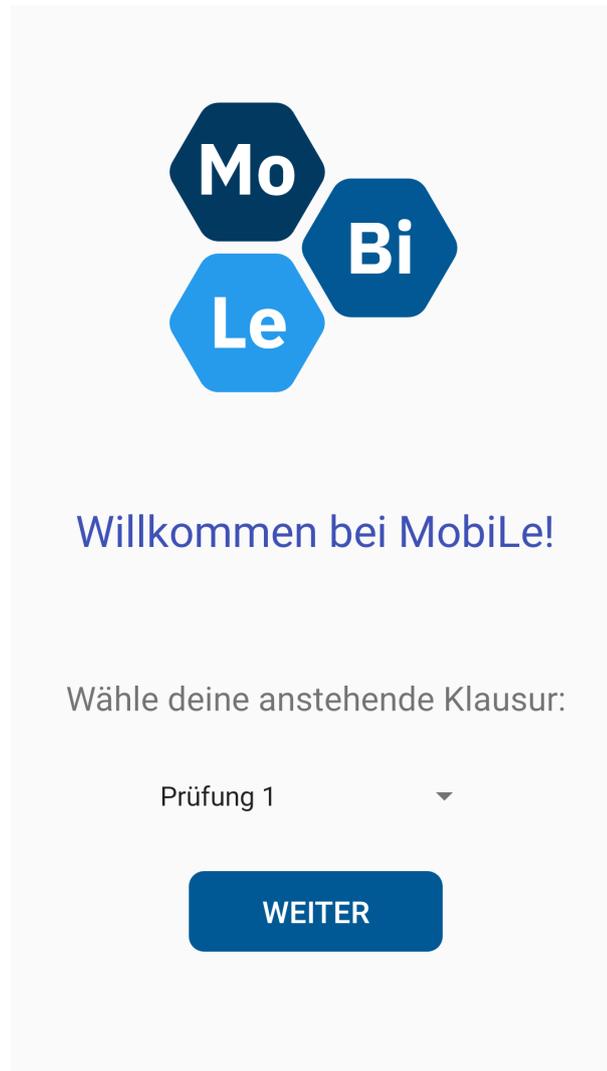


Fig. 4.2: Start view for first use

The first set up consists of a possibility to set the pending exam for which the user wants to learn and a confirm button. As this is a required value for using the app, the first exam is already set as default value. Pressing the confirm button leads to the next set up interface in the form of an inlay which is positioned at the same place as the previous set up inlay.

Animations: To help the user comprehend the changes effected through pressing the confirmation button, the second inlay will appear with a slide animation. This increases the traceability and the dynamics of the application. The latter is desirable because of the central intent of learning the exam content through gamification. Dynamics can at this point help to give the application a playful character without losing it's serious appearance.

The following mockup is an illustration of the second set up interface.

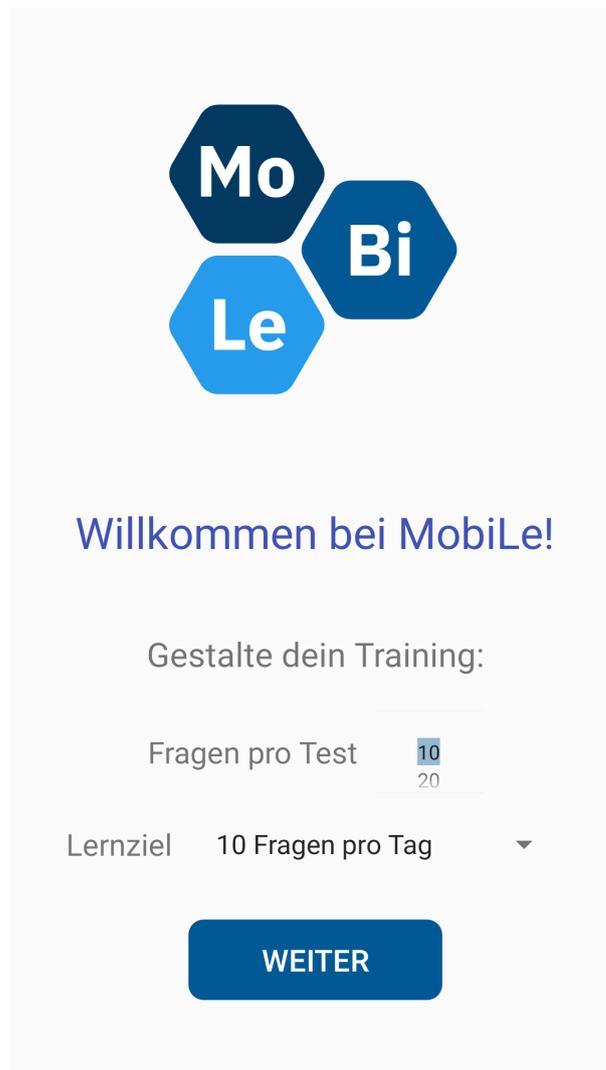


Fig. 4.3: Start view for first use

The second inlay allows the user to choose the desired number of questions per test and a personal training goal. The first value is used for topic, result type and training tests. Settable are all numbers from ten to one hundred in the steps of ten. For the training goal, the user has the choice between four different goals which are rather on a daily or weekly basis. This has the goal to support different intensities of use.

Main interfaces

The application consists of three central interfaces: topic overview, statistics and settings. Each of them is composed in the following way: a bottom bar which allows a quick navigation between them and an inlay with the respective content.

Topic overview interfaces: The topic overview interface shows all topics which will be part of the set exam. The topics are represented in a card carousel which allows the user to quickly

change the selected topic by swiping left or right. Each topic view is composed of its unique icon, title and the corresponding progress chart, as well as a start button and a difficulty selector. Beside the topic overviews, two buttons positioned on the bottom right, allow the user to easily start a training or an exam session. The structure of the interface is as follows:

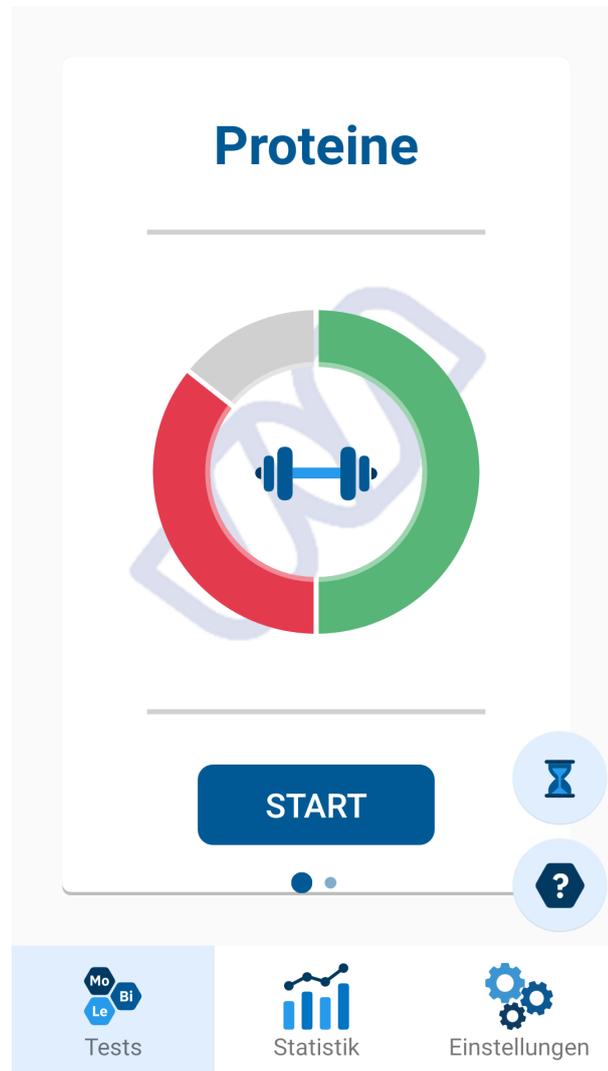


Fig. 4.4: Topic list overview

The progress chart gives information about the number of wrong and right answered questions on the currently selected difficulty level and topic. Therefore the progress bar consists of two colors for each result type. While the start button starts a new topic test session, clicking on the progress chart will start an result type test. The difficulty selector can set three different difficulty levels. The current difficulty is illustrated as a dumbbell, which gets heavier with the increasing difficulty.

Animations: The change of the topic card is clarified by a slide animation effected through a left or right swipe. Above all, this animation has the goal to support the traceability of the change, helping the user to understand, that he now see's another topic. On the other hand the animation supports the impression of a dynamic interface.

Since an exam takes about forty five minutes and there is no possibility to interrupt, the application always asks for confirmation of exam start requests. This succeeds via a dialog appearing above the topic overview interface and structured as follows:

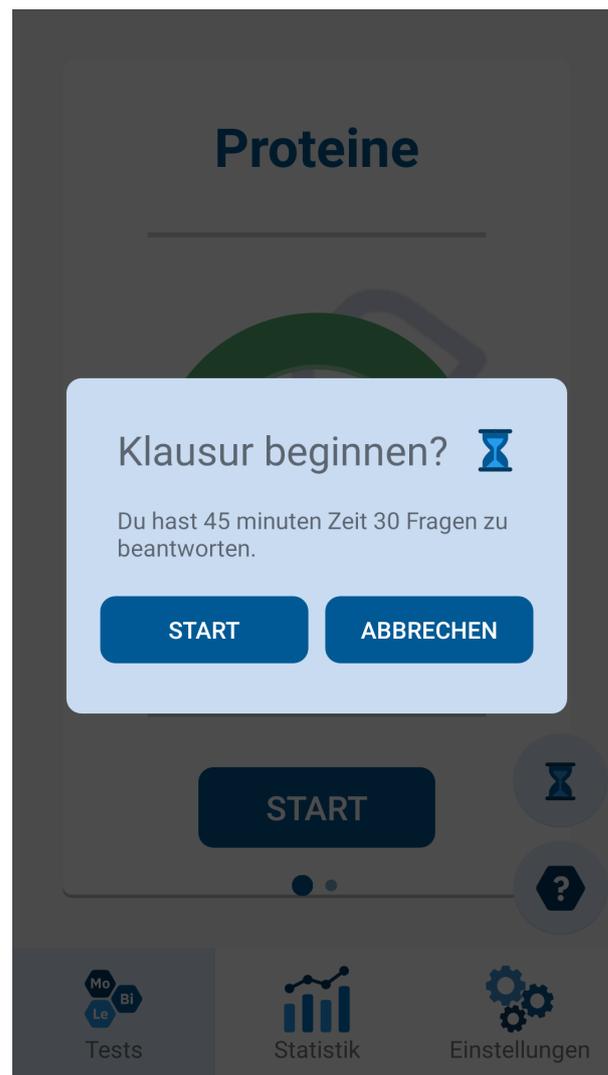


Fig. 4.5: Topic list overview: exam start request dialog

Beside an explanation text about an exam session and the start request text, the dialog offers via two buttons the opportunities to cancel the exam start or proceed. A cancellation leads back to the topic overview.

Main view with existing recovery

Every time, the user switches into the topic overview, the application checks, if there is an unfinished test session of the user. If so, the application displays an interface which allows him to choose via two buttons between starting a new test or continuing the aborted session. The former switches to the topic overview interface, while the latter loads the saved recovery and continues at the point of cancellation. The corresponding interface is illustrated bellow.

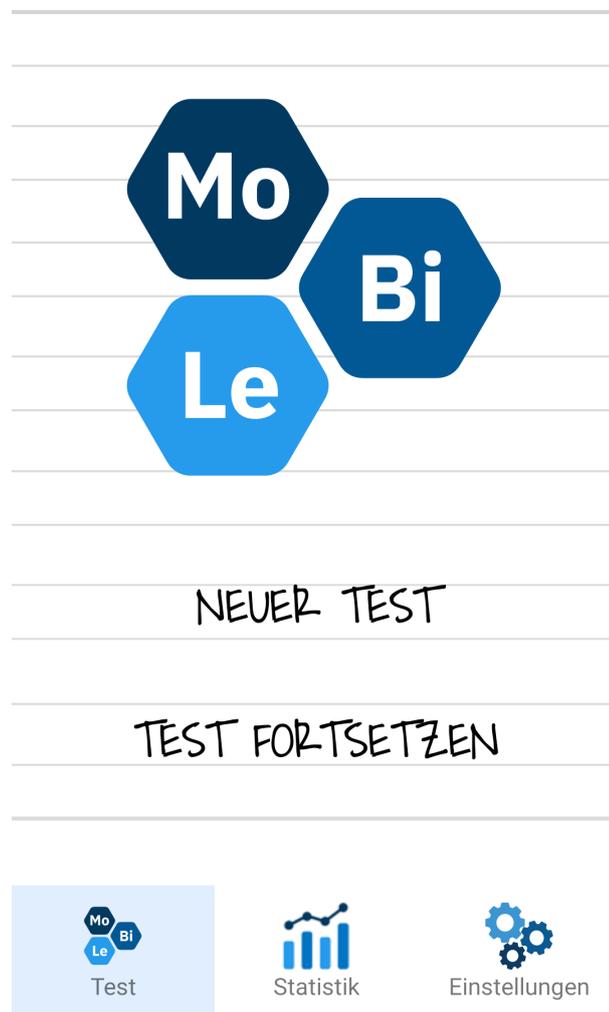


Fig. 4.6: Topic overview tab if recovery exists

Topic overview in case of missing data

The application handles two kinds of data lacks: partially missing questions, which is the case if there are none available for a difficulty level of an existing topic. In this case, the topic is shown

as normal, except for the missing difficulty. It will be represented as in the following mockup.

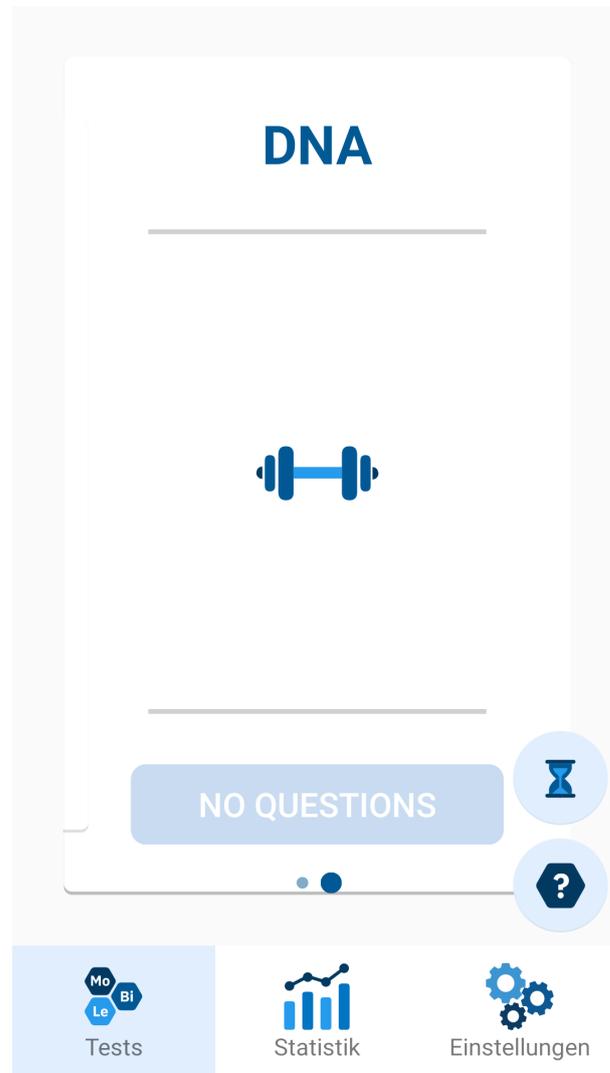


Fig. 4.7: Topic overview with missing questions for a specific difficulty

The view allows the user to change the difficulty but doesn't show the progress chart. Furthermore, the topic test start button is disabled to avoid confusion of the user.

If the user hasn't imported any question data yet, the app will replace the topic overview with a notification interface which allows the user, to download questions. The corresponding interface is illustrated in the mockup below.

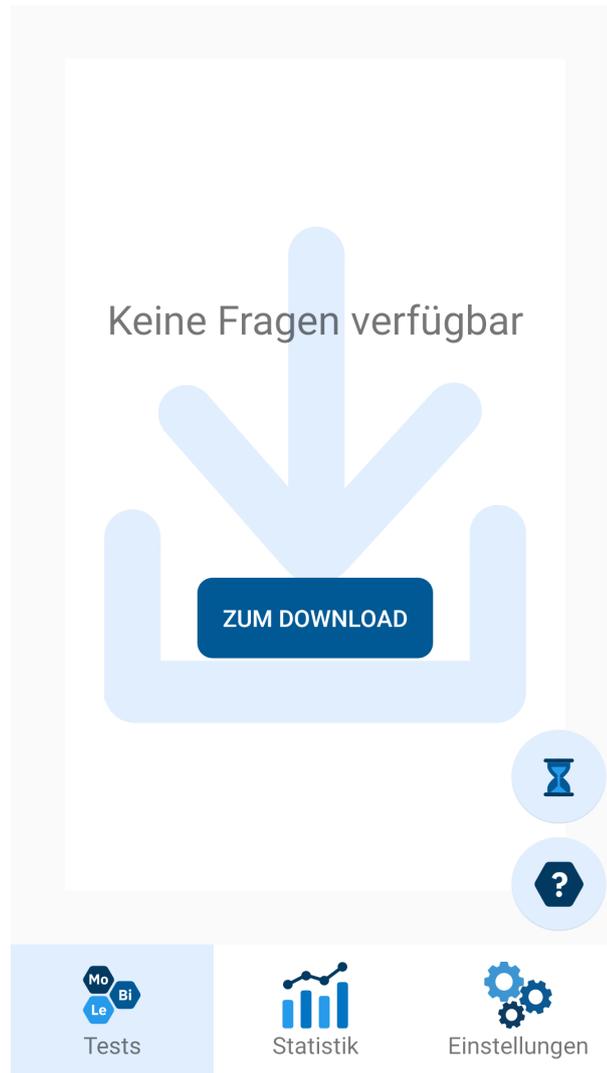


Fig. 4.8: Topic overview with missing questions

By clicking on the download button, the corresponding dialog gets displayed as in the section 4.1.2 described. The training and exam button are still enabled but will show an error hint dialog, which informs the user about missing data, if he clicks on it.

Statistics interfaces

The statistics interface enables the user to track his preparation progress. The statistics are always related to the currently selected exam. It consists of the results overview diagram over a settable period of time, the users current XP and streak value, as wells as a trophy room which is accessible via a button on the bottom. The overall structure is illustrated in the two graphics below.

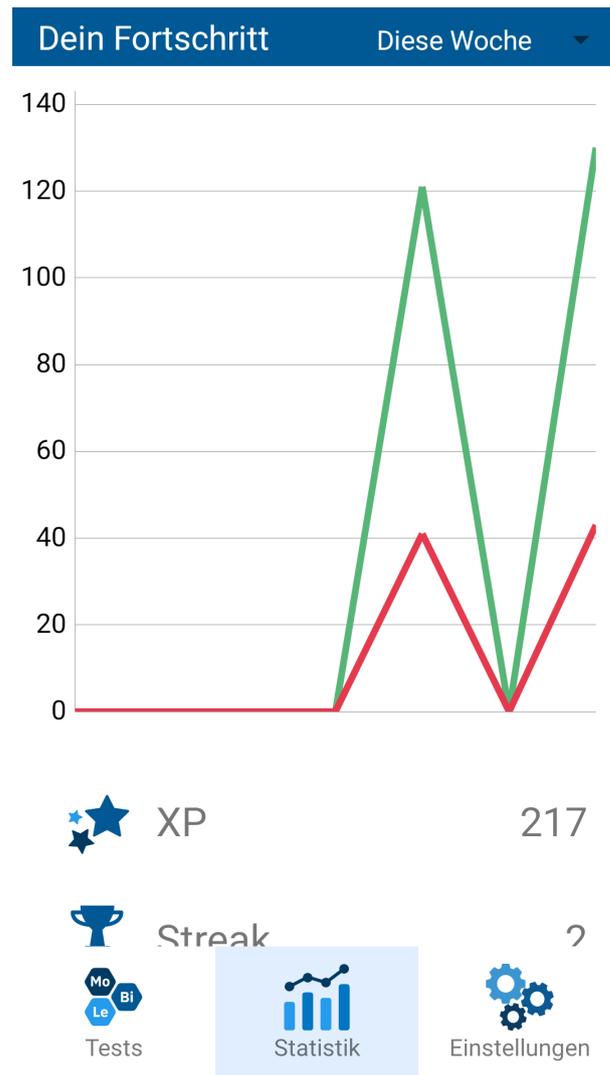


Fig. 4.9: Statistics overview: upper part

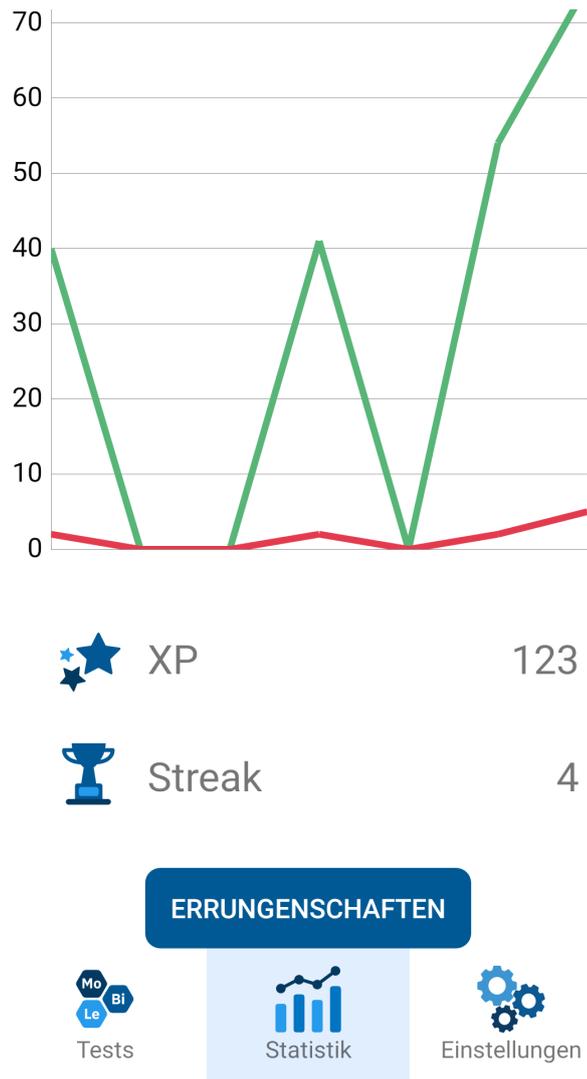


Fig. 4.10: Statistics overview: bottom part

The **trophy room** is an interface which shows all available and reached achievements. To support an easy and comfortable usage, the achievement icons are shown in a card view carousel. Like this, the user can switch between the different achievements via left or right swipe. The whole trophy room dialog is shown in the image below.



Fig. 4.11: Statistics overview: trophy room

Settings interfaces

The settings interface allows all necessary settings for a personalized testing. To avoid confusion, the settings interface is designed as a regular preferences interface of an android app. Each setting has a preferences entry of it's corresponding type.

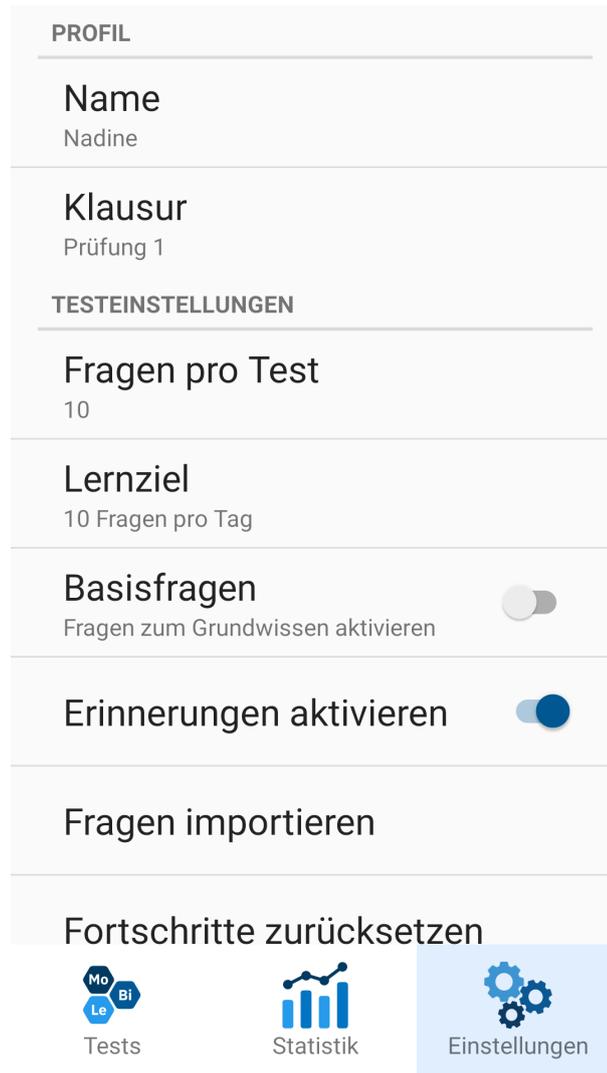


Fig. 4.12: Settings overview

In opposite to the settable preferences, the download preference is not realized with a regular preference dialog. As it consists of several steps, a custom dialog was necessary for its realization. The dialog consists of the three steps for importing a data file, each with an corresponding icon and instruction text. For the download in the first step, a button is used, which opens the corresponding website in an external browser application. To import the data file in the third step, a button is provided, which opens a file selector that allows the user to search and select the desired data file. Furthermore, to ensure the traceability through the user, a state information text is placed at the bottom of the dialog, giving information about the existence of an integrated file and its last date of change. Depending on the existence of the file, its text color automatically changes from red to green, which shall support a faster understanding. The corresponding download mockup is shown in the graphics below.



Fig. 4.13: Settings overview: download questions dialog

Another separate dialog is the instruction dialog, which consists of two menus and an instruction view. The first interface shows the main menu consisting of four manual topics. To allow a quick control, the menu items are visualized as four large icons in a grid layout. The corresponding mockup is shown below.

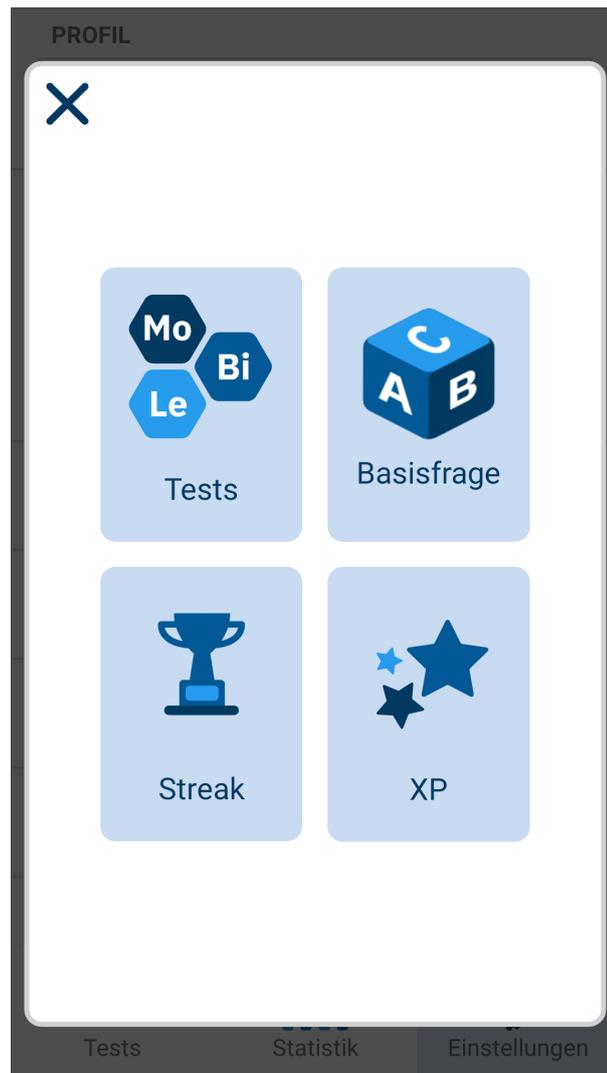


Fig. 4.14: Settings overview: instructions main menu

If the user clicks on the tests menu item, a sub menu of all test related instructions will be displayed. As it contains about six entries, a regular list interface has been chosen, to ensure the clearness of the menu. To allow the user to return to the previous menu, a back button appears on the left upper corner, on the right of the close button.

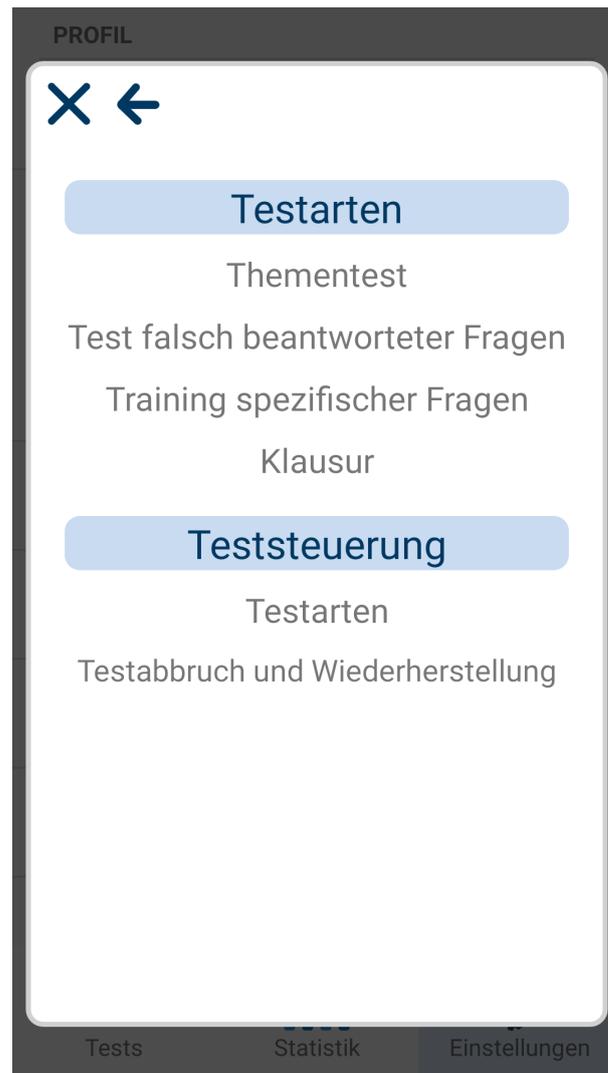


Fig. 4.15: Settings overview: instructions tests menu

The instruction view itself always consists of a title, which summarizes the following feature, an instruction image or video and a corresponding explanation text at the bottom. Everything is summed up in a scrollable view to allow text of a variable size. As in the instructions menus, the user can return to the previous view or close the dialog through a control bar on the upper left. The instruction interface is structured as in the following two mockups. The former is the view of the top and the latter of the bottom of the interface.

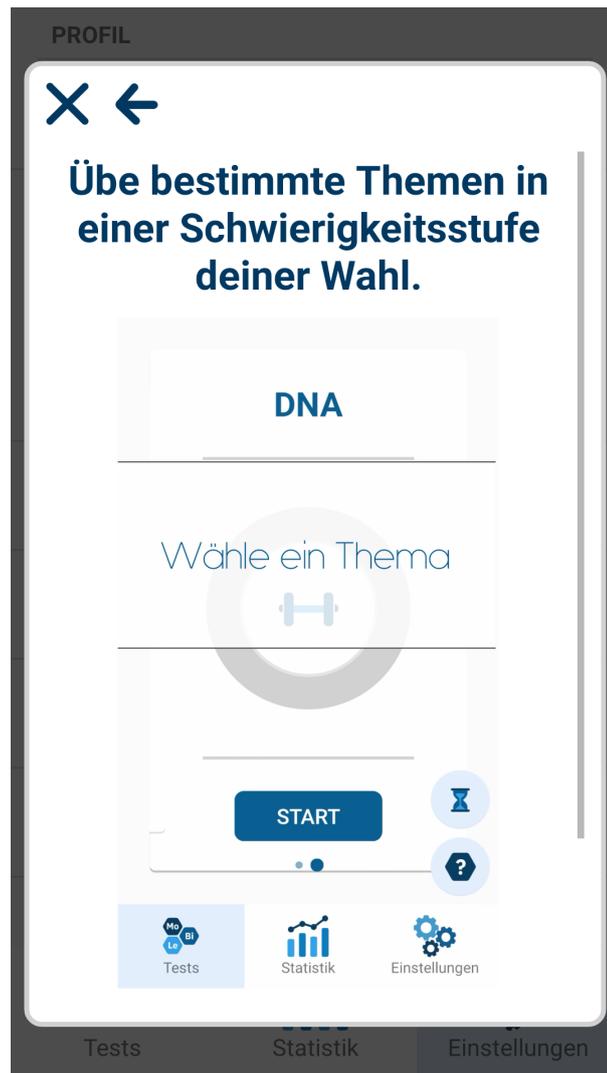


Fig. 4.16: Settings overview: instruction interface (top)



Fig. 4.17: Settings overview: instruction interface (bottom)

Test interfaces

The test procedure consists of three general interface states. The questioning, the result feedback and the final test summary. For the design, a classical quiz interface design has been chosen, to support a quick learnability of its usage. As a selected answer should be changeable, the question interface has a separate confirmation button below the responses, which is disabled until a response has been selected. The test progress is visualized as a progress bar at the top of the display. Depending on the type of test, a basic question icon appears on the left and a timer for the exam on the right of it. The overall question interface structure looks as follows:



Eine alpha-Helix in einem
Protein ist Teil der...?

Primärstruktur

Sekundärstruktur

BESTÄTIGEN

Fig. 4.18: Test: question interface with no response selected so far



Was untersucht man mit der
Röntgenstrukturanalyse?

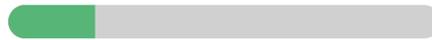
Primärstruktur

Sekundärstruktur

BESTÄTIGEN

Fig. 4.19: Test: question interface after answer has been selected

The question feedback is realized with a response depending colored background for the selected and right answer button. Furthermore, a result dialog is shown at the bottom of the question interface. This dialog allows the user to mark the answered question as training question and view its explanation. Otherwise, he can skip to the next question via a corresponding continue button.



Wo in der Zelle wird die
Peptidbindung hergestellt?

Im Zellkern

An den Ribosomen

? i

WEITER

Fig. 4.20: Test: feedback after response confirmation

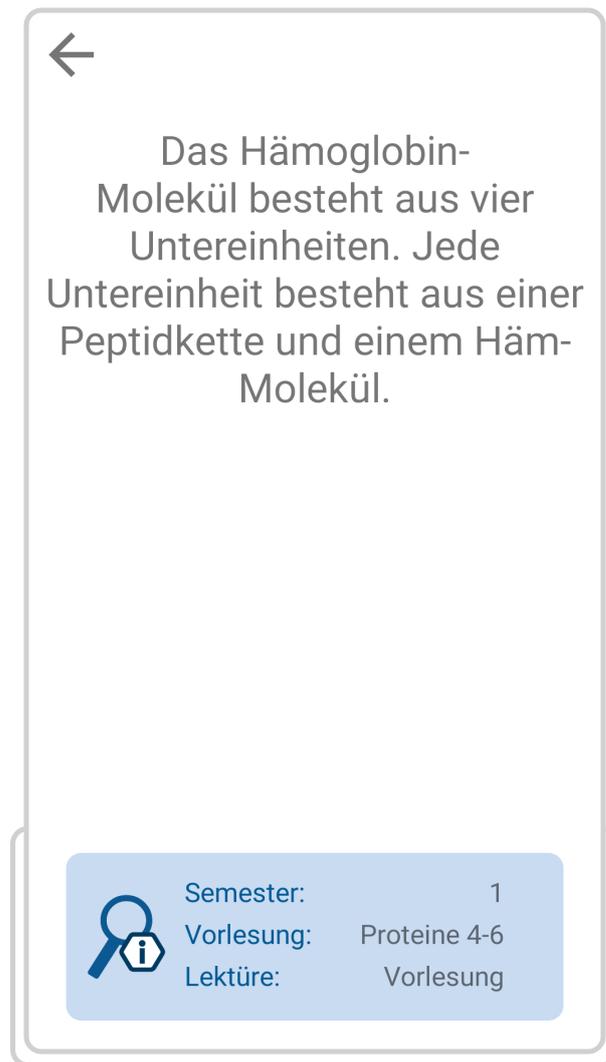


Fig. 4.21: Test: explanation view of the current question accessed through the result dialog

Animations: To help the user track his progress, the corresponding bar's updates are visualized via a continuous float animation. Furthermore, the result interface gets displayed through a float animation from the bottom to give the application a more dynamic appearance. Both animations don't have any impact on the controllability of the question interface, to avoid annoying the user with animation delays.

After confirming the last questions feedback, a test summary will be shown, which consists of a progress bar showing the number of right and wrong answered questions, as well as a textual summary and information about if the user reached his daily goal through completing this test session. In addition to this, the new streak value is shown in the foreground of the corresponding streak icon. A button on the bottom of the dialog allows the user to return to the topic overview.



Fig. 4.22: Test summary which is shown after completing a test.

Animations: To give the application a dynamical appearance, the progress bar of the test summary is animated via a continuous float animation. To avoid annoying the user with animation delays, it doesn't have any impact on the controllability of the test summary interface.

If the user has reached a new achievement through the test, the corresponding hint dialog will be shown, when he return to the main interface. The structure of this dialog looks as follows:

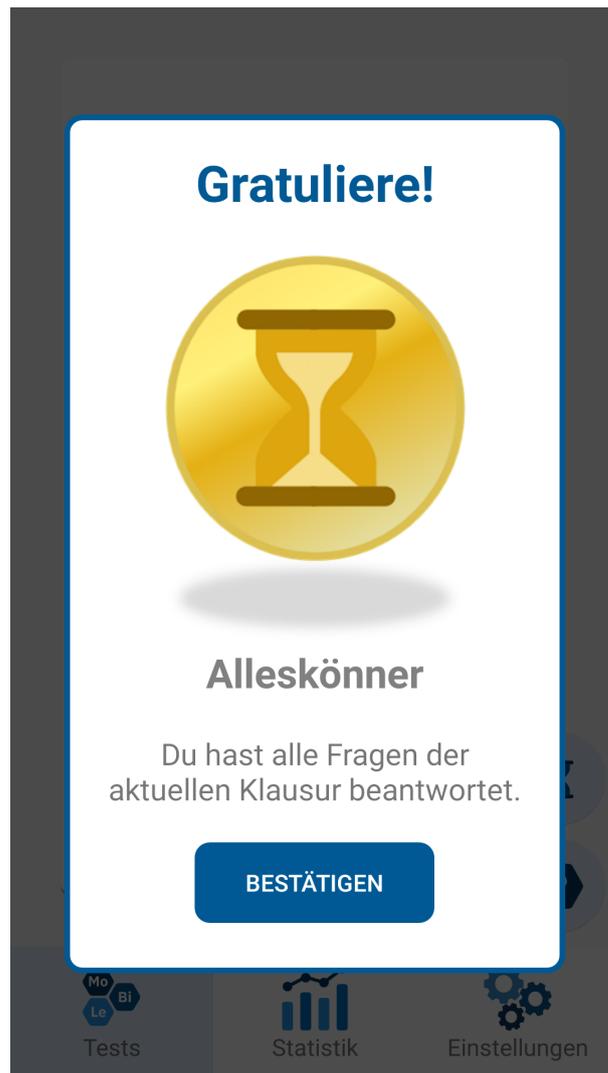


Fig. 4.23: Achievement reached notification which appears when returning to the main view after a test session.

Test cancellation: If the user wants to cancel a running test through pressing the back button, a confirmation request dialog will be shown, which ensures that the user doesn't cancel the session by accident. The dialog appears as in the following mockup:

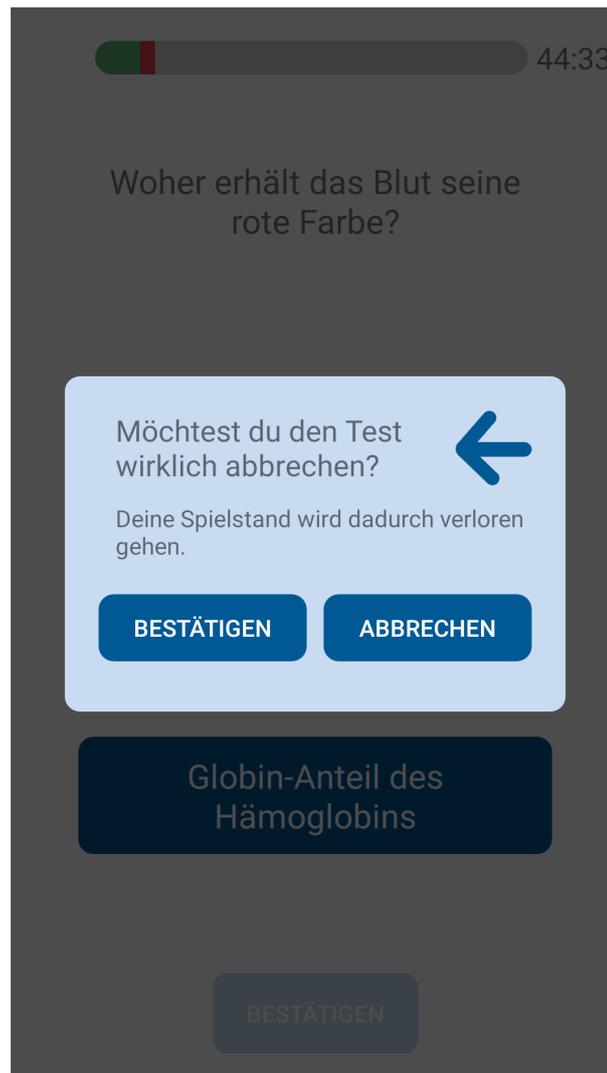


Fig. 4.24: Cancellation request which appears during a test session, if the back button has been pressed.

Notification

The user has the opportunity to activate notifications, which are shown, when he hasn't reached his training goal yet and the time is running out. The corresponding notification will be composed as in the mockup below.

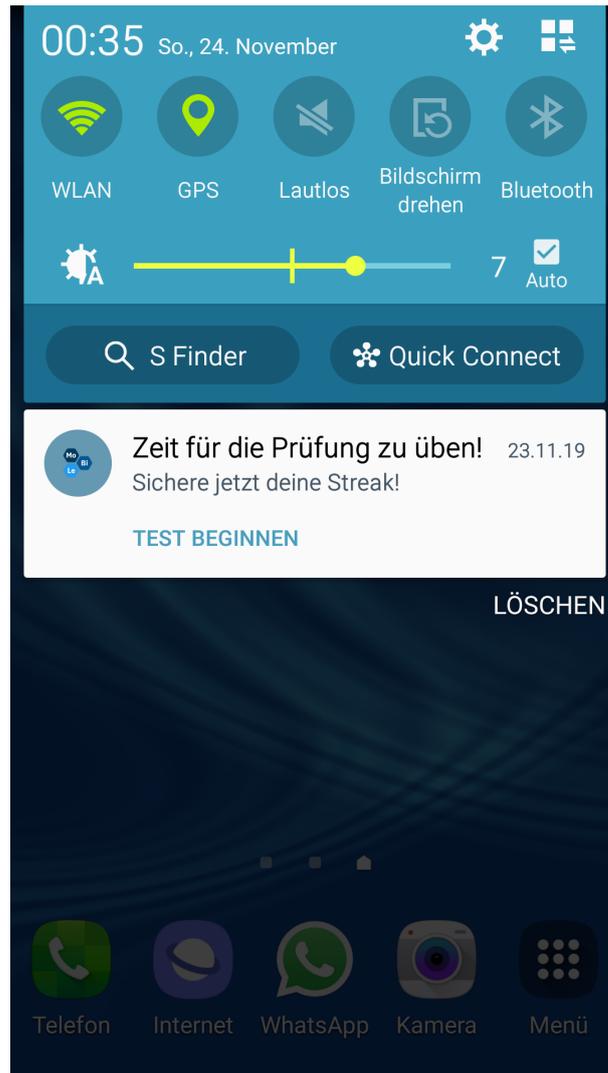


Fig. 4.25: Structure of a reminder notification.

4.1.3 Code architecture

The structure of the MoBiLe code is illustrated as class diagrams in the attachment. For reasons of clarity, the overview has been split up into three parts. A general overview of the correlations, the part of the start activity and model classes, and test activity and fragments in another separate illustration.

5 Conclusion

The goal of this project was to create a system for the students of biochemistry at the university of Ulm, which rectifies the uncertainties about choosing the right learning content for an exam and furthermore adding additional value in comparison to other learning methods. Through the cooperation with some of the students, a solution that fits all these requirements could be found. On the one hand, MoBiLe facilitates the preparation for exam through providing only the relevant questions and individual location-independent training, on the other hand the system helps the user keeping on track through the use of gamification elements. Almost every requirement of the students could be implemented. Not yet implemented are additional settings as a time for notifications or the date of the pending exam in order to get informed about the number of remaining days. Furthermore, the application is stable for regular use, but should pass another intense test phase through the students to ensure a maximum of usability and stability.

6 Vision

There are several extensions planned for the application. In a second development phase, the goal is to allow online functionalities as challenges between students, automatic data updates and a ranking. Furthermore, a tablet version will be implemented and the application will get optimized according to the feedback of the users. The goal of the institute is to promote the application and to constantly improve the quality of the studies of biochemistry through the given feedback about content and learning comfort. To achieve this, a psychologist will analyze the level of the users satisfaction as part of his doctoral thesis.

.1 Glossary

FR	Name	Data type	Description/Comment
FR 43	Basic Question Identifier	String	'BA'; Substring of question ID which identifies question as basic question.
FR 20	Difficulties	String[]	Names of the three different available difficulty levels
FR 40	Exams	String[]	Names of the available exams to learn for.
FR 30	Exam/Number of questions	int	30; number of questions per exam test
FR 30	Exam/Time limit	milliseconds	2700000 (45min); time limit for an exam session.
FR 18	Question/ID	String	Arbitrary identifier string
FR 15	Question/Image	String	Name of the image file including it's format.
FR 27	Question/ IsTrainingQuestion	boolean	Is true if question shall be part of training tests.
FR 21	Question/Priority	int	Used to ensure an even questioning by incrementing each question's priority after it's usage
FR 11	Question/Question text	String	Question text of any size
FR 11	Question/Responses	String[]	Up to five responses of any text size
FR 19	Question/Topic	String	Name of questions corresponding topic
FR 33	Statistics/Streak	int	increased when achieving training goal and reset otherwise
FR 32	Statistics/Time period	String[]	Available time periods for statistics diagram
FR 12	Question/Solution	String	Name of the right response.
FR 34	Statistics/XP	int	sum of all right answered questions, adding up their difficulties
FR 42	Training goals	String[]	The four available training goals
FR 43	User/ Basic questions activated	boolean	Is true if user wants to integrate basic questions
FR 39	User/ Name	String	String of any length and char type.
FR 41	User/Nr of questions per test	int	Desired number of questions per test
FR 40	User/ Pending exam	String	Necessary variable with name of desired exam to learn for
FR 42	User/ Training goal	String	Desired questions per day/ week as String

Tab. .1: Overview of the relevant data of the application

.2 Class diagrams

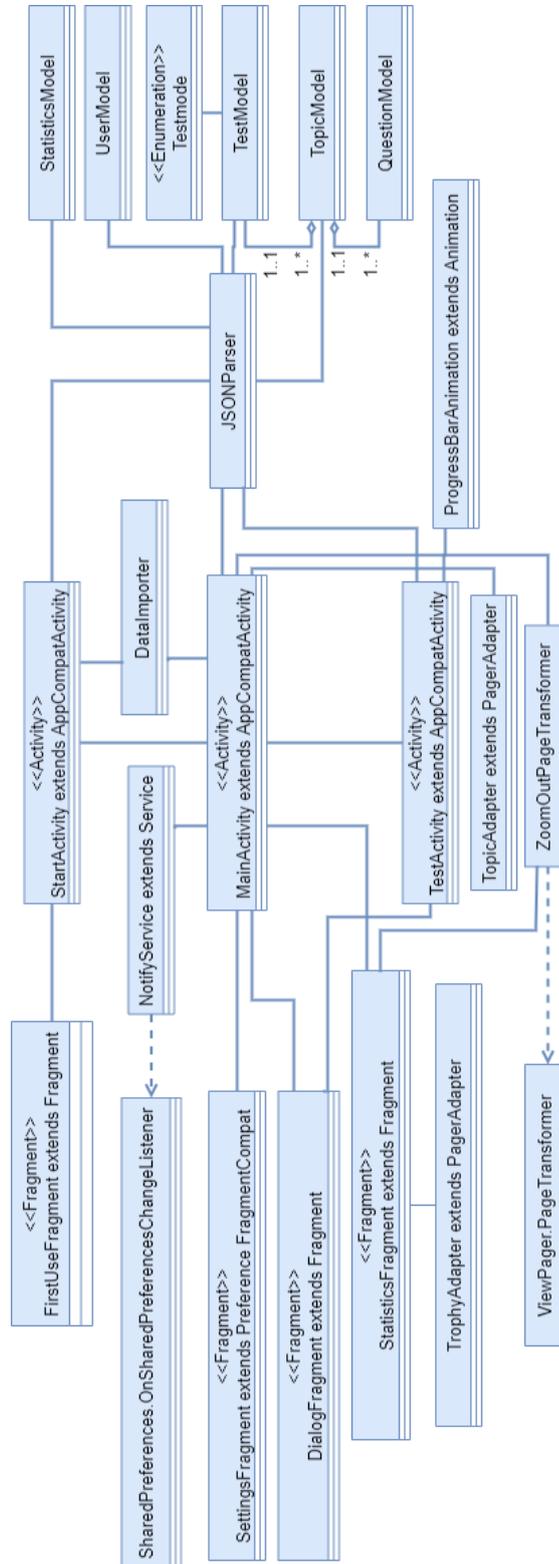


Fig. .1: Class diagram of all classes.



Fig. .2: Class diagram of start activity and context in detail.

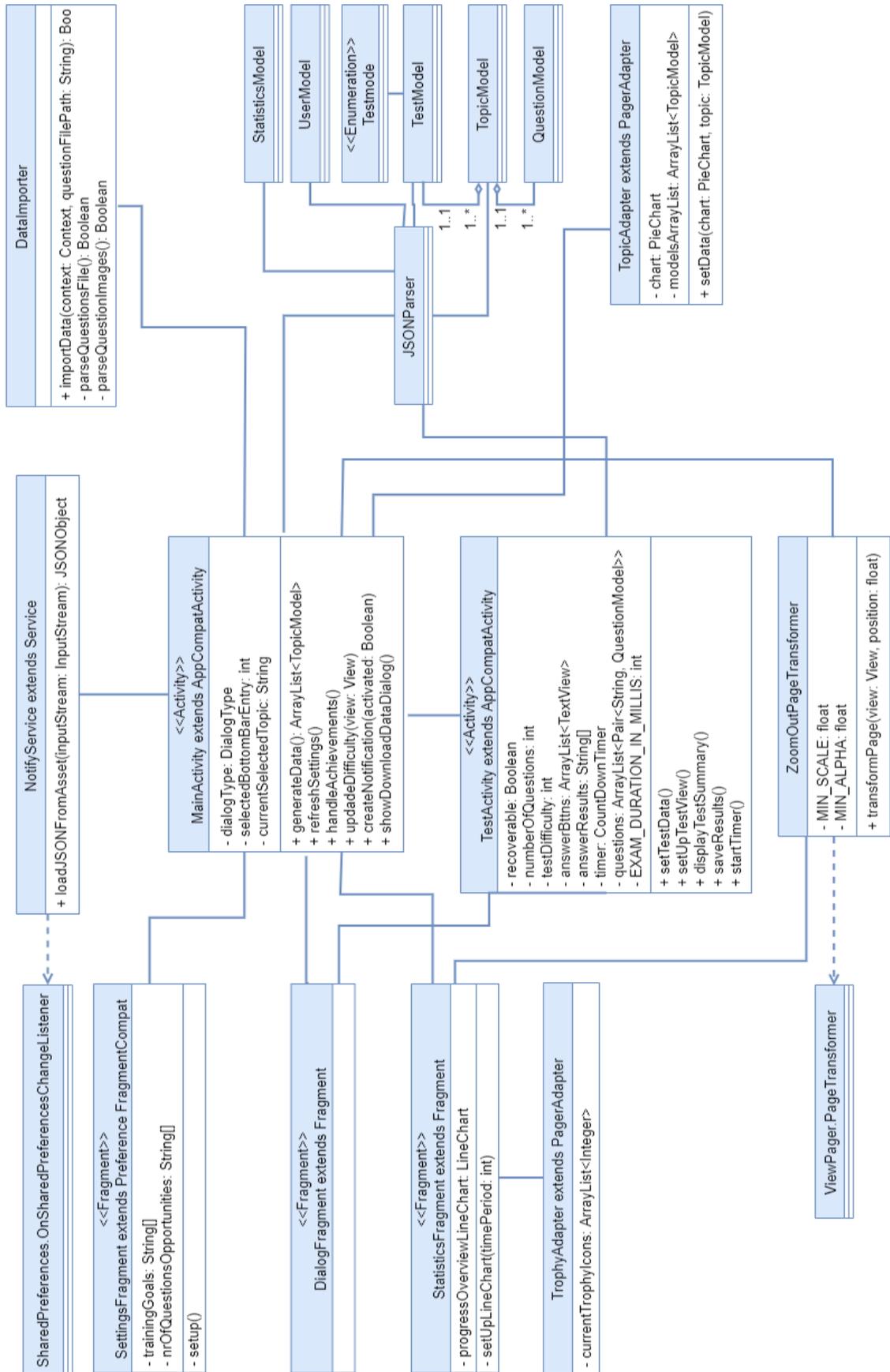


Fig. .3: Class diagram of main and test activity in detail.

.3 Use case diagram

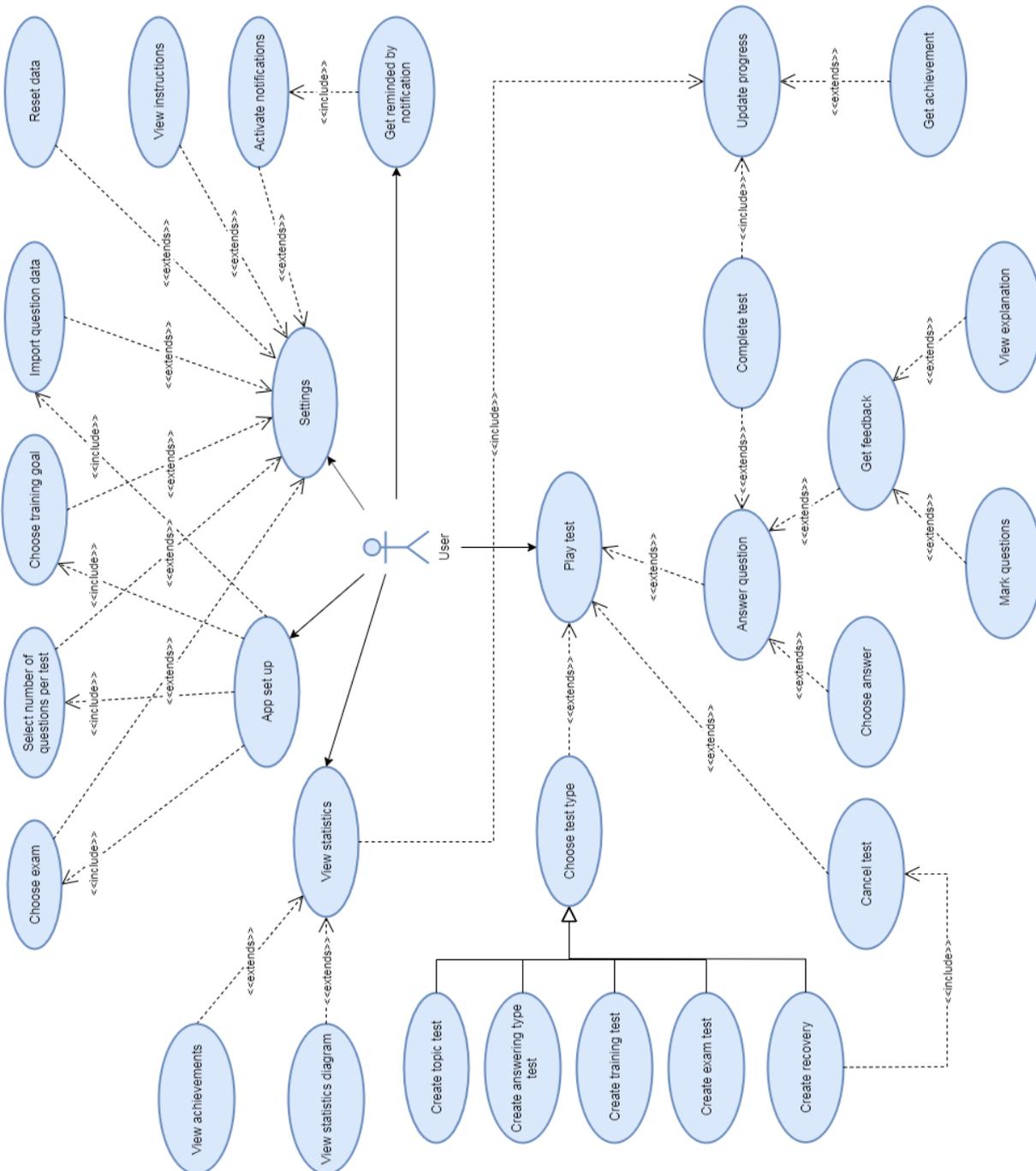


Fig. .4: General use case overview

.4 Dialog structure

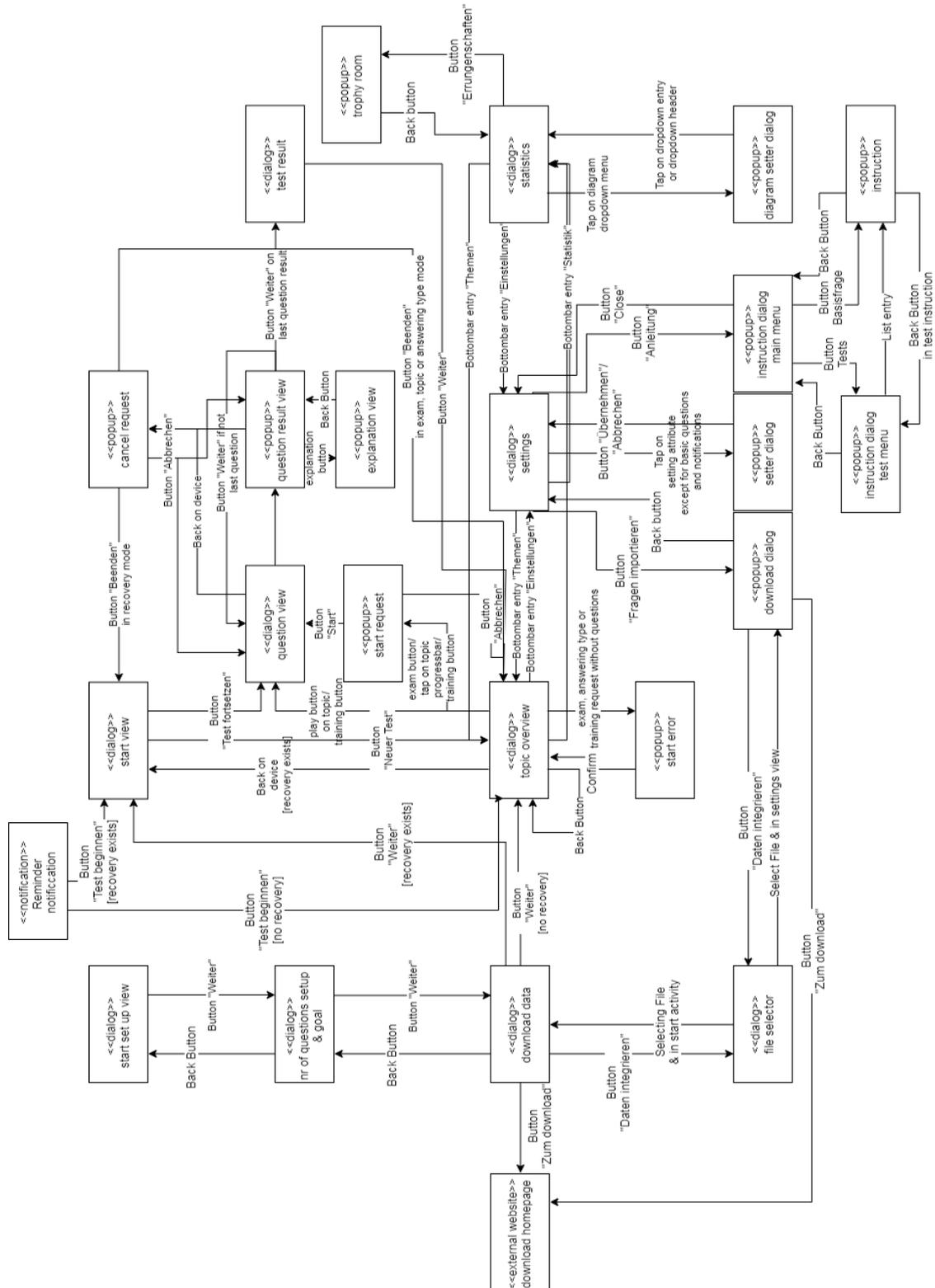


Fig. 5: Overview of all relations and transitions between the applications interfaces

List of Figures

2.1	Block architecture	5
2.2	Use case diagram of the MoBiLe application	7
2.3	Use case diagram of the MoBiLe application	7
2.4	Use case diagram of the application start	8
2.5	Typical use sequence of a test session	10
2.6	Interaction possibilities with the statistics view.	11
2.7	Interaction possibilities with the statistics view.	12
2.8	Determined user needs and discussed solutions	15
4.1	General dialog structure	32
4.2	Start view for first use	34
4.3	Start view for first use	35
4.4	Topic list overview	36
4.5	Topic list overview: exam start request dialog	37
4.6	Topic overview tab if recovery exists	38
4.7	Topic overview with missing questions for a specific difficulty	39
4.8	Topic overview with missing questions	40
4.9	Statistics overview: upper part	41
4.10	Statistics overview: bottom part	42
4.11	Statistics overview: trophy room	43
4.12	Settings overview	44
4.13	Settings overview: download questions dialog	45
4.14	Settings overview: instructions main menu	46
4.15	Settings overview: instructions tests menu	47
4.16	Settings overview: instruction interface (top)	48
4.17	Settings overview: instruction interface (bottom)	49
4.18	Test: question interface with no response selected so far	50
4.19	Test: question interface after answer has been selected	51
4.20	Test: feedback after response confirmation	52
4.21	Test: explanation view of the current question accessed through the result dialog	53
4.22	Test summary which is shown after completing a test.	54
4.23	Achievement reached notification which appears when returning to the main view after a test session.	55
4.24	Cancellation request which appears during a test session, if the back button has been pressed.	56
4.25	Structure of a reminder notification.	57
.1	Class diagram of all classes.	62

.2	Class diagram of start activity and context in detail.	63
.3	Class diagram of main and test activity in detail.	64
.4	General use case overview	65
.5	Overview of all relations and transitions between the applications interfaces . . .	67

List of Tables

2.1	Overview about all used gamification elements according to Majuri et al. [1] . . .	13
2.2	User type summary	16
3.1	Non functional requirement summary	29
.1	Overview of the relevant data of the application	60

Bibliography

- [1] J. K. Jenni Majuri and J. Hamari, “Gamification of education and learning: A review of empirical literature,” 2018.
- [2] J. H. Benedikt Morschheuser, Karl Werder and J. Abe, “How to gamify? A method for designing gamification,” 2017.
- [3] N. Brenner, “Mobile app: Conception and realisation of mobile serious games for learning support in biochemistry with the ios operating system,” 2019.