

ASSIGNMENT 2

Advanced Software Engineering



Presented to: Dr. Islam El-Maddah

Made by:

* Khaled Bahaa El-Din (2101397)
* Abdullah Aml (2101398)

**# Assignment 2 statement**

**it is required to build a software application in:**

Virtual Reality Educational game based on topics from one course, **students have levels** and the **course is divided into stages** **and each student must complete all stages before he proceed to the final stage**, the course stages can be edited and updated by course instructors.

**Answer:**

**System actors:**

* Course instructors.
* Students.
* System admin.

1. **User stories:**

|  |  |  |  |
| --- | --- | --- | --- |
| Story ID | As a/an | I want to… | So that… |
|  | Student | Register in educational game | I participate in course curriculum. |
|  | Student | Get notified with updates made on the system | I can check recent updates made by course instructors. |
|  | Student | See all course stages before starting | I can manage my time correctly. |
|  | Student | Have easy controls | I can navigate the game easily. |
|  | Student | Complete all the game stages | I can proceed to the final stage and pass the course. |
|  | Student | Get a second chance if I made a mistake | I can get high grades. |
|  | Student | Receive email with final grades | I know whether I succeed or not. |
|  | Student | Pause the game | Can encounter for sudden interrupts |
|  | Course instructor | Make quizzes inside the game | Measure the student’s understanding. |
|  | Course instructor | Get notified when a student finishes the final stage | I can check their score and approve his grade. |
|  | Course instructor | Edit/update the course material | I can modify the game stages. |
|  | Course instructor | Get a list of scores of all students at the end of the semester | I can calculate the success percentage. |
|  | Course instructor | Be able to approve the final results for all students taking courses | The system start sending mails to them. |
|  | System admin | Monitor the system | I make sure the system is working properly. |

**2.1- Functional requirements:**

|  |  |
| --- | --- |
| Requirement ID | Requirement description |
|  | All the topics present in the game must be stated in the course. |
|  | The system of the game shall allow editing the game to course instructors only. |
|  | Student level shall increase if and only if they perform the required tasks within a stage. |
|  | Only authorized students shall have access to the game. (The authorized students are students that registered for the corresponding course). |
|  | The game system shall prevent any kind of cheating. (e.g. by adding login system to make sure the student himself is accessing the game) |
|  | A student cannot pass their current stage in the game unless all the stage objectives are done. |
|  | No student shall reach the final stage unless all the previous stages are passed on their profile. |
|  | The final stage is not accessible for a student until the student level meets minimum allowed. (e.g. level 10). |
|  | The game system shall automatically generate a report once detected any cheating. The report must include student details and how was cheating detected. |
|  | No student is allowed to participate in the game after the semester is finished. |

**2.2- Non-functional requirements:**

|  |  |
| --- | --- |
| Requirement ID | Requirement description |
|  | The game shall not fail during runtime for any reason. |
|  | Safe login/logout must be maintained. |
|  | The game may lag due to poor internet connection. But lag shall not exceed 200ms. |
|  | The game system shall preserve a history of student’s grades and levels for future reference. |
|  | The game system may allow the students to access the game anytime. |
|  | If an update was issued during runtime (when the student is already inside the game), the student has to be prompt by update so that he can leave the game within a given timeout. |
|  | The figures in the game shall be friendly and interactive to student’s gestures. e.g. no violence is allowed by any means |
|  | The game system shall be able to receive all the students changes at the same time and update itself correspondingly with no delay. |
|  | The game system shall have considerations for students having motion sickness. They shall have different visualizations accustomed to their mental state. |
|  | The game system must be reliable enough to let the students play the game flawlessly. |

**3- Stakeholders:**

* **Student**: This is the person that actually play the game.
* **Course** **instructor**: The one responsible for editing and updating course materials and game stages. Also, responsible for approving the final grade.
* **System Admin**: The one responsible for maintaining system functionality and performance.

**Use case diagram:**

**Use case descriptions:**

|  |  |
| --- | --- |
| Use case | Communicate with students |
| Actor | Course instructor |
| Trigger | Several students start playing the game related to the course. |
| Pre-condition | Game system allow communication between course instructor and students |
| Post-condition | Both students and instructor perform effective communication |
| Main scenario |  |
| extensions |  |

**Additional information:**

* Graphical user interface, application

  Description automatically generatedComparison between functional and non-functional requirements:

**References:**

* [The app solutions](https://theappsolutions.com/blog/development/functional-vs-non-functional-requirements/)
* [Occulus Rift S](https://www.youtube.com/watch?v=RuiqRQAYaeA)