ASSIGNMENT 2
SYSTEMS ANALYSIS AND DESIGN

REQUIREMENTS DOCUMENT

Maroun AlAlam, Pham The Nam, Do Tran, Abdullah Akram, Matthew Wedding
| ZEIT2104 | 03/11/2019

The requirements for this app have been broken down into the functional and nonfunctional requirements and are as follows.

FUNCTIONAL REQUIREMENTS

- It must be a PC application.
- The application should be able to add a student to the database, including at least their first name, last name, date of birth (DOB) and which class they attend.
- A student's attendance must be recorded for each session (a session can be either a weekly meeting or a weekend intensive course).
- The app's database must be able to record the parts, topics and tests completed during the meetings.
- The app's user (staff) should be able to search for each student individually through a number of ways. A few such examples include, the student's unique student number, name, age or date of birth.
- Additionally, the teacher should be able to view the student's personal details and achievements.
- Each student is required to be assigned a unique student number to enable quick and easy identification. This is essential as it can be very difficult to distinguish between two students if they both have the same name.
- An additional inclusion to the app will be a leader board. The leader board will be used to provide a visual illustration of how each of the students are performing.
- There must be a "badge hierarchy" which will be used to differentiate the students on a merit-based system.
- Each test, topic and part should be assigned a unique code to enable quick and easy identification purposes.
- It is required that the user is able to login to the app, thereby ensuring the privacy of their profile.
- The application is an offline application.

NON-FUNCTIONAL REQUIREMENTS

- The users will be added to the database when they initially sign up to the app. Their details will be updated whenever the user chooses to.
- The student's attendance will be recorded by the teacher entering the name of the students who are in attendance and their class.
- The information regarding what parts each student has completed will be manually input by the teacher. The system will then automatically update the topics, tests and badges the student can achieve and save this information in respective tables.
- To search for a student, the teacher will be asked to input a piece of information to specify the student. This information could include the student's name, the student's age or their student ID number. The order that the information will be inputted is the student ID number, the student's first name, their last name and finally, the student's age. If a piece of information is unknown the teacher will simply press the "enter" key to move to the next search field. After a search field is successfully filled, the system will return all of the student's whose information matches the search request.
- After having successfully searched for a student, the system will return a page filled with all of the information relevant to the student that was searched. Included among the returned pieces of information will be the student's personal details and achievements.
- A student record can be deleted using the student's ID in the Student table, followed by updating other tables' relevant data in order to prevent data redundancy. It is noted that the attendance of deleted students will not be changed.
- The software should run on Windows OS.
- The database will be stored locally (offline) on the hard drive.
- There is one database with 16 tables.
- The software features a command line interface.

PERFORMANCE REQUIREMENTS

- The time taken for the system to provide a response to a search query is ±0.5 seconds.

- The start-up time is less than 1 second.

AVAILABILITY REQUIREMENTS

- The system should be available twenty-four hours a day, seven days a week.

SECURITY REQUIREMENTS

- The design should avoid SQL injections.
- Login credentials are required to access the application.

PRIVACY AND ETHICAL REQUIREMENTS

- Only the app users which are the staff have access to the application data.

ASSUMPTIONS

- It is required that one session is managed by one teacher.
- It is required that a student has to do three parts to finish one topic. From here, three topics are required to be completed to finish one test. Finally, at least ten tests must be completed to get a badge. The ten tests that are completed must include seven compulsory tests and three optional tests.
- It is also required that three topics are required to be successfully completed to finish one test.
- To aid the teacher in identifying the "less popular" pieces of assessment, any topic within a test that is completed by less than 20% of the number of students who completed the most popular test will be highlighted to the instructor.
- The format of the ID is as follows: a badge will be assigned numerals in the format "1". The test will be assigned numerals in the format of "1.1", a topic within a test "1.1.1" and a part which is contained within a topic will be in the format of "1.1.1.1".
- It is also specified that the badge scoring system will be broken down as follows. Any student who has between 1000 and 3000 badges will be within the gold category. From 3001 to 5000 badges will be the diamond category and any student with more than 5001 badges will be within the platinum category.
- All students are divided into 5 classes.