

# Konzeptreview

**Team: Review 5-1**

*Johannes Kraml 01116375*

*Markus Ninz 11727598*

*Linus Wald 11722746*

**Proseminargruppe: Gruppe 3**

**Datum: 28.03.2019**

# 1. Overview

The presented system is a web-based multiplayer online quiz game meant for teaching in schools. It uses a concept, where each player gets a question and the answers (wrong and right) are distributed between all users. Thus, the players have to work as a group to answer all questions.

Overall, the concept seems very sound, with only two major issues and a few minor issues. But in general, the group addressed all needed tasks and the concept is already in a quite good state. The group failed to mention the time limit at any point in the concept and should revise this. The GUI is very well designed and easy to understand.

# 2. System Overview

The system overview, presented by the group is very clear and concise. It is clear, which kind of users they want to address (pupils). Except for a few points, the system overview addresses each instruction given.

The GUI, presented as point 6, mirrors the concept quite well and complements the system overview given at this point.

The biggest point is that at no point in the overview, and also only at one point in the entire concept, the group actually mentions time limitations for the questions. As this seems to be a very central point to the program, they should address this more often.

# 3. Use Cases

The use cases seem quite well defined and, in our opinion, cover all instructions, with the exception of the aforementioned time limitation.

The diagram for the use cases is missing and only a list of the use cases with the main actor at the beginning is given. This method does not help the reader, it would be nice, to either include a diagram or write a short description of many (hard to understand) use cases next to them.

In the main text the use cases are quite well defined, to the point, where some of the seem overzealous, e.g., Select QuestionSet. It would also be better to split the "Create Manager Account" into two different use cases, "Create Manager Account" and "Manager Login", since "Manager Login" and "Manager Logout" are missing completely. The group misses some basic use cases, like the editing of the player by himself, it is mentioned in the "Change Own Password" use case that there is something called "Benutzereinstellungen", but otherwise these are never mentioned.

The use cases are sometimes called differently in the main body of text compared to the list at the beginning, e.g., quite often the name Quizroom is replaced with only room, some of the use cases have more elaborate names in the main body of text.

For the manager, login and logout are not described and it is not explained how it is different to the player's login, if at all. The same holds true for change/reset own password.

The player seems to have no chance, by himself, to reset the password, this could, however, be an intentional security measure (mostly because it is in a school).

The use case for editing a question set is mentioned in the beginning, but not described in the main body of text.

## 4. Class Diagram

The class diagram is very well structured. The use cases and class diagram complete each other and define tasks quite well. The diagram realizes the concept of the project in a defined way. Thanks to the description of the different classes is it easily understandable.

However, the class ActiveQuestion which is included in the diagram isn't explained or mentioned at all. There it would be nice to write at least a sentence to explain the class. Moreover, we noticed that the cardinality is missing completely.

Overall we think that the class diagram has a clear structure and is very well thought out.

## 5. SW-Architecture

The sw-architecture lists the components (main actors) sufficiently. The only documentation given is the component diagram. The component diagram seems not to be completely correct. As shown in the diagram the user is the only connection to the database. So even the game and the UI seem to get all the data from the user.

Therefore, the component diagram isn't consistent with the other parts of the concept. Furthermore, the components should be described. We think the sw-architecture still needs a bit of work and the component diagram should be over thought.

## 6. GUI Prototype

The GUI already has a clear and well defined structure. They are also in agreement with the use cases and the diagrams.

The GUI seems very clean and easy to understand. It seems that the group tried to implement an easy usability. It is very understandable how to navigate through the system and the concept by itself is very easy to understand.

The design uses usability concepts such as clarity and operability. It uses a simple structure which offers clear configurations.

## 7. Project Plan

No detailed tasks were distributed to the individual group members (at least for the first few weeks). The issues in git could be divided into smaller subtasks. Timetable seems to be reasonable and the task are evenly balanced over time

Minor: Use Cases	A good idea would be to introduce the use cases either in a diagram or as a list, like it was done here, but with additional information at each point.
Minor: Use Cases	Switches between German and English sometimes occur and are not 100% understandable.
Minor: Class Diagram	Cardinality for the different sets would help.
Minor: Project Planning	The timetable could have been more detailed and fine grained.
Minor: Project Planning	The last bullet point of milestone 5 seems utopian.
Medium: SW-Architecture	The connections between the different components are not very well defined and some are missing, e.g., Game seems to get persistent questions from the user, also, the UI seems to get all data from the user.
Major: SW-Architecture	The only documentation given is the component diagram, which seems not to be completely correct.
Major: Use Cases	Some of the use cases mentioned are not at all described in detail.
Major: System Overview / Whole concept	Time limit is missing completely