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 proceeds to the final stage**, the course stages can be edited and updated by course instructors.

Notice: We used git hub (as a version control tool) to synchronize our edits together. Please visit the repository: Github Link.

Answer:

Initial assumptions:

- At the beginning of the semester, students register the course through the dashboard system.
- Once their registration request is made, the System Admin ensures that student's registration is compliant with the laws. If it is, registration is done successfully.
- Then, Instructors are notified with the registered students to communicate with them.
- Instructors can edit/delete/modify stages within the course (following CRUD metric).
- On each modification from the instructor, students are notified by their dashboard.
- The game types within the system are (design, exploration, assembly and puzzles).
- Each instructor specifies the suitable game type related to their course.
- The VR game system is connected to the database that contains details to specific course.
- Student can enroll in multiple course games.
- The game consists of multiple stages.
- Each stage consists of materials to browse and puzzles to solve.
- Once a student completes the stage, he can move to the next one to complete the course game.
- If the student failed to solve the puzzle more than three times the system.
- Restart the stage and multiplies the score by decreasing factor.

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.

Basic Design Thinking (not covering all cases)

Instructor

Phase	description
1. Empathize	- Instructor needs to monitor students' easily. - Instructor wants to chat with students taking course.
2. Define	- A table with student progress can help the instructor greatly. - Instructor can be notified when someone sends them a message.
3. Ideate	- The table with students' progress with filters can be shown in the dashboard - Notification panel can blink when a new message arrive.
4. Prototype	- Such design can be made.
5. Test	- Feedback of the instructor after seeing prototype is the test result.

Student

Phase	description
1. Empathize	Student can get perplexed due to strange objects inside the VR game.
2. Define	Student needs obvious metaphors inside the game to react upon.
3. Ideate	Using common internet metaphors will be straight forward.
4. Prototype	Some figures can be designed in VR.
5. Test	A student can see the designed figures so we can know whether he got to the point or not.

Admin

Phase	description
1. Empathize	Admin needs to monitor everything in the system.
2. Define	Admin needs to be notified immediately with system unexpected behavior.
3. Ideate	Push notifications can catch admin's attention immediately.
4. Prototype	A push notification routine can be designed for testing.
5. Test	Get admin's thoughts about the notification tone and message.

User stories

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

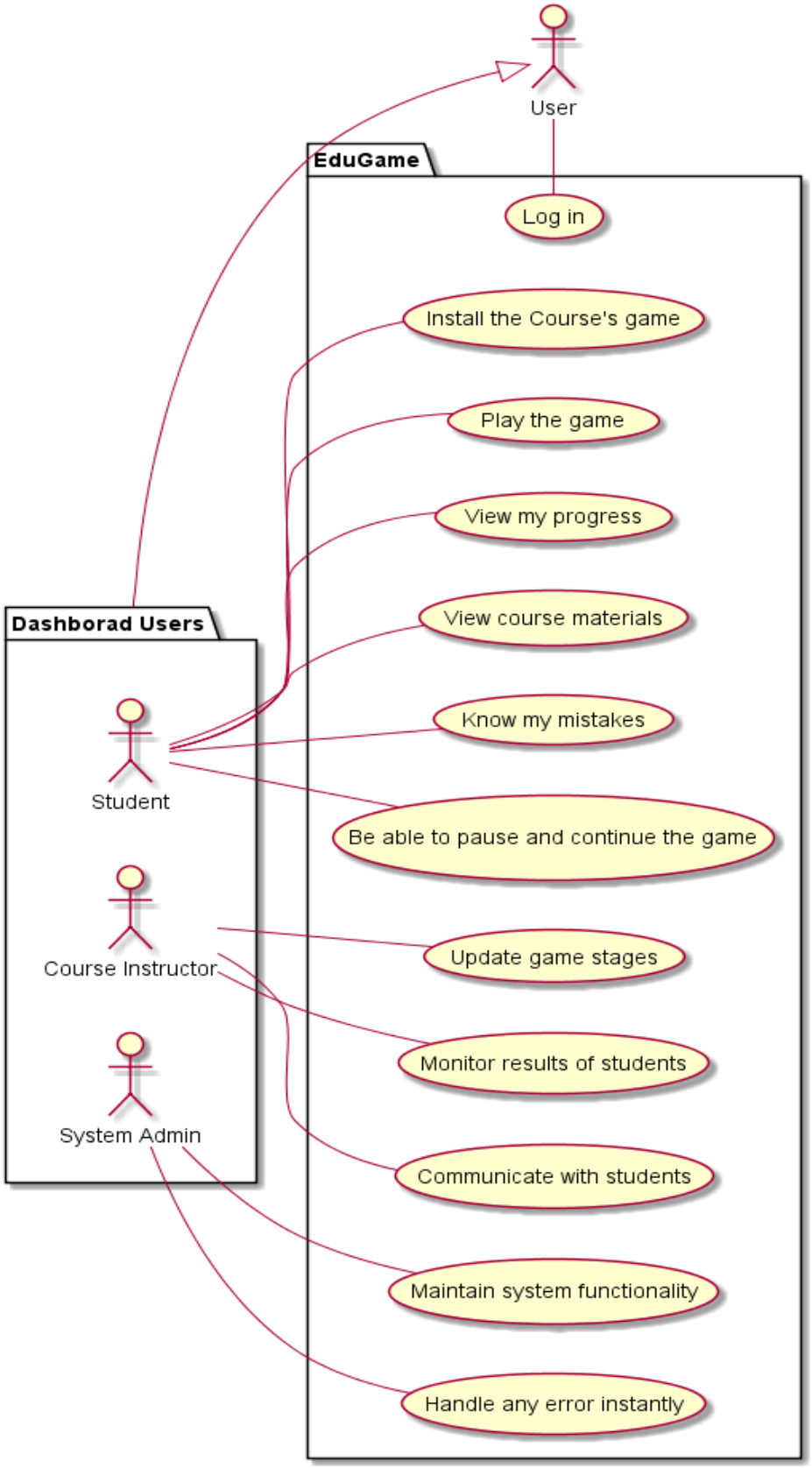
Functional requirements

Requirement ID	Requirement description
1.	All the topics present in the game must be stated in the course.
2.	The system of the game shall allow editing the game to course instructors only.
3.	Student level shall increase if and only if they perform the required tasks within a stage.
4.	Only authorized students shall have access to the game. (The authorized students are students that registered for the corresponding course).
5.	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)
6.	A student cannot pass their current stage in the game unless all the stage objectives are done.
7.	No student shall reach the final stage unless all the previous stages are passed on their profile.
8.	The final stage is not accessible for a student until the student previous stages are cleared.
9.	The game system shall automatically generate a report once detected any cheating. The report must include student details and how was cheating detected.
10.	No student is allowed to participate in the game after the semester is finished.

Non-functional requirements

Requirement ID	Requirement description
1.	The game shall not fail during runtime for any reason.
2.	Safe login/logout must be maintained.
3.	The game may lag due to poor internet connection. But lag shall not exceed 200ms.
4.	The game system shall preserve a history of student's grades and levels for future reference.
5.	The game system may allow the students to access the game anytime.
6.	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.
7.	The figures in the game shall be friendly and interactive to student's gestures. e.g. no violence is allowed by any means
8.	The game system shall be able to receive all the students changes at the same time and update itself correspondingly with no delay.
9.	The game system shall have considerations for students having motion sickness. They shall have different visualizations accustomed to their mental state.
10.	The game system must be reliable enough to let the students play the game flawlessly.

Use case diagram

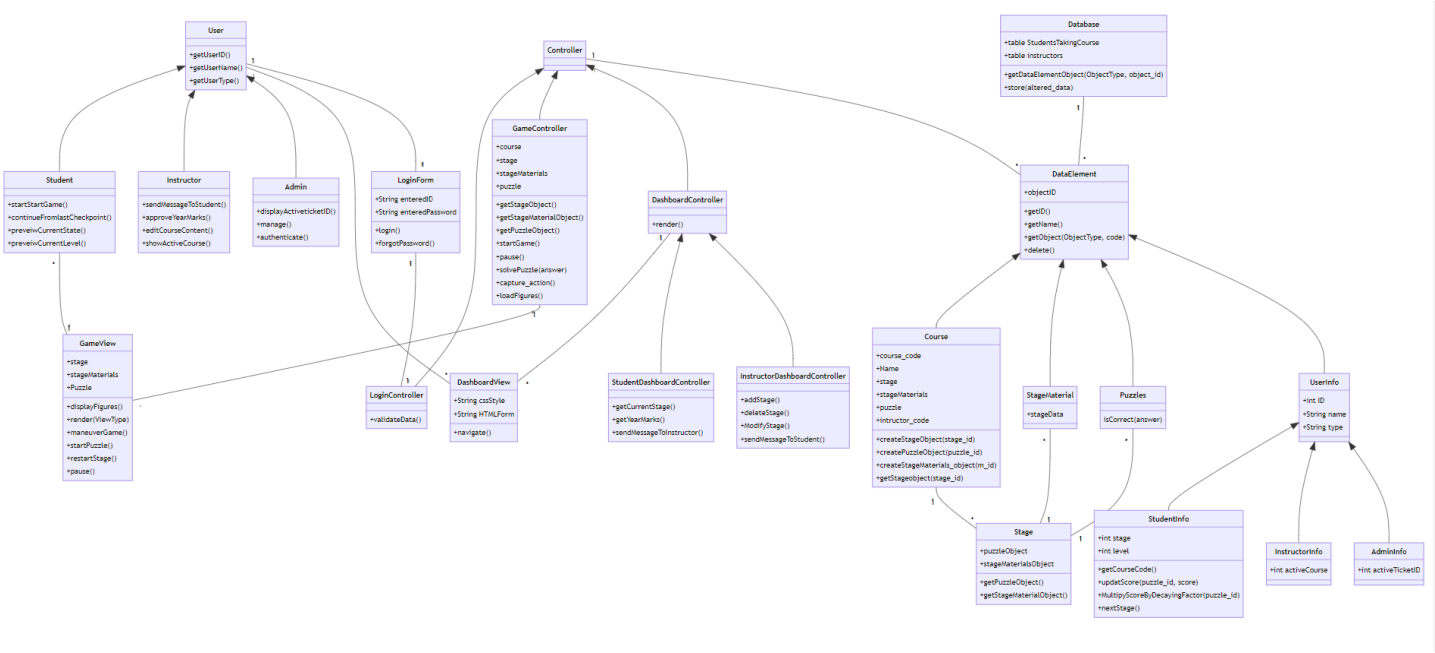


Use case descriptions

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	1.	Course instructor (CI)	Log in to the system
	2.	Game system (GS)	Check input credentials and deduce that the user is an instructor. Then display the instructor dashboard.
	3.	CI	Check if there is any student tried to reach the instructor.
	4.	GS	Provide a notification panel that contain chats from students.
	5.	CI	Read students inquiries and reply to them.
	6.	GS	Provide a clear UI that contain student name and his message. Then transmit the instructor's answer and finally notify the student.
	7.	CI	Send a global message to all students to notify them with latest changes.
	8.	GS	Provide a global message option that broadcasts instructor's message across chat server.
	9.	CI	Categorize students' messages according to content.
	10.	GS	Provide filters in the chats section to categorize messages.
Extensions	3a. If there are no students tried to reach the instructor, the chat window has to display "No messages yet". 5a. If any student written an insult or improper words, the system replaces such words with asterisk '*'.		

Use case	Play the game		
Actor	Student		
Trigger	The semester starts		
Pre-condition	Student installs the game, and log in		
Post-condition	Student finishes a stage or, multiple ones		
Main scenario	1.	Student (S)	Press start button
	2.	Game system (GS)	The game system renders the student stage
	3.	S	explore/design tasks in the game, and browse course materials
	4.	GS	Fetches the course materials form the dashboard database and render it to virtual reality glasses.
	5.	GS	The GS loads quiz and question form the dashboard database
	6.	S	Answers the questions and continue the game.
	7.	S	Finishes all the materials, and answers all questions
	8.	GS	will move the student to the next stage, stores the score, and progress of the student to the dashboard database
Extensions	1a. the virtual reality (VR) fails to connect to the server 1b. GS will pop up a check internet connection message 3a. If S is disconnected 3b. when internet is back, GS will render the game of the last stop 5a. S fails to answer the question 5b. GS will give S 3 trials to answer the question. IF S fails again GS will restart the stage again, and multiplies the score by a decreasing ratio ex (.9, then .8 next reset and so on)		

Class diagram

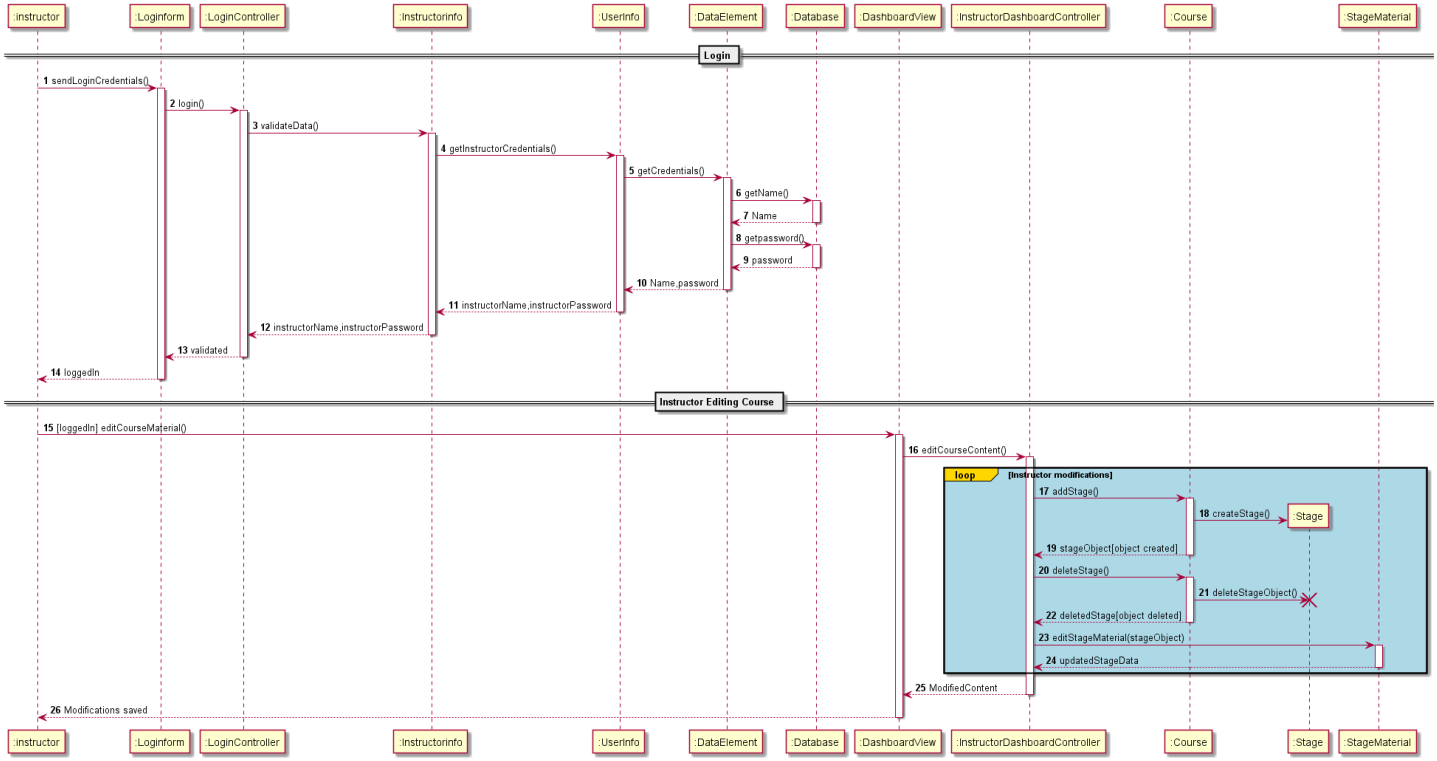


For high quality image please visit: [Link]

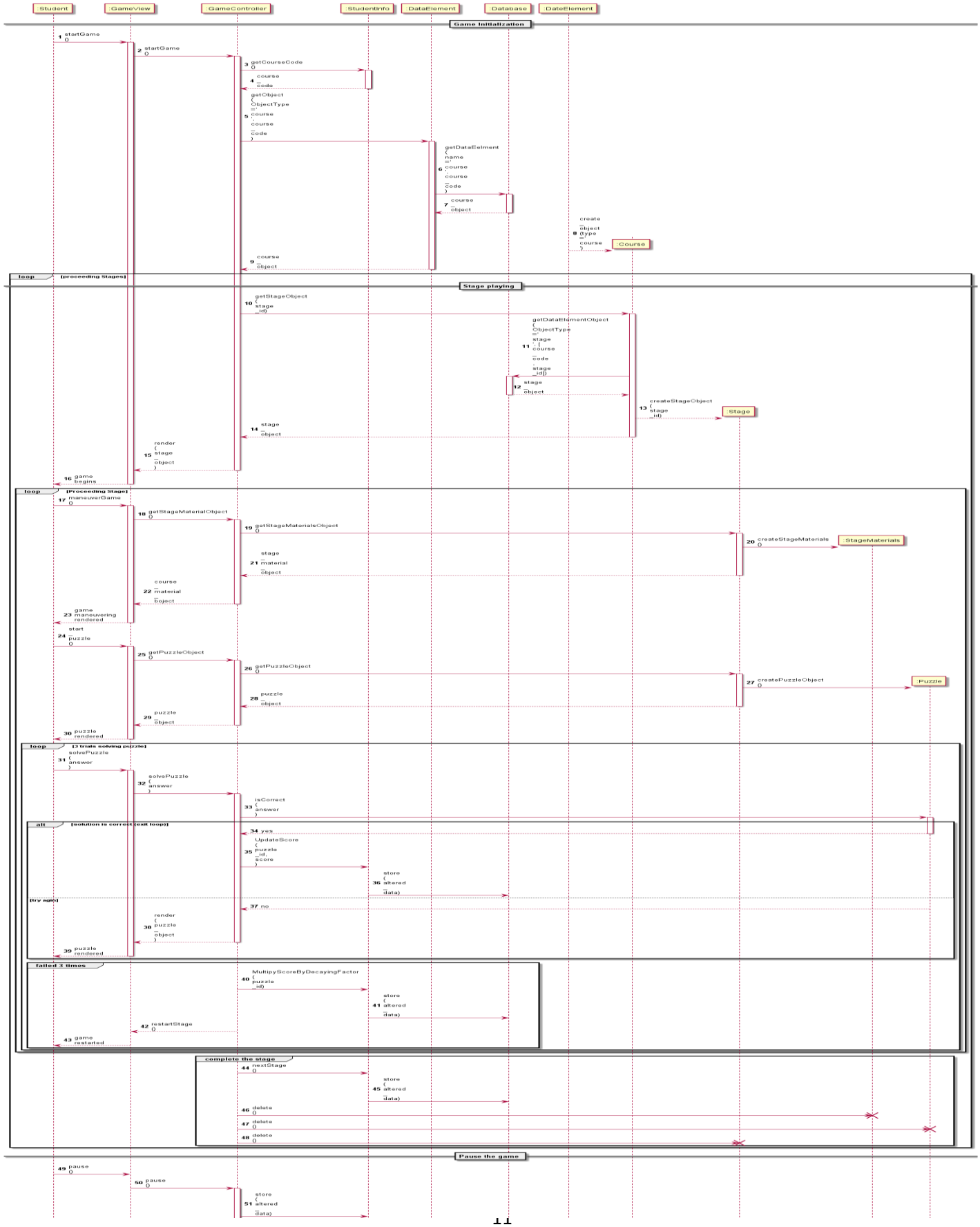
Sequence diagrams:

For the sequence diagrams, we split the system into two main sub-systems:

Dashboard sequence diagram: [Link]

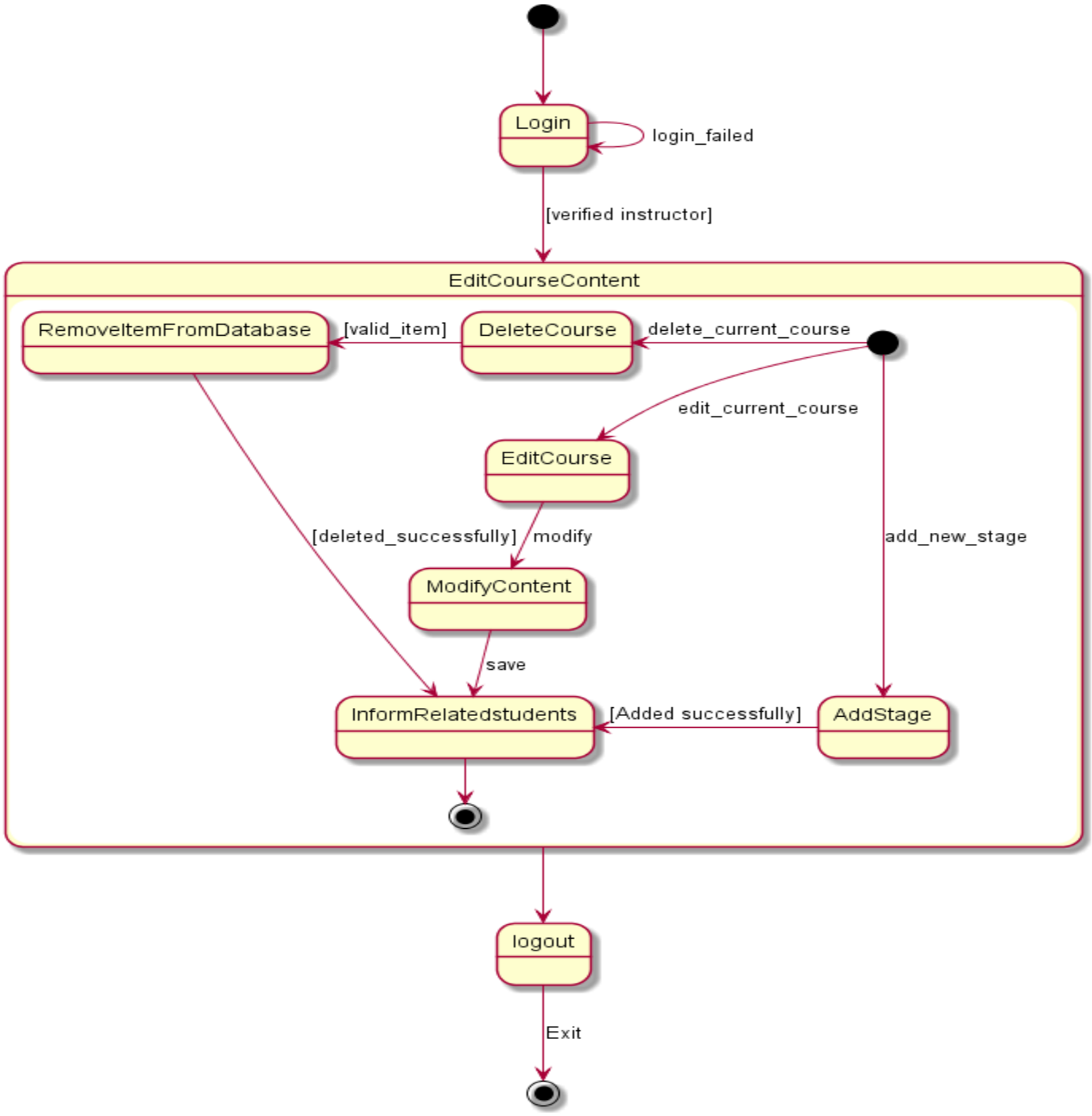


Game sequence diagram: [Link]

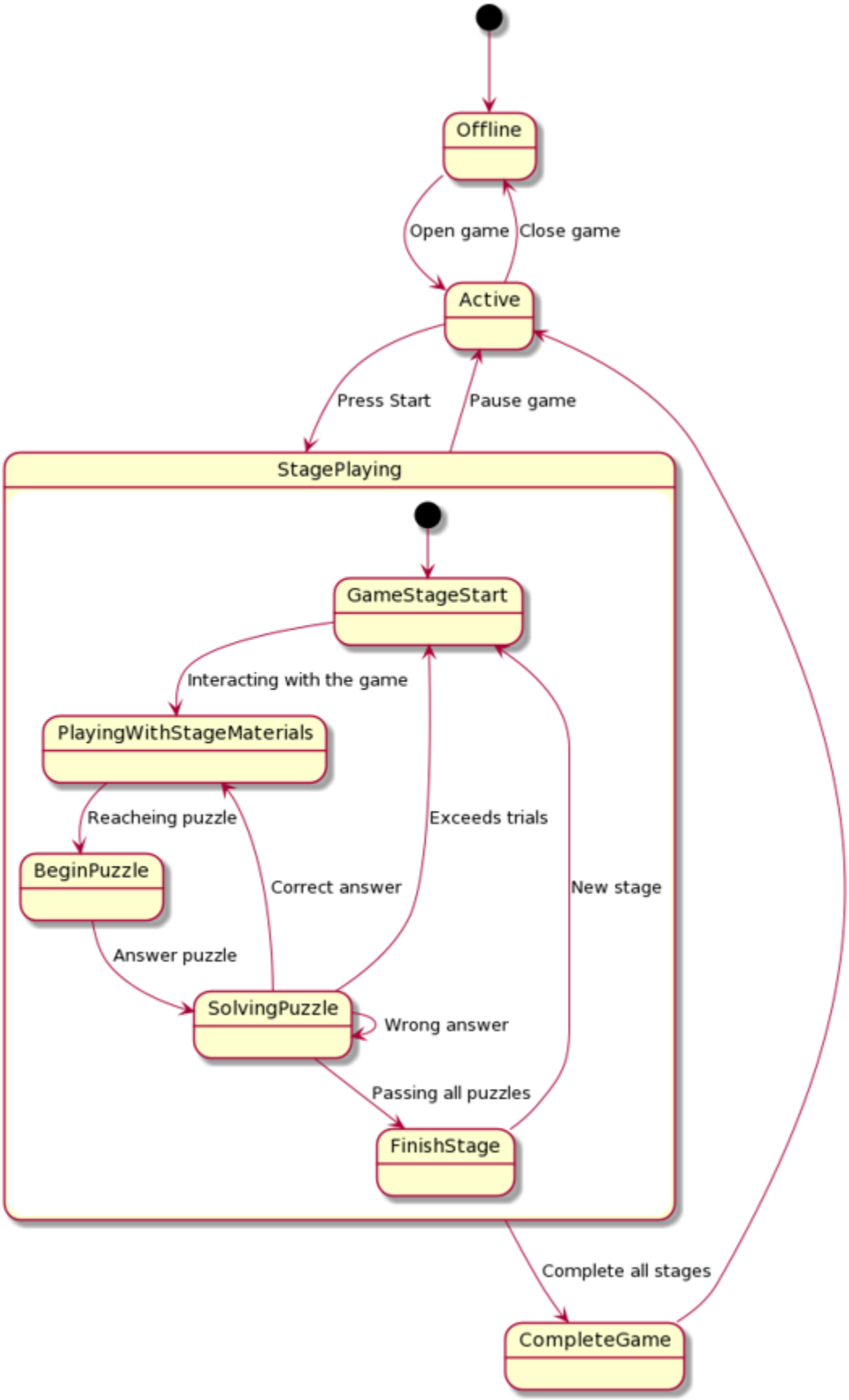


State diagrams:

Dashboard: [Link]



Game: [Link]


Offline
Open game
Close game
Active
Press Start
Pause game
StagePlaying
GameStageStart
Interacting with the game
PlayingWithStageMaterials
Reacheing puzzle
Exceeds trials
BeginPuzzle
Correct answer
New stage
Answer puzzle
SolvingPuzzle
Wrong answer
Passing all puzzles
FinishStage
Complete all stages
CompleteGame

Short research about main system users UX

Instructor

- The instructor age is between 20-40.
- The instructor should have material developing experience.
- The instructor should be familiar with technology. If not, there will be hand on session to get the instructor ready.

Student:

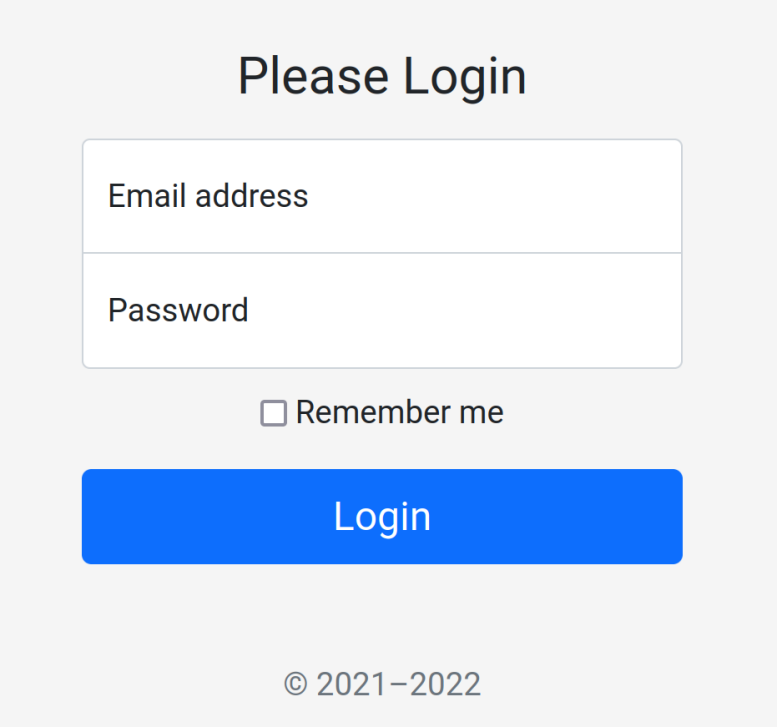
- The student's age is between 15-20.
- The student is expected to be familiar with smart-phones, and technology, and if not, there will be an intro game to make the student familiar with the game.
- The student is not supposed to have basic scientific knowledge (can read, write, apply basic mathematical operations.

System Admin

- The admin should have two years of experience.
- The admin sets all time on disk so interface should be reliable and consistent.
- Age is not a concern if there were an experience.

UI screens

Login screen:



10.2 Game menu for VR biology course for a student at stage 1

References

- PlantUML(script drive UML maker)
- The app solutions
- Occulus Rift S
- Prof. Barry Boehm quotes
- User research