# 

**Space Learning Virtual Reality**

**Xavier Souffront, Muhammad Salman, Andre Simmons, and Fredy Diaz**

# **Table of Contents**

**1. Introduction**

1.1 Background

1.2 Problem Statement

**2. Feasibility Study**

**3. Proposed System Requirements**  3.1 Functional Requirements

3.2 non-Functional Requirements

**4. Requirement Model**

## 4.1 Use Cases

## 4.2 Use Case Diagrams

## 4.3 Detailed Use Case Specification of Core Features

## 4.4 System sequence diagram

## 4.5 Task assignment and schedule

**5. Design of User Interface**

**6. Implementation**

## 6.1 Core modules

## 6.2 User management

## 6.3 Multiple user connection

## 6.4 Prototype

## 6.5 Code repository

**7. Design of Tests**

**Appendices**

**1.Introduction**

**1.1 Background**

Space Learning VR,There will be a room replicating a museum environment. In this room there will be a panel with a few buttons, On or Off, planetarium, and the exhibit. The panel is controlled by the user, they have the ability to change the exhibit which will consist of a few planets/ stars. If the user clicks the on button for Earth, Earth will be projected in the exhibit. Beside the Earth there will be a window that will pop up with information about that planet. The user can control the rotation and zoom into the Earth. Users can walk around the room as well. Users will also be able to take a quiz based on what they previously read about each planet. These quiz scores will be recorded and used to create a leaderboard. Users will be able to go to a room and interact with objects that replicate a planet's physics, such as if they are weightless.

**1.2 Problem Statement**

Museums are very helpful in helping people learn various things by providing exhibits to visitors. Teachers often go on field trips with students to museums to enhance their teachings which is very effective. However, there are some limitations to this. Not all students may have the ability to go on the field trip, so we will create a virtual environment replicating one, Space Learning VR. Museums lack the ability to engage with visitors, especially students, due preserving and maintaining their exhibits and artifacts. SLVR plans on making the virtual space museum more engaging by allowing visitors to select which planets they would like to learn about, rotate and zoom into the planets, take quizzes on what they are being taught, enter planetariums, and go into a room that will mimic the gravity of the planet they choose.By using Unity we create multiple 3D objects that will represent planets and create different rooms for the planetarium and room that will mimic the gravity. Each room will be made in a new scene from unity. The rotation and zooming of objects will be done by using the “transform.Rotate” function and the “GetMouseButtonDown()”function. The quizzes will be made by downloading and using the MaterialUI package. Visitors will be able to move around by using the asset “GvrEditorEmulator”.

**2 Feasibility Study**

**Operational feasibility:**

There will be a room replicating a museum environment. In this room there will be a panel with a few buttons, On or Off, and the exhibit. The panel is controlled by the user, they have the ability to change the exhibit which will consist of a few planets/ stars. If the user clicks the on button for Earth, Earth will be projected in the exhibit. Beside the Earth there will be a window that will pop up with information about that planet. The user can control the rotation and zoom into the Earth. Users can walk around the room as well.

Space Learning will allow users to change the planets that they are learning about, they can touch and move the planet, they can move around the space. We plan on adding more features to make it more like a space museum. No training is needed for Space Learning, the only instructions provided will be how to move and touch things, the rest will be self-explanatory. These instructions will be displayed before the user enters the app. Space Learning will help users understand facts about the planets and other space facts that we provide.

**Technical feasibility**

*VR functions:*

* Will allow multiple users to walk around the space.
* Users can click buttons to change which planet they want to learn about.
* allow users to trigger an event
* When users enter the planetarium we will play a short movie.
* Planets will rotate on a platform.

*VR modeling:*

*VR objects*

* Planets
* Consoles
* Buttons

*VR environment*

* Planetarium
* Planet Exhibits
* Quiz Room
* Gravity Simulator
* Planets

*VR avatars*

* Visitors
* Students
* Teachers

Yes, Unity offers enough features for this team to develop this app. We all searched for tutorials we can refer to if we don’t know how to create them.Google Cardboard is enough because we will take advantage of the reticle. Will need a rigid body for the users and objects created so it can simulate an actual museum.

**3. Proposed System Requirements**

## 3.1 Functional Requirements

**The Exhibit System Requirements (Module 1)**

|  |  |  |
| --- | --- | --- |
| **Number** | **PW** | **Requirement** |
| **REQ1** | **1** | **Console has buttons that will display a planet on the exhibit platform** |
| **REQ2** | **1** | **Another console has buttons that will take users to different rooms** |
| **REQ3** | **2** | **An info screen pops up beside the planet** |
| **REQ4** | **2** | **The planet rotates and can be zoomed into** |
| **REQ5** | **3** | **Multiple choice quiz at the end of the info screen** |
| **REQ6** | **2** | **Users can move around room** |
| **REQ7** | **3** | **When user leaves console area ground changes color** |
| **REQ8** | **3** | **System alerts user to go back to console area if planet is exhibited** |

**Gravity Simulator System Requirements (Module 2)**

|  |  |  |
| --- | --- | --- |
| **Number** | **PW** | **Requirement** |
| **REQ9** | **1** | **The system will have an interface containing buttons that will trigger that planets gravity** |
| **REQ10** | **1** | **Another interface has buttons that selects different objects** |
| **REQ11** | **3** | **The system will automatically drop the selected object from the ceiling** |
| **REQ12** | **4** | **The system allows students to move around the environment** |
| **REQ13** | **1** | **The system will be able to simulate each planet’s gravity through the speed at which the selected object is dropped** |
| **REQ14** | **2** | **The system contains objects of varying masses** |

**Grading System System Requirements (Module 3)**

|  |  |  |
| --- | --- | --- |
| **Number** | **PW** | **Requirement** |
| **REQ15** | **1** | **The system will keep track of students progress on quizzes and other activities** |
| **REQ16** | **2** | **The system will provide the instructor with student’s grades on all attempts** |
| **REQ17** | **2** | **The system will compute a grade** |
| **REQ18** | **1** | **The system will keep track of correct and incorrect input** |
| **REQ19** | **1** | **The system will store a grade for each attempt specific to each student** |
| **REQ20** | **1** | **The system will rank the students’ grades from highest to lowest** |
| **REQ21** | **2** | **The system will allow multiple attempts on quizzes and other activities** |
| **REQ22** | **3** | **The system will create a leaderboard consisting of each student’s highest graded attempt** |

**Puzzle (Drag and Drop) System Requirements (Module 4)**

|  |  |  |
| --- | --- | --- |
| **Number** | **PW** | **Requirement** |
| **REQ23** | **1** | **System will display puzzles once its selected from interface** |
| **REQ24** | **2** | **System will have a reset button** |
| **REQ25** | **1** | **System will allow users to drag and drop puzzle pieces** |
| **REQ26** | **2** | **System will allow users to exit the puzzle** |
| **REQ27** | **1** | **System will keep score of each student.** |

**Quiz Implementation module (Module 5)**

|  |  |  |
| --- | --- | --- |
| **Number** | **PW** | **Requirement** |
| **REQ28** | **1** | **The user will select a button to take quiz** |
| **REQ29** | **2** | **The system will display MCQs** |
| **REQ30** | **3** | **The user will choose the answer** |
| **REQ31** | **2** | **System will calculate the result** |
| **REQ32** | **2** | **System will display the result** |

## **3.2 non-Functional Requirements**

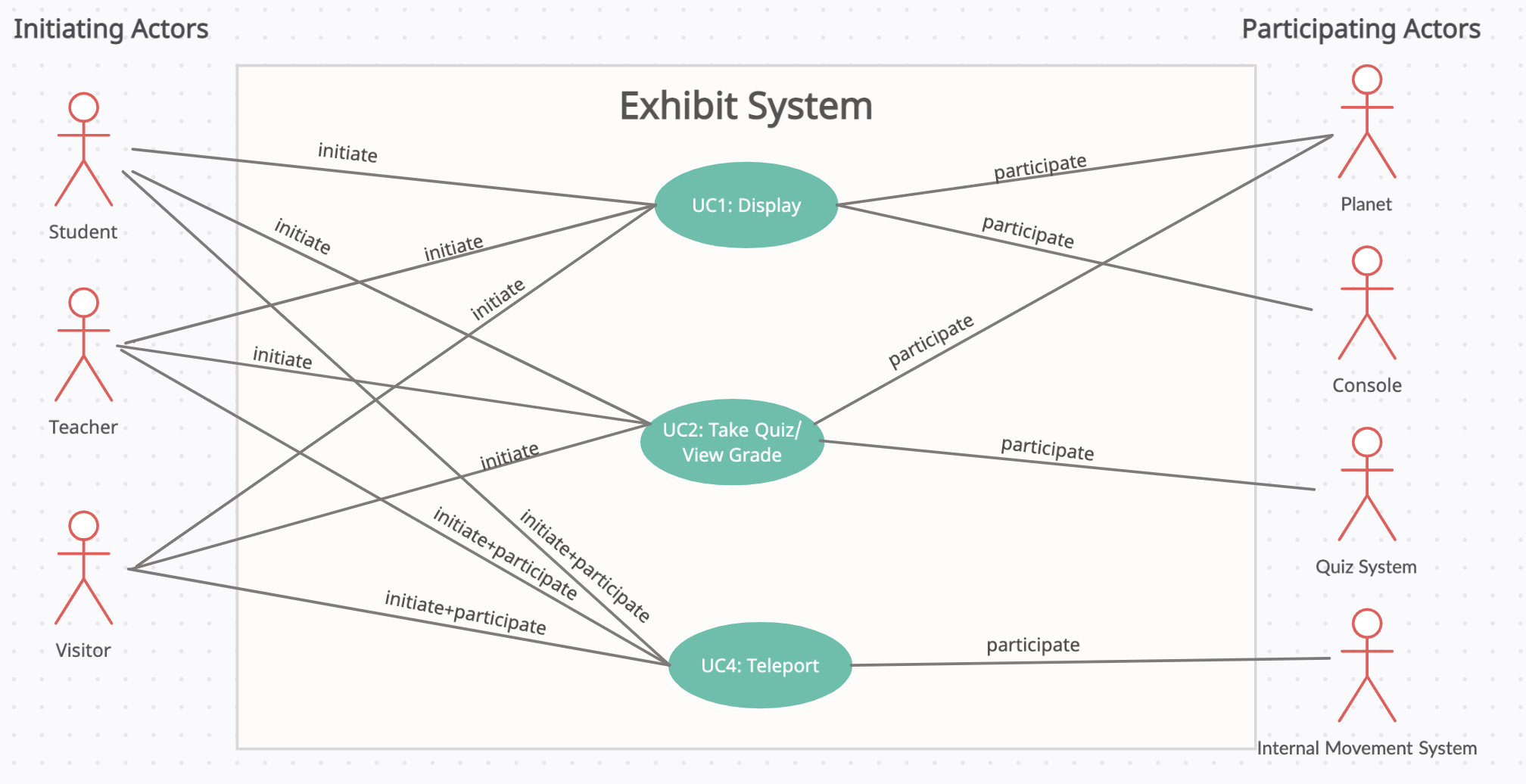
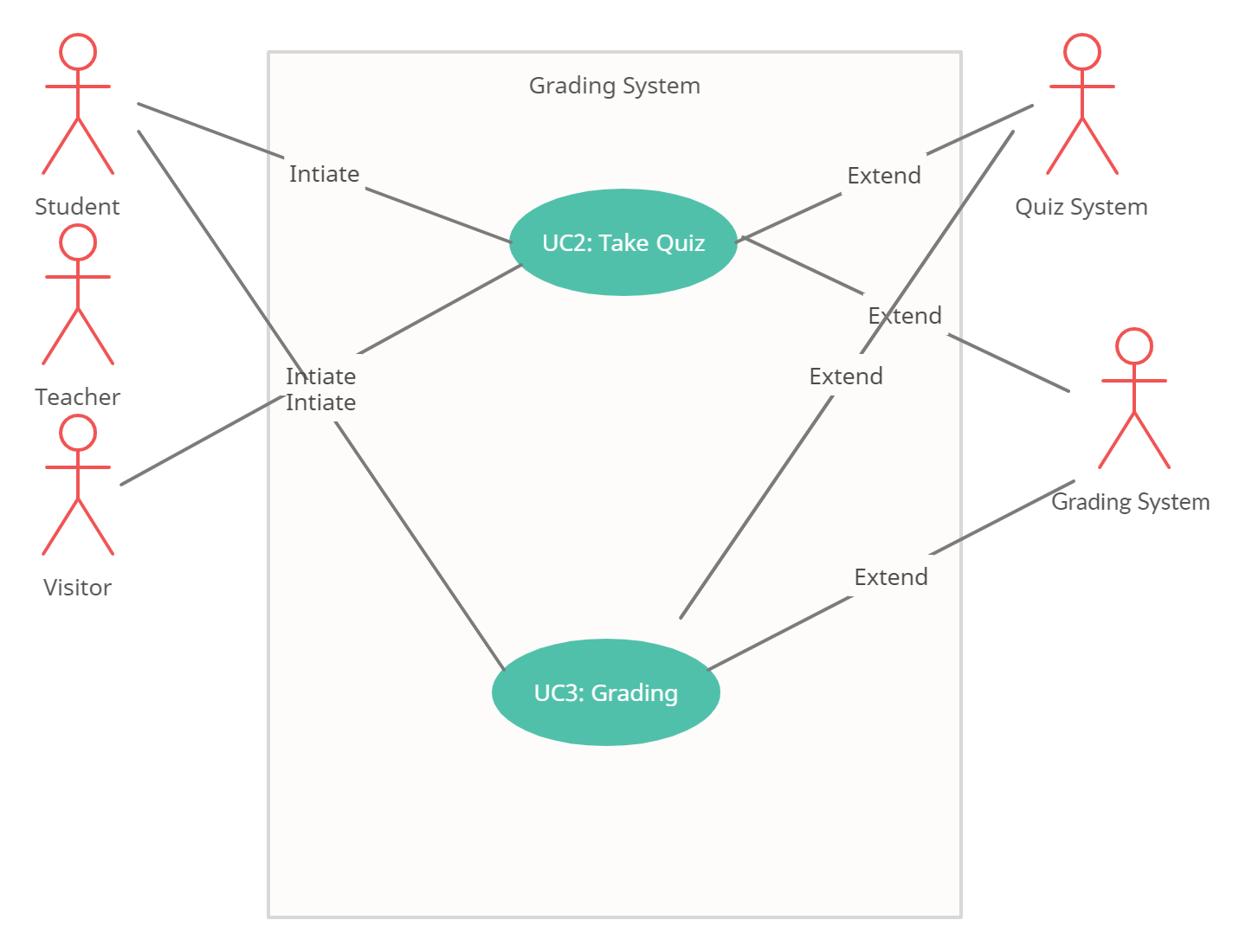
|  |  |  |
| --- | --- | --- |
| **Number** | **PW** | **Requirement** |
| **REQ23** | **2** | **The system will allow the user to move around** |
| **REQ24** | **1** | **The system will provide the user with options of activities** |
| **REQ25** | **2** | **The system will allow the user to enter any room as many times as they wish** |
| **REQ26** | **3** | **The system will provide background music for the main room** |
| **REQ27** | **1** | **The system will allow users to view the solar system** |
| **REQ28** | **1** | **The system will allow users to play a puzzle** |
| **REQ29** | **1** | **The system will allow users to go to any planet** |

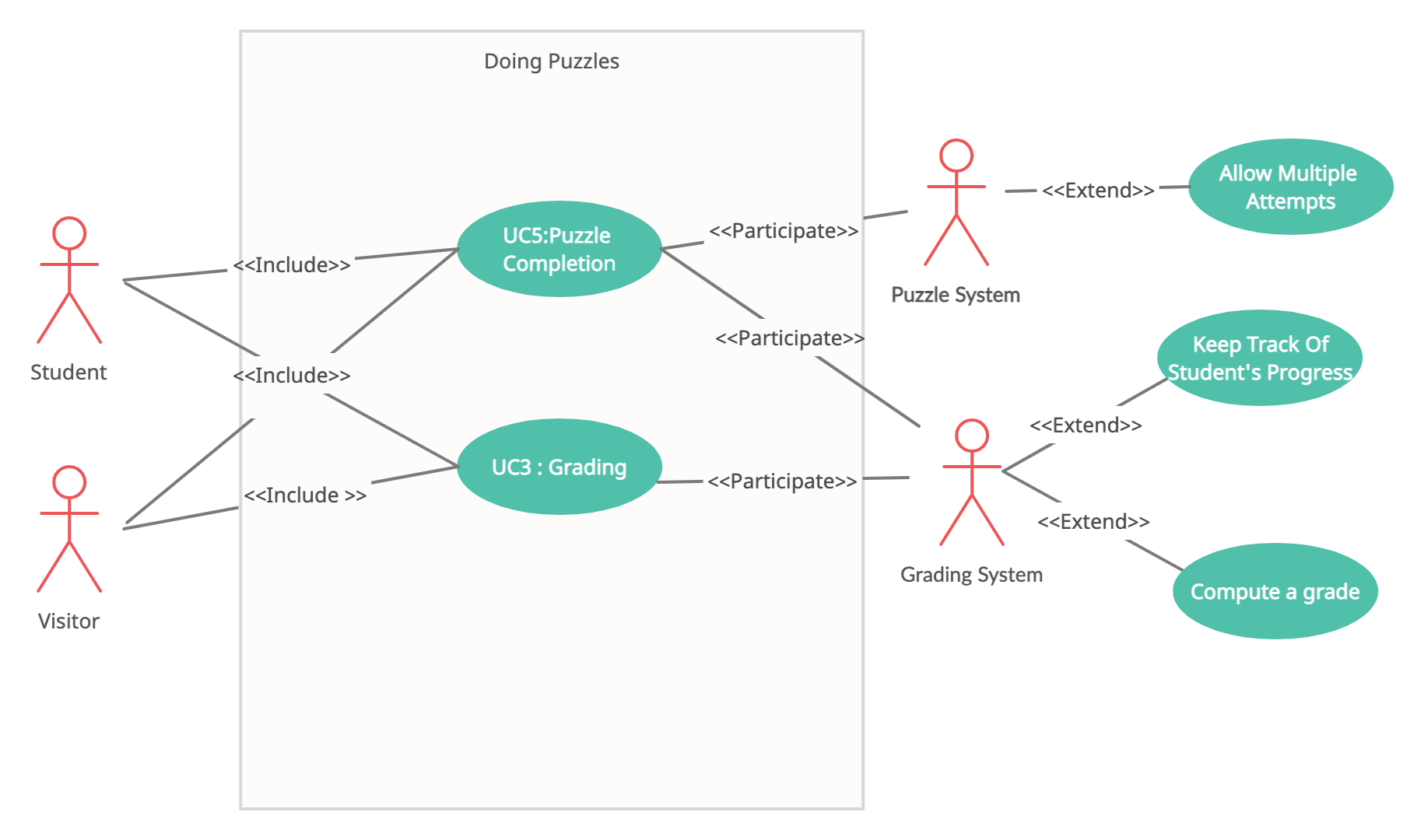
**4. Requirement Model**

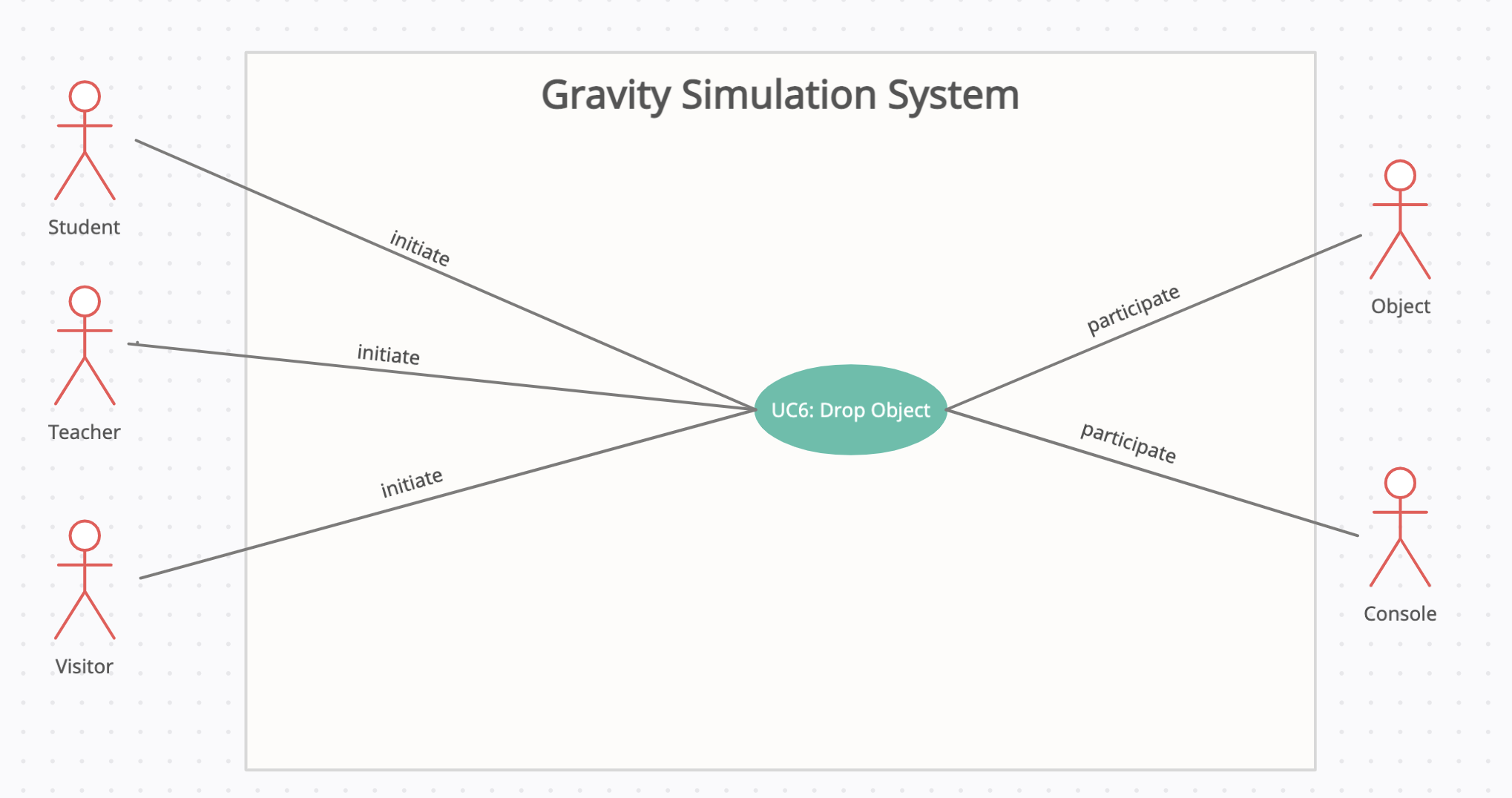
## **4.1 Use Cases**

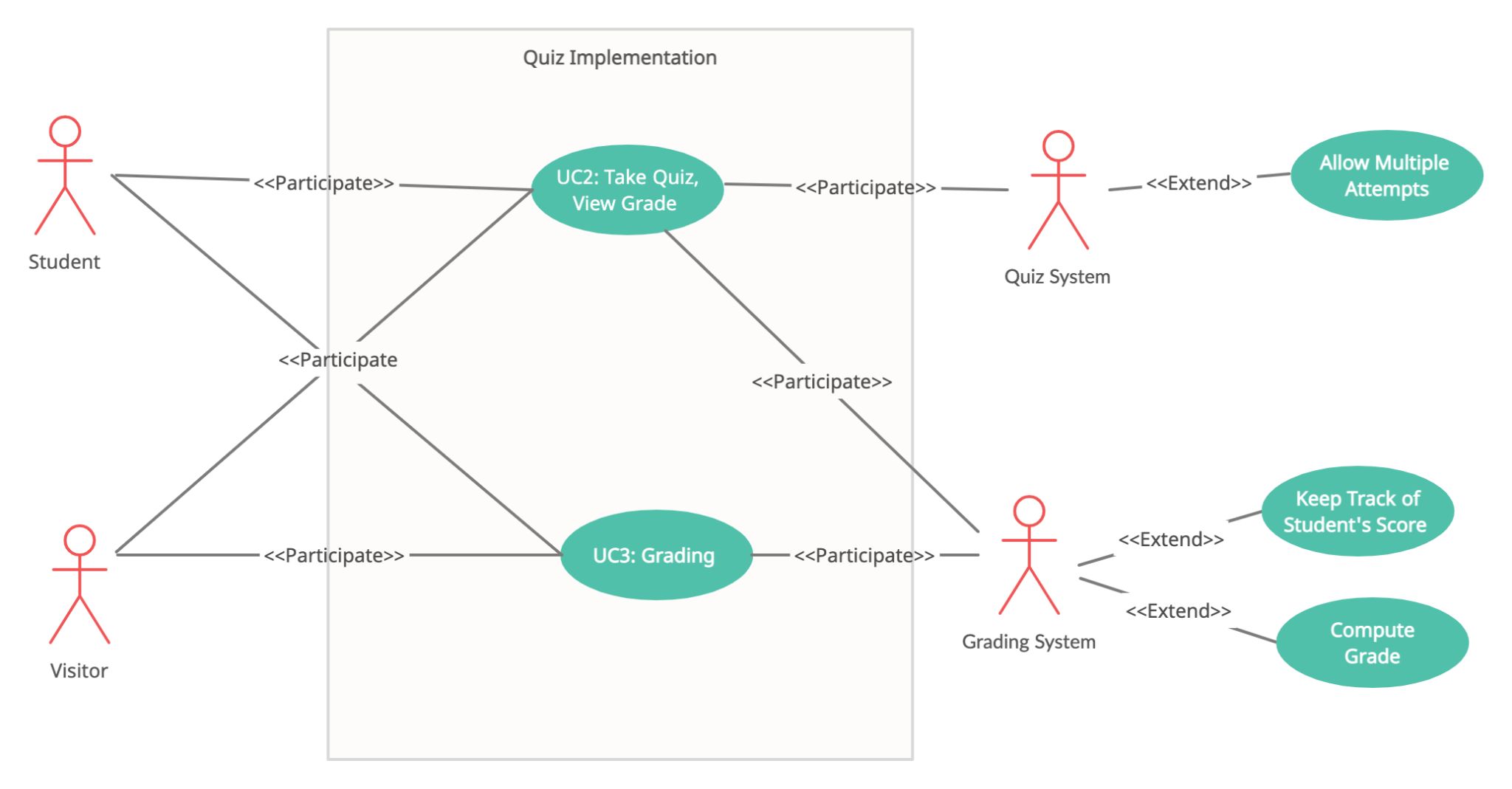
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | **Name** | **Description** | **Requirement** |
| **UC-1** | **Display** | **Displays videos, consoles, planets, quizzes, and reticles** | **REQ1, REQ2, REQ3, and REQ5** |
| **UC-2** | **Taking Quizzes** | **Allows users to take a quiz** | **REQ5, REQ15, REQ22, REQ28, REQ32** |
| **UC-3** | **Grading** | **Calculates total amount of correct and incorrect answers** | **REQ15, REQ16, REQ20, REQ22** |
| **UC-4** | **Teleport** | **Takes user from room to room** | **REQ 1, REQ7, REQ8** |
| **UC-5** | **Doing Puzzles** | **Allows users to play a drag and drop puzzle** | **REQ15, REQ16, REQ17, REQ21, REQ23, REQ26, REQ27** |
| **UC-6** | **Drop Objects** | **Displays accurate physics of an object being dropped from a constant height from different planets** | **REQ9, REQ14** |

## **4.2 Use Case Diagrams**

****

****

****

****

## **4.3 Detailed Use Case Specification of Core Features**

|  |  |
| --- | --- |
| **Use Case# 1 (Module 1)** | **Display** |
| **Related Requirements:** | **REQ1, REQ2, REQ3, and REQ5** |
| **Initiating Actor:** | **Student, teacher and or Visitor** |
| **Actor’s Goal:** | **To display the selected planet as well as information about that planet** |
| **Participating Actors:** | **Planet, Console** |
| **Preconditions:** | **Button will be displayed to select planet** |
| **Postconditions:** | **User will be able to see the facts about the planets** |

|  |
| --- |
| **Flow of Events for Main Success Scenario:**   1. **Teacher/Student**  arrives at the control panel/ interface. 2. User selects planet to be displayed 3. System displays the selected planet along with relevant information. 4. User has the option to zoom in and rotate selected planet 5. User has the option to select a different planet 6. User has the option to go to a different room 7. User has the option to move around the room |
| **Flow of Events for Extensions (Alternate Scenarios):** |
| 1. User walks away from console area once a planet is on display 2. Ground changes color to alert user 3. System alerts user to go back to console area 4. Once the user comes back to the console area, ground color changes back to normal. 5. System alert disappears 6. System resumes planet display |

|  |  |
| --- | --- |
| **Use Case# 2 (Module 1, Module 3, Module 5)** | **Taking Quizzes/ View Grade** |
| **Related Requirements:** | **REQ5, REQ15 - REQ22, REQ28 - REQ32** |
| **Initiating Actor:** | **Student, teacher and/or Visitor** |
| **Actor’s Goal:** | **Take a quiz and view the grade of that quiz** |
| **Participating Actors:** | **Quiz system** |
| **Preconditions:** | **Button to select the quiz** |
| **Postconditions:** | **System will display the answers for the quiz taken** |

|  |
| --- |
| **Flow of Events for Main Success Scenario:**   1. User will see a button to take the quiz 2. Once the button is pressed, MCQs will be displayed for that planet. 3. User will select the options 4. User will submit the result 5. System will calculate the result 6. System will display the result. |
| **Flow of Events for Extensions (Alternate Scenarios):** |
|  |

|  |  |
| --- | --- |
| **Use Case# 3 (Module 3)** | **Grading** |
| **Related Requirements:** | **REQ15 - REQ16- REQ20- REQ22** |
| **Initiating Actor:** | **Teacher** |
| **Actor’s Goal:** | **View and review student’s grades.** |
| **Participating Actors:** | **Grading System, Quiz System** |
| **Preconditions:** | **Button to view student’s grades.** |
| **Postconditions:** | **System will display each student’s grade.** |

|  |
| --- |
| **Flow of Events for Main Success Scenario:**   1. The teacher will have an option button to review the student's grades. 2. The grading system will display and organize student’s grades. |
| **Flow of Events for Extensions (Alternate Scenarios):** |
|  |

|  |  |
| --- | --- |
| **Use Case# 4 (Module 1)** | **TELEPORT** |
| **Related Requirements:** | **REQ 1, REQ7, REQ8** |
| **Initiating Actor:** | **Student, teacher and or Visitor** |
| **Actor’s Goal:** | **To move the user, whenever the button is pressed.** |
| **Participating Actors:** | **Student, Teacher, Visitor, Internal Movement System** |
| **Preconditions:** | **Button to select. The room where the user wants to move.** |
| **Postconditions:** | **User will be moved or teleported.** |

|  |
| --- |
| **Flow of Events for Main Success Scenario:**   1. User will see a button to move 2. Once the button is pressed, the user will be moved into a specific room. |
| **Flow of Events for Extensions (Alternate Scenarios):** |
|  |

|  |  |
| --- | --- |
| **Use Case# 5 (Module 4)** | **Doing Puzzles** |
| **Related Requirements:** | **REQ15 - REQ16- REQ17 -REQ21 - REQ23 - REQ26 - REQ27** |
| **Initiating Actor:** | **Student/Visitor** |
| **Actor’s Goal:** | **To complete puzzle** |
| **Participating Actors:** | **Puzzle system** |
| **Preconditions:** | **The puzzle will give instructions on how to complete it.** |
| **Postconditions:** | **The puzzle will be solved by the Student/Visitor. System will keep track of student’s puzzle completions.** |

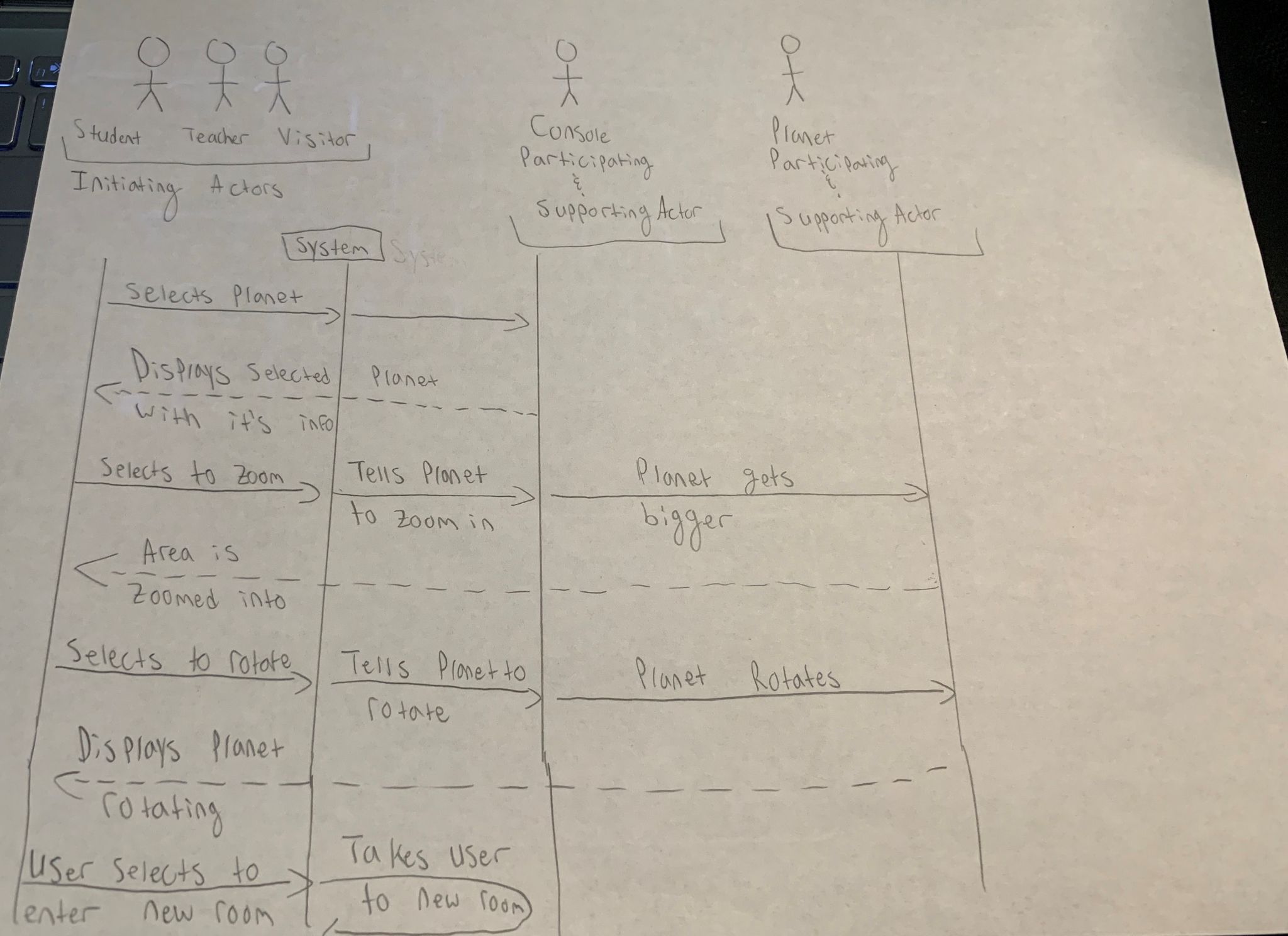
|  |
| --- |
| **Flow of Events for Main Success Scenario:**   1. The student/visitor will be given the option to go the puzzle room 2. The system will provide the student/visitor with instruction on how to complete the puzzle. 3. The student will complete the puzzle. 4. System will record the student’s completion of said puzzle 5. The Student will be given the choice to return to the main room after completing the puzzle. |
| **Flow of Events for Extensions (Alternate Scenarios):** |
|  |

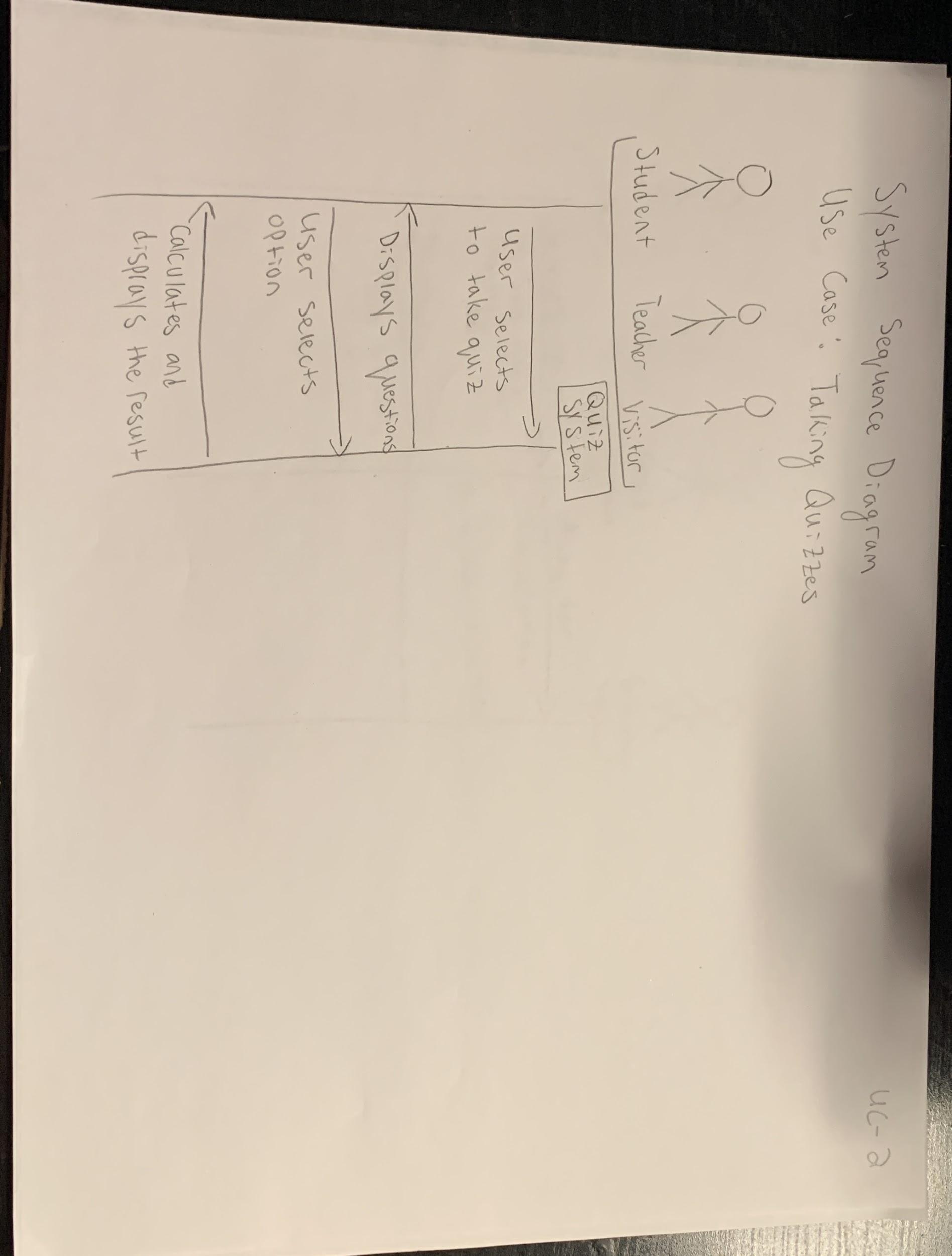
|  |
| --- |
| **Flow of Events for Main Success Scenario:**   1. User will see a button to move 2. Once the button is pressed, the user will be moved into a specific room. |
| **Flow of Events for Extensions (Alternate Scenarios):** |
|  |

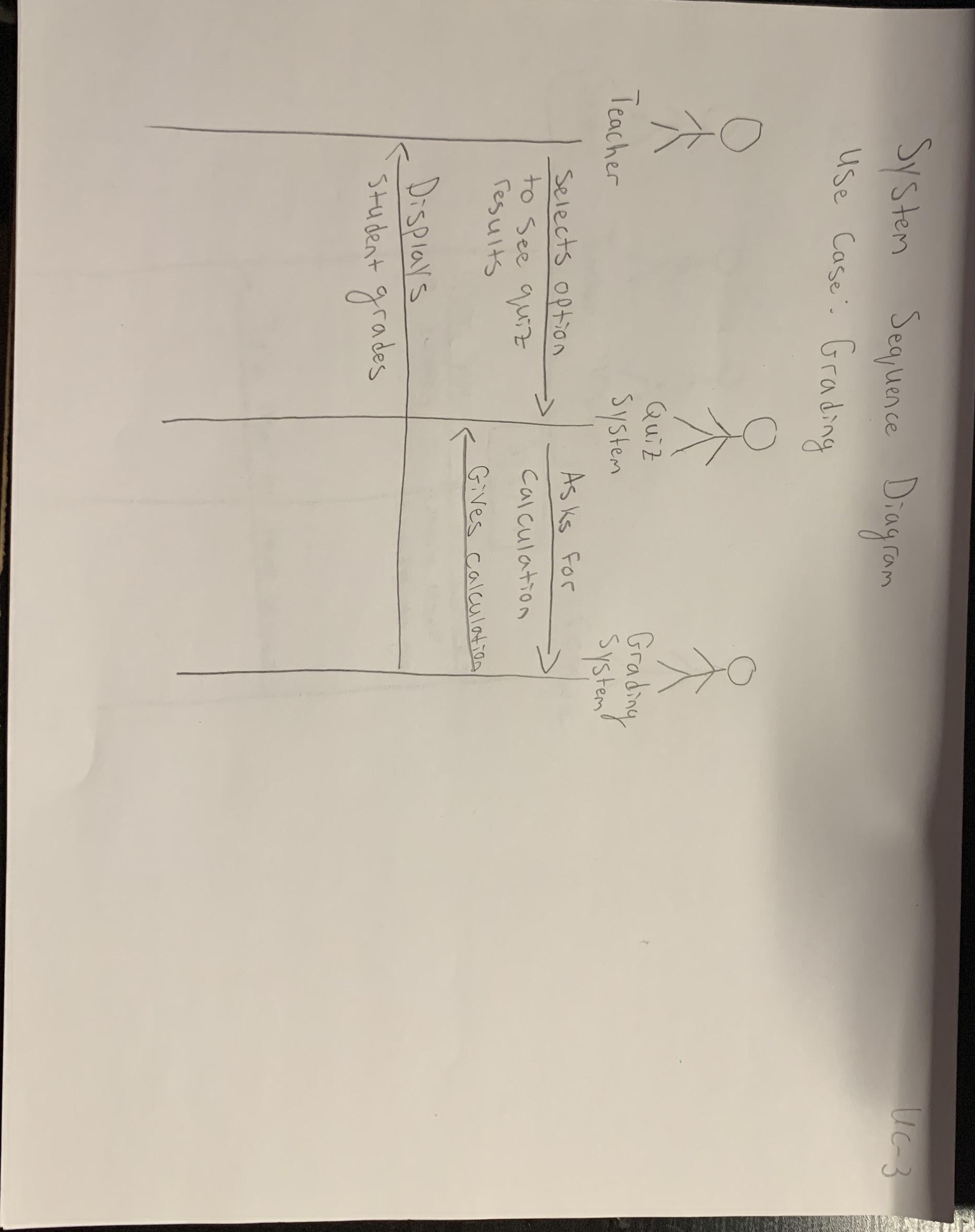
|  |  |
| --- | --- |
| **Use Case# 6 (Module 2)** | **Drop Objects** |
| **Related Requirements:** | **REQ9 - REQ14** |
| **Initiating Actor:** | **Student/Visitor/Teacher** |
| **Actor’s Goal:** | **To simulate each planet’s gravitational speed by dropping an object** |
| **Participating Actors:** | **Object, Console** |
| **Preconditions:** | **The user will be given the option to select a specific object to drop in a particular planet’s exhibit** |
| **Postconditions:** | **The object is dropped at a speed relative to the planet’s gravitational speed** |

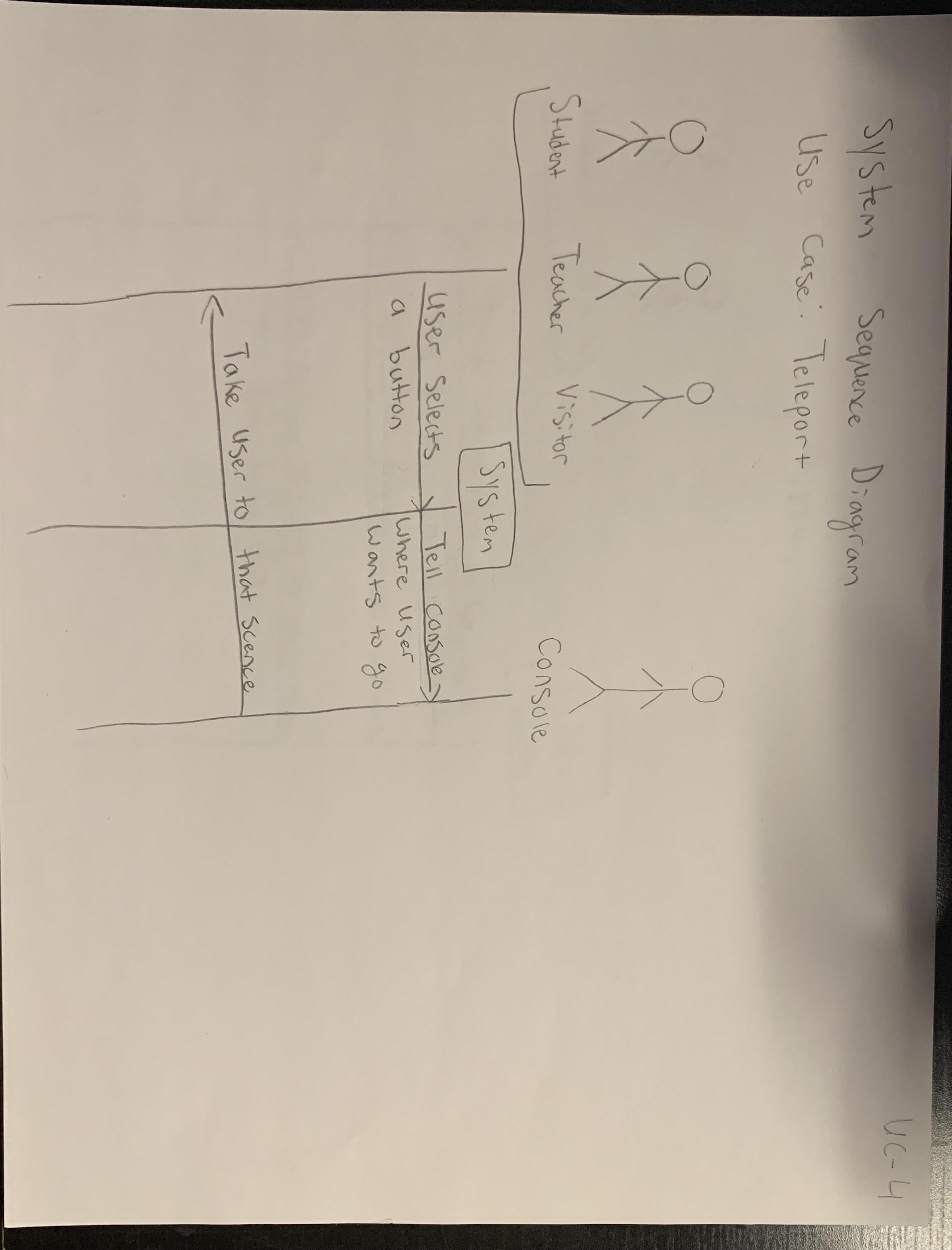
|  |
| --- |
| **Flow of Events for Main Success Scenario:**   1. The user selects a planet exhibit to view 2. Within the selected exhibit, the user selects the gravity simulator activity 3. The user selects an object to drop 4. The object is dropped from the ceiling of the environment 5. The object falls at a speed relative to the planet’s gravity |
| **Flow of Events for Extensions (Alternate Scenarios):** |
|  |

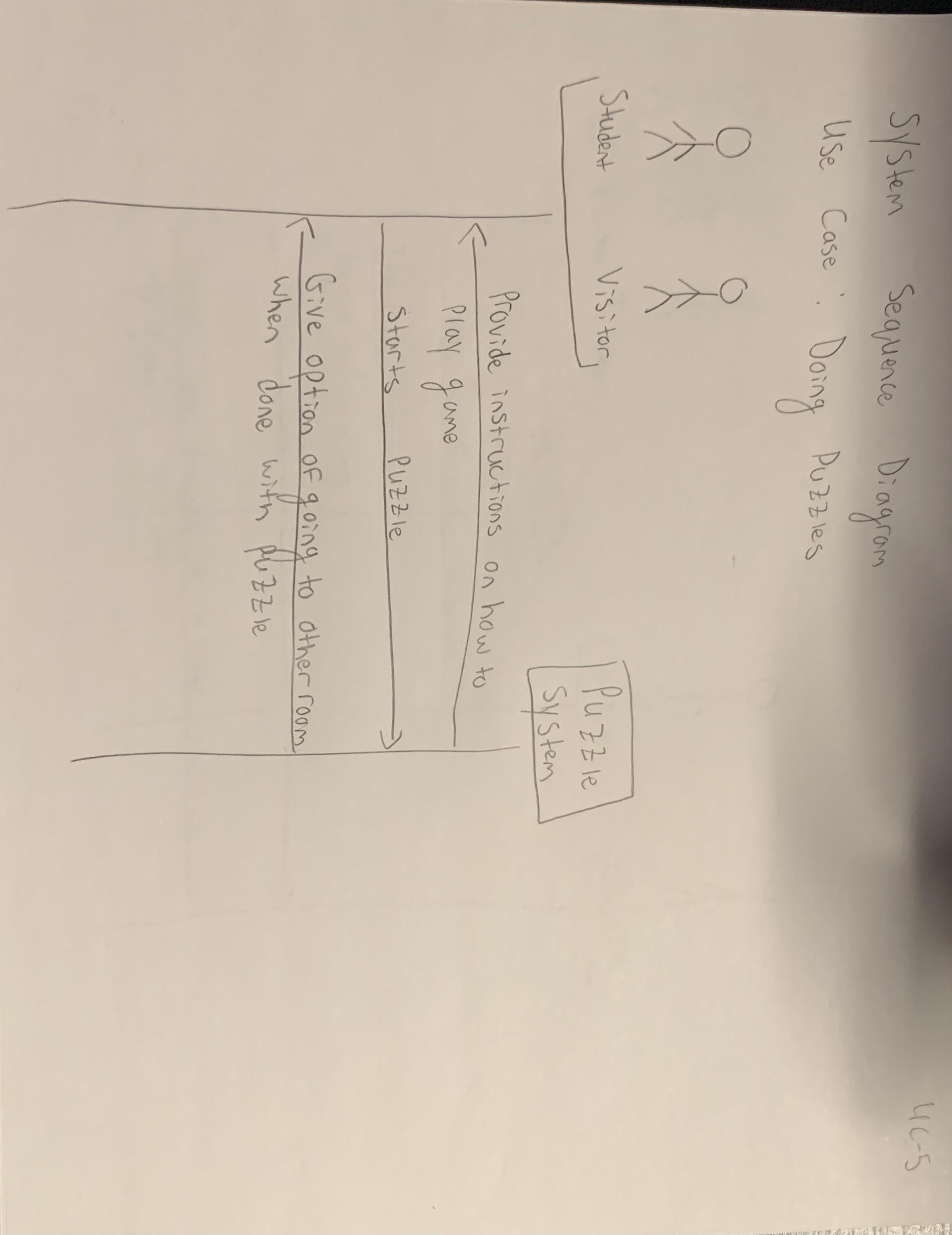
## **4.4 System sequence diagram**

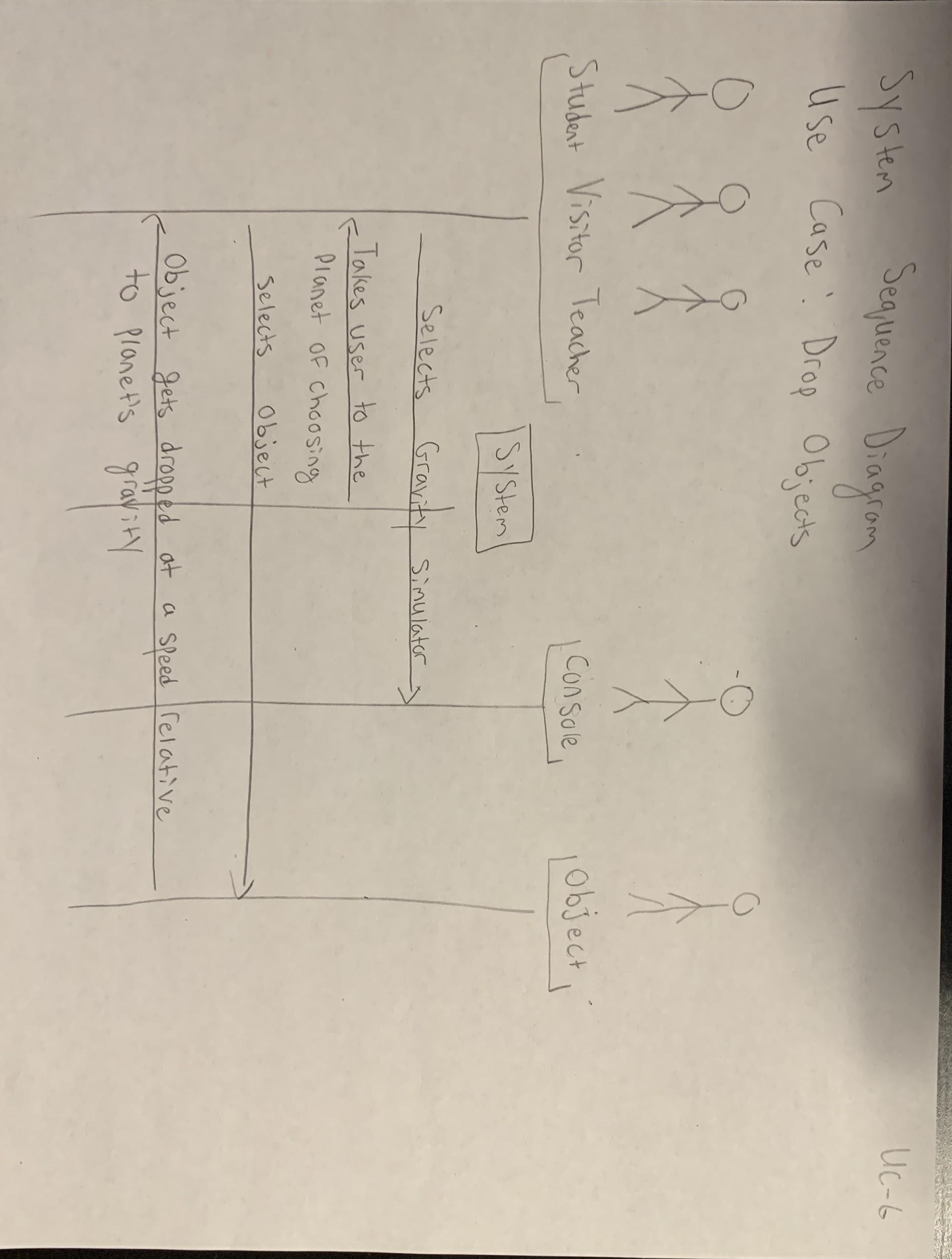
****

****

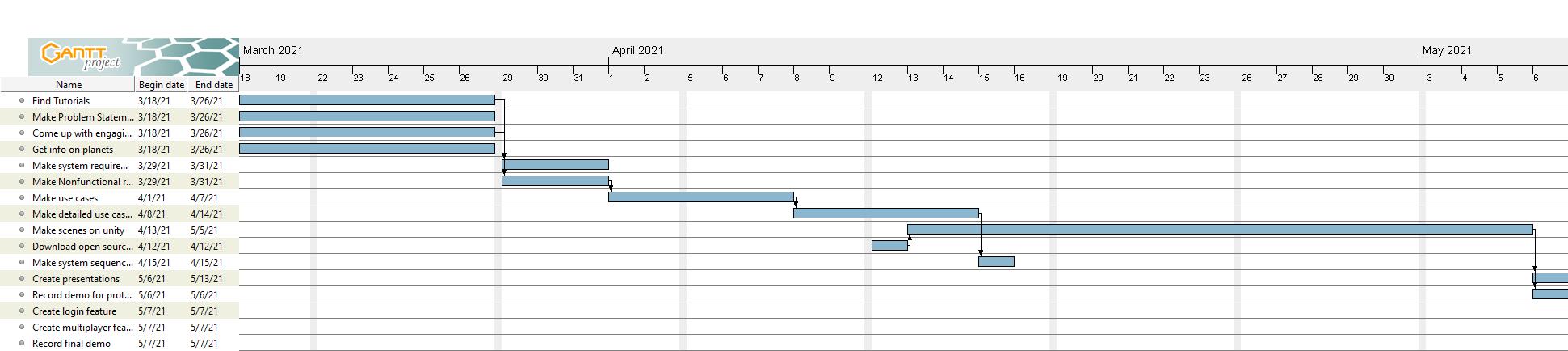
****

****

****

****

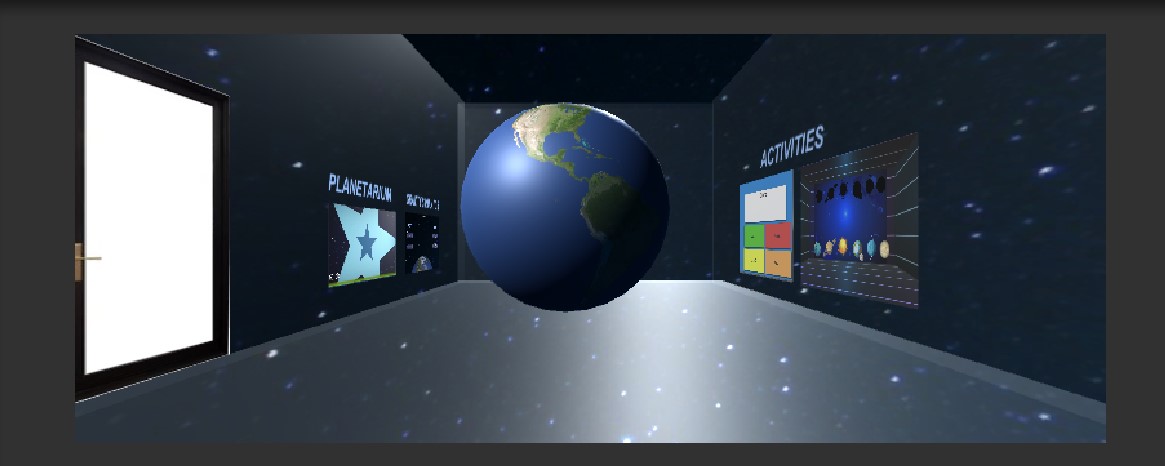
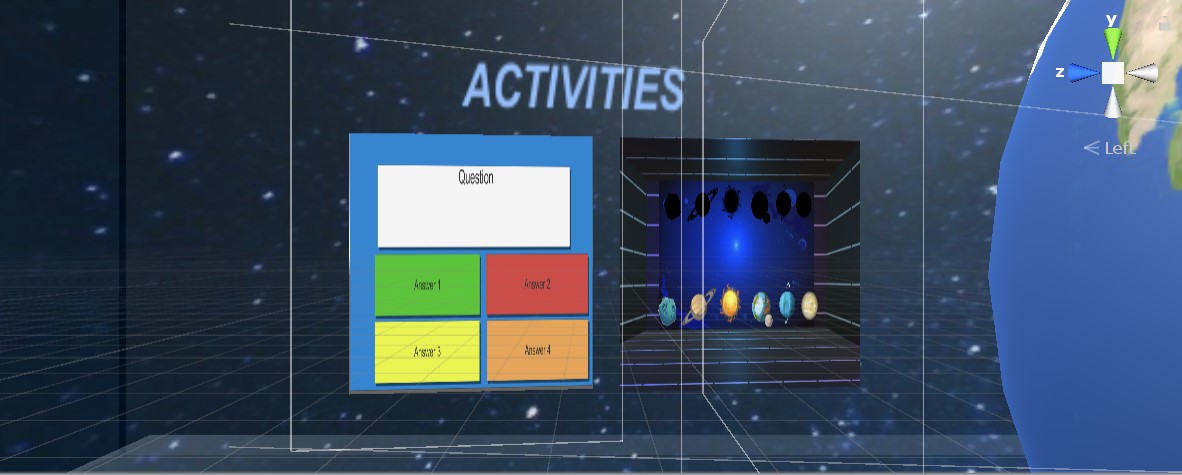
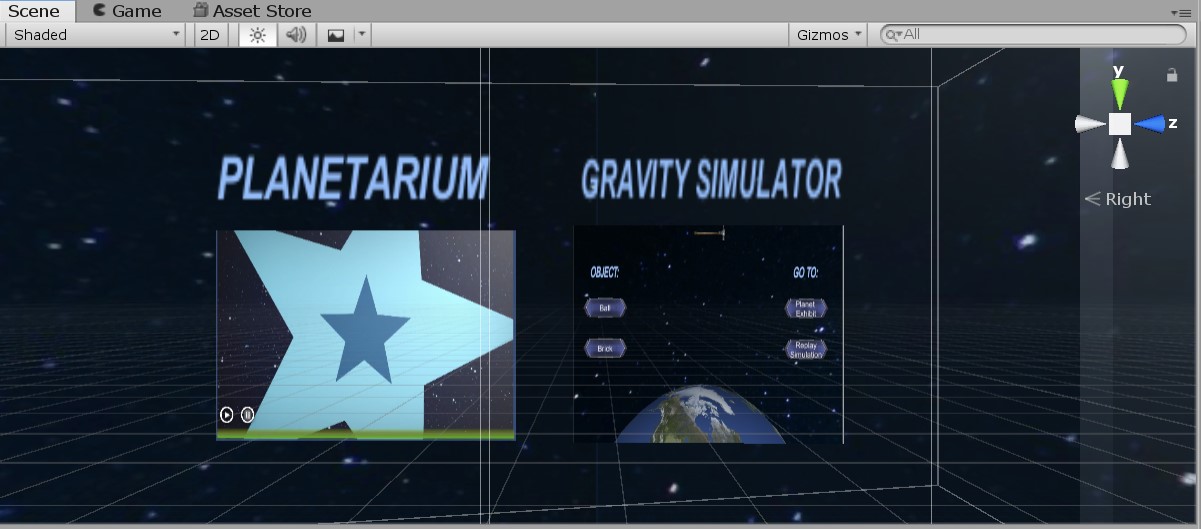
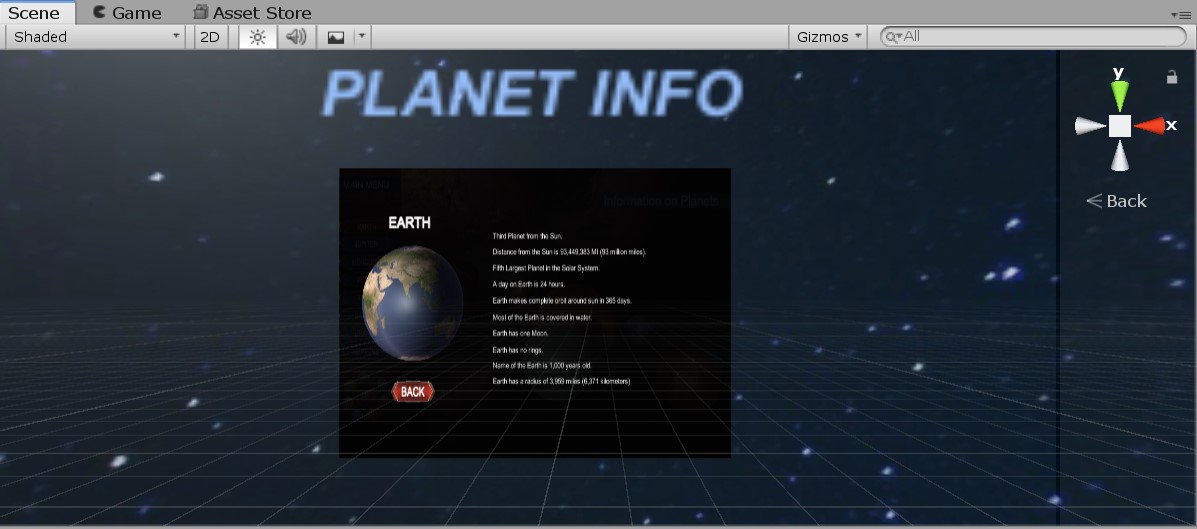
## 4.5 Task assignment and schedule

****

**5. Design of User Interface**

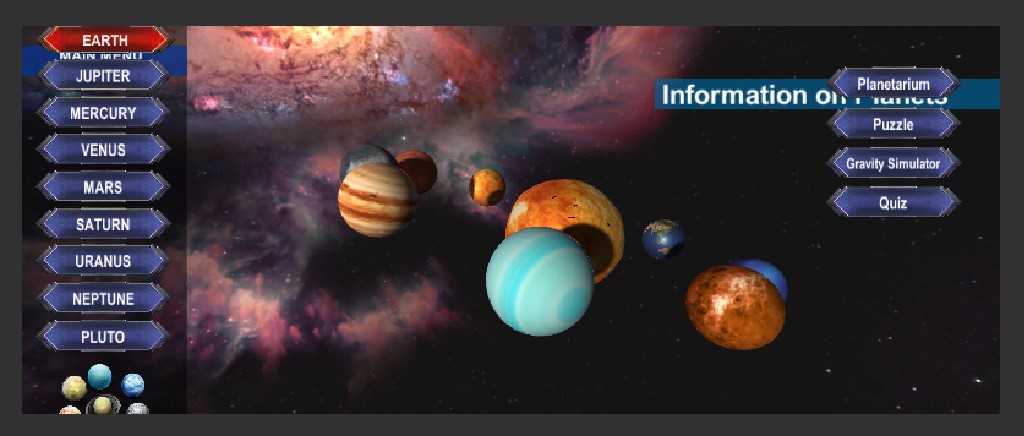
**Console Options:**

We have multiple consoles embedded in the wall of the main room that displays options that a user can select. Our options include Planet Info, Planetarium, Gravity Simulator, and Activities. When a user moves towards any of these options they can use their reticle and click what they would like to do. We use the Google SDK open source code to map head movement and implement a pointer event trigger.

****

**Buttons:**

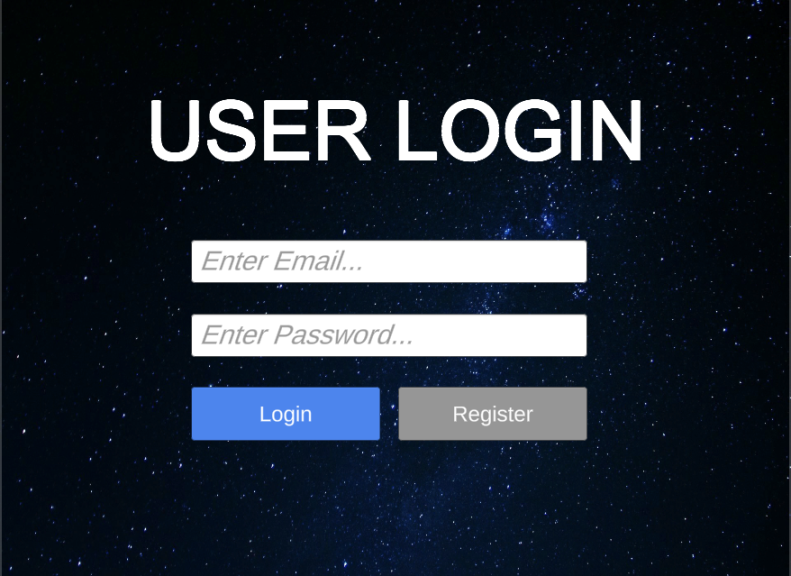
We implemented many buttons throughout Space Learning VR because of our limitations of our platform, Google Cardboard. However, we made sure our buttons functionality was obvious and looked good. In Unity there is an option to create an UI object and from there we made our buttons, and to make sure these buttons did something we incorporated OnClick functions to each. Examples of our buttons being implemented can be seen in our Gravity Simulator and Planets Room.

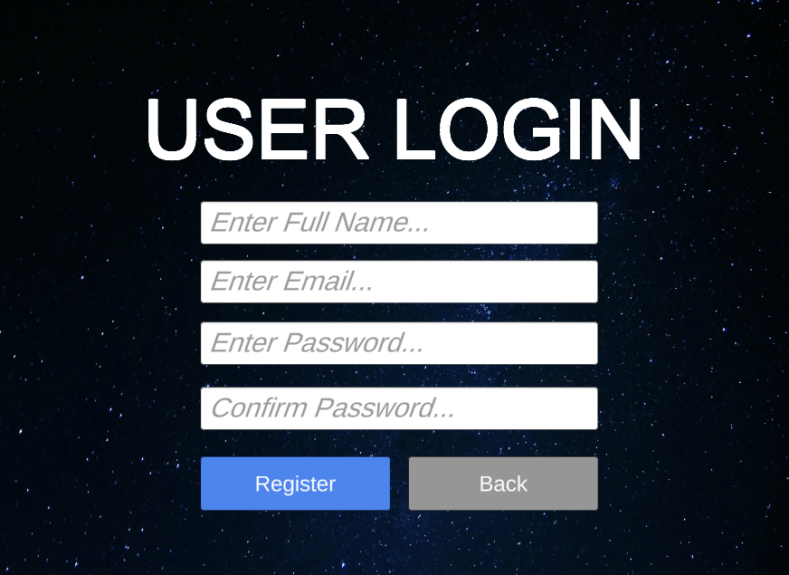
****

**Login:**

We incorporated multiplayer in Space Learning VR so we need user management, in order to do this we implemented a login feature using Firebase. The user first types in their email and password, then gets prompted to type their name and finally they have access to SLVR. If they have not registered they will be prompted to do so.

****

****

****

**Planet:**

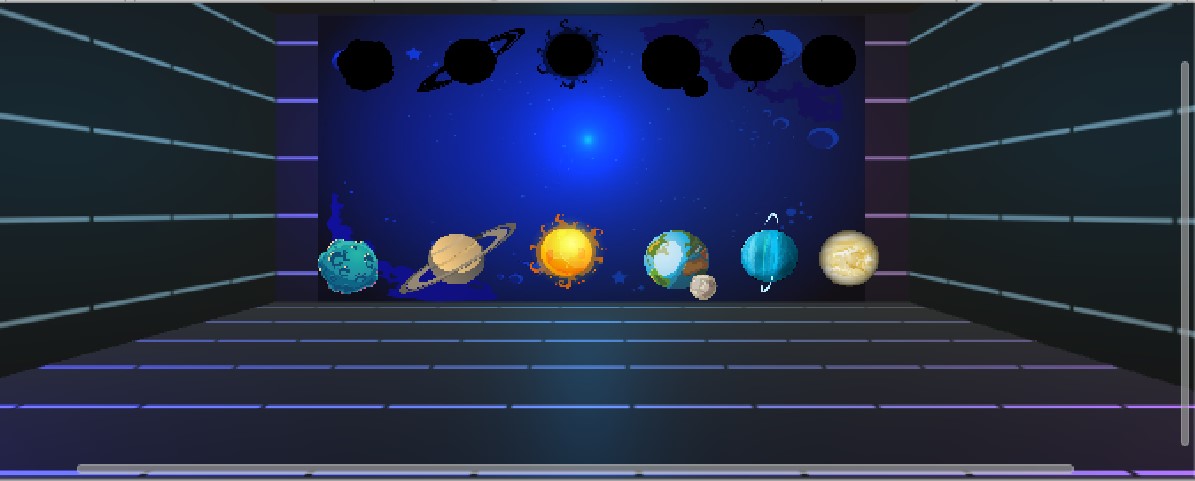
The UI behind our Planets is simple, when a user passes their reticle over the planet it rotates, when it does not the planet stops rotating. This is possible with the use of a pointer event trigger.

****

**(Note: Each planet is in the same room, to see movement refer to our final demo)**

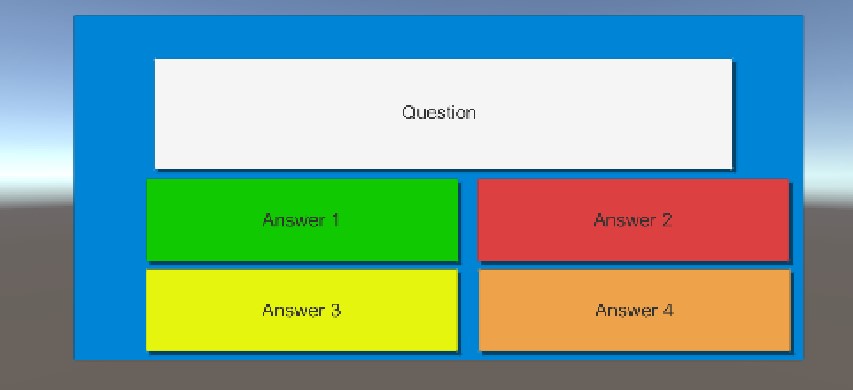
**Puzzle Pieces:**

We allow users to play a game where the objective of the game is to match the puzzle piece to the proper place. This puzzle is a drag and drop puzzle, therefore users use their handy reticle to click on a puzzle piece and hold on until they drop it where they want.

****

**Quiz:**

SLVR is an educational application that is influenced by museums, therefore we want users to have the ability to test their knowledge. We created this by creating an empty game object and assigning a script that manages all the questions and answers. This also incorporates the use of buttons so we implemented the OnClick function as well.

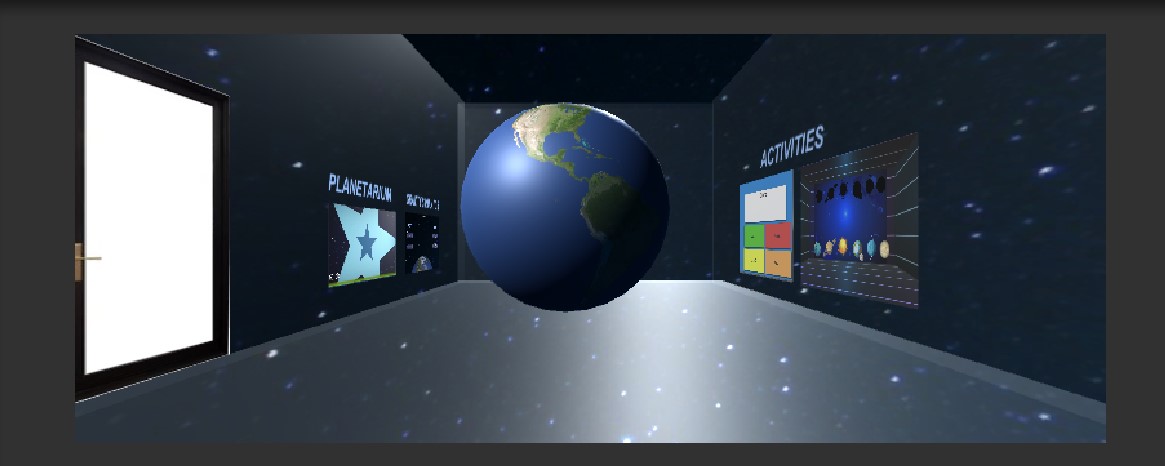
****

**6. Implementation**

## 6.1 Core modules

**The Exhibit (Module 1)**

We have 5 core modules and our first one is on the exhibits. This module is how we display each planet and consoles. When the user starts off in the main room (Earth Exhibit) they see the planet before them, we are able to display the planet through the implementation of a sphere object that is covered with an Albedo map which gives it the appearance of that planet. When the user passes their reticle over the planet it rotates, this is due to the event trigger of a pointer enter and exit with a script named Rotate. Rotate is essentially a script that is applied to all planets that makes them rotate when need be. The user has the ability to move around the room which is possible through the implementation of the reticle pointer. When the reticle pointer hits the ground at an angle, the player moves in that direction due to the event trigger of pointer enter and exit as well. Now if a user wants to interact with the consoles they must use their reticle to select an option, this uses the same implementation as the other reticles.



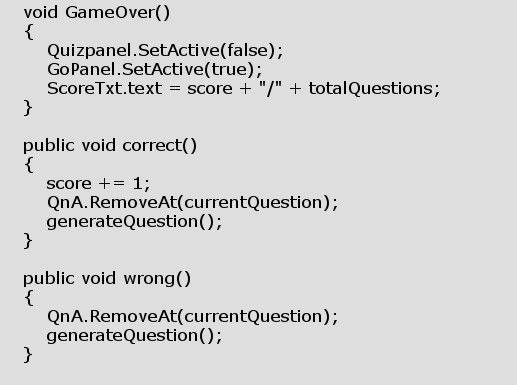
**Gravity Simulator (Module 2)**

When the user is in the gravity simulator room they have 4 buttons to select from: Object 1, Object 2, Go To option 1, and 2. We implemented these buttons by creating them in Unity and adding the OnClick function to each one. These buttons are also linked to their corresponding action, for example, the Go To Planet Exhibit button is linked to the Planet Exhibit. As soon as they get here the object gets automatically dropped from a constant height. For this we implemented a rigid body and a script called attractor. The rigid body is the physics of the object and the attractor script interacts with the rigid body by replicating the gravity of that planet.



**Grading System (Module 3)**

The Grading System is implemented for the quiz. For the quiz we allow users to test their knowledge and to check their progress on how much they learned. We implemented this by the console giving feedback, when the console gives an indication of a correct answer we keep a running total and display the amount of correct out of total questions.

****

## **6.2 User management**

We first had to make the user interface for our user management, so in Unity we made a Panel and created some buttons that accept input as described in **Section 5.** Through the use of Firebase, we had to register and set up an authenticator, then we downloaded the Firebase SDK and imported it into Unity. Following this we implemented code that was provided by a tutorial with minor edits to the code that took care of the majority of the functionality, we made scripts that managed the authenticator and the UI. In Unity we used the authenticator, a game object, and made use of the OnClick function to create a successful login.

## 

## **6.3 Multiple user connection**

Space Learning VR replicates some elements of being in an in-person museum, and one of those elements is sharing the space with others. In order to do so we made sure to incorporate multiple user connections. We implemented this by heading into Unity and creating a screen that allows users to type in their name and we as well created a button to play. We made a script called Launcher that loads you into the main room, more specifically it counts the amount of people that want to enter, loads the room, then sends all of them there. This functionality works in tandem with the login function.

6.4 Prototype

Link to Prototype: <https://youtu.be/EaGUT8E1P1o>

## 6.5 Code repository

Link to Code Repository: <https://github.com/eggSavorSouffle/Space-Learning-VR/tree/main/Scripts>

**7. Design of Tests**

**Gravity Simulator**

|  |  |
| --- | --- |
| **Action** | **Output** |
| Click Object button | **SUCCESS:** Object gets dropped  **FAILURE:** Object does not get dropped |
| Click Go To button | **SUCCESS:** User gets taken to a planet exhibit  **FAILURE:** System does not register clicked button |
| Click Replay button | **SUCCESS:** Object gets dropped  **FAILURE:** Object does not get dropped again |

**Puzzle**

|  |  |
| --- | --- |
| **Action** | **Output** |
| Drag puzzle piece | **SUCCESS:** Puzzle piece locks in correct spot  **FAILURE:** Puzzle piece gets sent back to original spot |

**Login**

|  |  |
| --- | --- |
| **Action** | **Output** |
| Enter email | **SUCCESS:** Gets prompted to enter password  **FAILURE:** Application sends error message and crashes |
| Enter password | **SUCCESS:** Gets prompted to enter name  **FAILURE:** Application sends error message and crashes |
| Register an account | **SUCCESS:** Gets prompted to enter email, password, and name  **FAILURE:** Application sends error message and crashes |
| Enter name | **SUCCESS:** Gets sent to main room  **FAILURE:** Application sends error message and crashes |

**Console**

|  |  |
| --- | --- |
| **Action** | **Output** |
| Select an option | **SUCCESS:** Gets sent to the room  **FAILURE:** Reticle does not work and nothing happens |

**Appendices**

## **Summary of Individual Work**

## **Team Cooperation Chat Record (discord link or other social media)**

<https://discord.com/channels/824065065375498240/824065908934180896>

### **Weekly progress updates and work plans**

**4/18 --- 4/24** – Worked on the system requirements and use case (Flow of Events and Alternate Scenario). Started working on video player part of project.

**4/25 --- 5/1 –** Helped to create use case #2, #3 and #5. Also created use case diagrams for use cases #4 & #5. Finished working on the video player part of project.

**5/2 --- 5/8 –** Started to work on creating an interactive puzzle for project.

**5/9 --- 5/15 –** Finished working on puzzle, worked on implementing multiplayer aspect for our project by following the provided tutorial.

**5/16 – 5/22 –** Worked on providing an update for our project, worked on creating our final presentation slides, recorded video presentation with teammates.

**Lessons and Experiences that I learned**

I learned how to work in a team with the unity application, more specifically, I learned how to synchronize with a team, delegate work for each one of us, and combine our individual work onto our project.

We used discord to talk to each other, along with unity’s built in collaborate feature to share our work, which made it easier for us to collaborate.

Along with getting familiar with unity, I also learned how to work with a new programing language, C#. This new programing language helped me code some of the features that we needed for our project.

### **Unity Tutorials used in your project**

How to create a video player with controls

<https://www.youtube.com/watch?v=9LwOmMzOrp4>

Drag and Drop to make puzzle

<https://www.youtube.com/watch?v=yjmWL02W2-A>

ProBuilder to make rooms

<https://www.youtube.com/watch?v=ncznTdzVnV0&t=344s>

Jeremy’s tutorial to implement multiplayer

<https://doc.photonengine.com/en-us/pun/current/demos-and-tutorials/pun-basics-tutorial/intro>