Education Gamification

Team Name: Education Gamification Team

Team Members:

Justin Mitchell

Matt Adams

Nathan Luiten

Trevor Stewart

Date: 10/16/2024

Eastern Washington University

Table of Contents

Project Overview

The Education Gamification Team is working with Erik Powell, a teacher from Ferris High School. The desired outcome of this project is to take a portion of Eriks' curriculum and create a more effective, enjoyable, learning experience for the English students at Ferris High School. This will be achieved by gamifying a section of Eriks' curriculum. Creating a game out of one of the curricular topics will make the material more engaging for students and boost their comprehension of the topic. The primary users are English students at Ferris High School and their teachers. Ideally, students will play the game to aid in their learning of the given curricula on their school-provided, Windows-based, laptops. Teachers will be able to monitor the progress of their class, and each individual student in order to verify they are grasping the material that is provided to them. Erik provided us with a helpful question that we hope to answer with this project. "*How do you prevent the slog of a high school English class?"*. The Education Gamification Team has decided to create an isometric RPG based on Macbeth. The game will take the player through the trials and tribulations of Macbeth's journey to insanity. This game will act as a learning aid as a student reads through the Shakespearean text and will provide important contextualization through visuals and common English translations. Erik expressed a potential interest in multiplayer functionality which does not fit this style of game, but if project time allows, we will implement multiplayer educational mini games that a class can all participate in.

User Stories and Scenarios

User Story 1: Saving Progress

U1: As a student, I want my progress to be saved so that when I return to the game, I can continue where I left off without losing my previous progress.

Scenario 1: Resuming the game from where the user left off

- Given I have completed part of the game and logged out of the game
- When I log back into the game
- Then I should see an option to continue the game from where I left off
- And when I choose to continue, I should be taken to the same part of the game I was last playing

Scenario 2: Progress saved after completing a level

- Given I have completed a level in the game
- When I log back into my account after logging out
- Then I should see the completed level marked as finished in the level selection screen
- And the next level should be unlocked for me to start

User Story 2: Unlocking New Levels

U2: As a student, I want to unlock new levels after completing the previous ones so that I can progress through the game and continue learning.

Scenario 1: Unlocking a new level after completing the current level

- Given I have completed a level in the game
- When I return to the level selection screen
- Then I should see the next level unlocked
- And I should be able to start playing the new level

Scenario 2: No level unlocked without completing the current level

- Given I have not completed the current level
- When I return to the level selection screen
- Then the next level should remain locked
- And I should see a message prompting me to complete the current level first

User Story 3: Stats page

U3: As a teacher I want to be able to see the progress of my students.

Scenario 1: Teacher reviews student progress

- Given I am a teacher
- When my students are done playing the game
- Then I should have some pages that show the stats of my students to get a gauge of my students' understanding

User Story 4: Ease difficulty

U4: As a student, I want to be able to ease my way into the game.

Scenario 1: Selecting difficulty or give hints

- Given I am a student who is struggling with the material,
- When I am playing the game and am stuck,
- Then I should be able to get help from either adjusting the difficulty or receiving a hint.

User Story 5: Easy to access the game

U5: As a teacher, I want my students to be able to access this game on their school Chromebooks with little hassle.

Scenario 1: Avoid needing to install the game

- Given I am a student with restricted privileges on the school computers,
- When it comes to accessing the game we will not be able to install the game,
- So it would be much easier if we could get to it through the browser.

User Story 6: RPG with educational elements to explore a Shakespeare play

U6: As a teacher, I want my students to be able to explore a world that makes one of Shakespeare's plays more digestible and engaging.

Scenario 1: Make the view of the game world isometric

- Given the setting in these plays are quite full and complex,
- When it comes to displaying the player's environment,

- Then the camera/point of view should be isometric as 2D would be too restrictive in terms of displaying the world.

Scenario 2: Players should be able to interact with characters from the play.

- Given these stories are full of different complex characters,
- When the player comes across with on of theses characters,
- Then they should be able to interact with that character to gain a better idea of who they are.

Scenario 3: Illusion of choice

- Given we want these students to be engaged in an already existing story,
- When playing the game we should give the illusion of choice,
- So the player feels more engaged and connected to the story and game.

User Story 7: Understanding Early Modern English

U7: As a student, I want early modern English to be translated to modern English so that I can understand the material better.

Scenario 1:

- Given I am playing the game and encounter early modern English text from a Shakespeare play,
- When I come across a passage or dialogue that is hard to understand,
- Then I should be able to see a modern English translation or an option to view the translation so that I can comprehend the text.

Scenario 2:

- Given I am playing a dialogue-heavy part of the game,
- When I click on certain phrases,
- Then I should be shown both the original text and a brief explanation or modern equivalent to help me understand the language better.

User Story 8: Leaderboards

U8: As a student, I want to compete with other students to make me want to learn more

Scenario 1:

- Given I have finished the last level

- When I look at the overview of the levels
- Then I should be able to see a leaderboard with point totals based on the choices I made in the game

User Story 9: Mini-Games

U9: As a player, I would like to be able to interact with other players while learning about Macbeth

Scenario 1:

- Given I have a class of students who need to learn about specific elements of Macbeth
- When I click on the mini-games button
- Then I should be able to create a room for students to join for a specific mini-game

Scenario 2:

- Given I am a student in class
- When I click on the mini-games button
- Then I should be able to join a room that my teacher has set for a specific mini-game

User Story 10: Login

U10: As a student, I would like to play the game at home from where I left off at school.

Scenario 1:

- Given I have saved my game at school,
- When get home and try to play the game some more,
- Then I should be able to login into my account to resume the saved file I had created at school.

User Story 11: Audio Settings

U11: As a student, I would like to play the game in the computer lab which is much quieter, so it would be nice to adjust the music and sound effects of the game accordingly.

Scenario 1:

- Given I am now playing the game somewhere else,
- When the sound is too quiet or loud based on my surroundings,
- Then I should be able to access an options menu to adjust them.

Functional Requirements

Progression

Functional Requirement	Description/Specification	Priority
FR-1(U1)] Save progress	At any point in the game, the student should be able to open a menu and save their progress. They then should be able to re access that point in time at a later date by logging in again.	Level 0 (Essential)
FR-2(U2)] Unlocking levels/Opening new parts of the game	As the player explores the game, when they complete a specific task, new things should be opened up in the game so further progress can be made.	Level 0 (Essential)

Teacher Monitoring

Functional Requirement	Description/Specification	Priority
[FR-3 (U3)] Teacher Dashboard for student progress	The system must save the student's progress after each level is completed and allow the student to resume from where they left off when logging back in.	Level 3 (Low Priority)

Game Accessibility/Login

Functional Requirement	Description/Specification	Priority
[FR-4 (U5)] Browser-Based Access	The game must be accessible through a web browser without requiring installation, to accommodate restricted privileges on school-provided computers.	Level 0 (Essential)
[FR-5 (U1,10)] Log in to resume save file	When the student has made progress in the game on one device, they should then be able to resume their progress on a different device by logging in with their student email.	Level 0 (Essential)

Difficulty Adjustment

Functional Requirement	Description/Specification	Priority
[FR-6 (U4)] Difficulty Adjustment and Hints	The system must provide an option for students to adjust the game difficulty or access hints when they are struggling to understand the material.	Level 2 (Medium Priority)

RPG Gameplay

Functional Requirement	Description/Specification	Priority
[FR-7 (U6)] Isometric Game Environment	The game must feature an isometric view of the world to allow students to explore the setting of Macbeth.	Level 0 (Essential)
[FR-8 (U6)] Character Interaction	The game must allow students to interact with key characters from the play to gain a better understanding of the narrative and their significance.	Level 0 (Essential)
[FR-9 (U6)] Illusion of Choice	The game must give the illusion of making meaningful choices to increase engagement in the story, even if the outcome is predetermined.	Level 1 (High Priority)

Education

Functional Requirement	Description/Specification	Priority
[FR-10 (U7)] Translating the story into modern English	To help make the story of Macbeth more digestible, we need to translate the characters speech and narration to use more modern wording.	Level 0 (Essential)

Multiplayer

Functional Requirement	Description/Specification	Priority
[FR-11 (U9)] Multiplayer minigames	To add more options to the game, we could add a multiplayer mode where students could create lobbies with their friends to play minigames with.	Level 1 (High Priority)

Competitive Leaderboard

Functional Requirement	Description/Specification	Priority
[FR-12 (U8)] Leaderboard	The game must include a leaderboard where students can view their point totals and compare their scores with other students based on actions taken throughout the game.	Level 2 (Medium Priority)

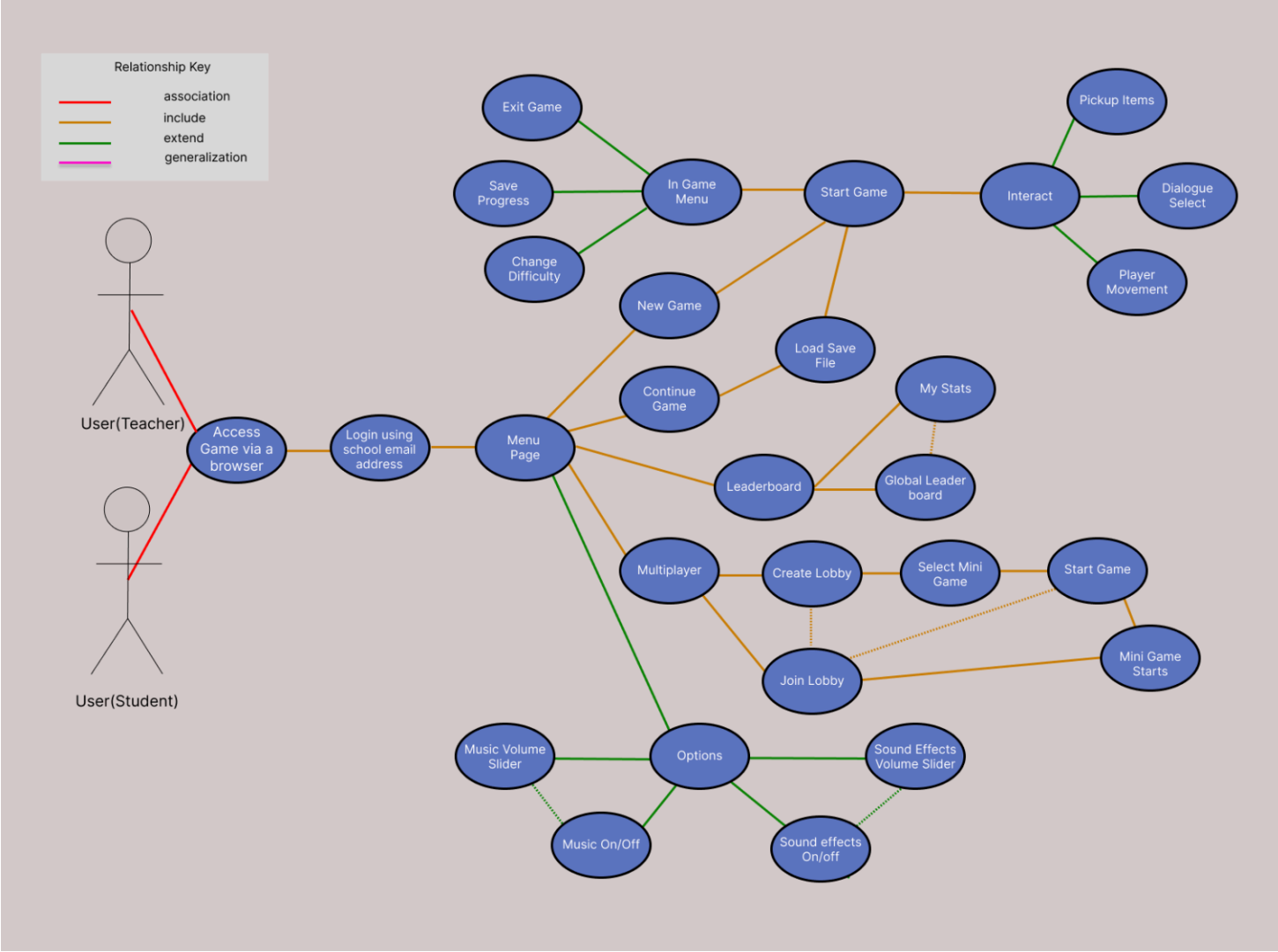
Audio Settings

Functional Requirement	Description/Specification	Priority
[FR-12 (U8)] Audio Settings	The game must have the option so adjust the music and sound effects volume levels.	Level 2 (Medium Priority)

Non-functional Requirements

Non-functional requirement	Description
[NFR-1] Security	Only students from Ferris will be able to play
[NFR-2] Performance	The game must load within 5 seconds on the school-provided laptops under normal conditions.
[NFR-3] Usability	The game's UI should be easy to navigate for students and teachers.
[NFR-4] Security	One student should not be able to affect someone else's save without logging out of their account
[NFR-5] Scalability	The mini-games should support up to 30 players

Use Case Diagram



Use Cases

Use Case 1: Log in

- **Actors:** Student
- **Triggers:** Navigating to the game on the web
- **Preconditions:** At the login screen
- **Postconditions:** Should be navigated to the main menu
- **Mainflow:** Student has account
    - User enters in their student email
    - The system will confirm that email is in use
    - The system will then navigate to the main menu with the students account info
- **Alternative Flow:** Student doesn't have an account
    - If the email is a valid school email but has not profile attached yet
    - A new account will be added.
    - The system will then navigate to the main menu with the students account info
- **Exceptions:** When the email submitted is not a Ferris High School email, reject.
- **Related Requirements:** FR-4,5

Use Case 2: Start new game

- **Actors:** Student
- **Triggers:** Player selects 'Starts new game'
- **Preconditions:** Player has logged in and at the start menu screen.
- **Postconditions:** The player will have started a new save file.
- **Mainflow:** Creates new save file
    - The student logs into the game and selects 'new game' from the menu.
    - Since there is no save file for this student account, the system will initialize a new game from the beginning.
- **Alternative Flow:** Creates a new save file by overwriting a old one
    - The student navigates to 'new game' even though they have a previous save file.
    - The system will ask to confirm they want to start from the beginning and delete their previous save.

        - If ‘yes’ is selected, delete the previous save file and initialize a new game from the beginning.
        - If ‘no’ is selected, the system will just return to the start menu’s default state.
- **Exceptions:** System fails to save the game and alerts the student, then asks if they want to try saving again.
- **Related Requirements:** FR-1

Use Case 3: Continue game

- **Actors:** The student
- **Triggers:** Player selects ‘Load Game’
- **Preconditions:** The student has saved progress in the game and is at the main menu.
- **Postconditions:** The student resumes the game from the last saved point.
- **Mainflow:** Loads saved game
    - The student logs into the game and navigates to the main menu
    - The student selects the ‘Load Game’ option.
    - The system loads the saved game state, and the student resumes from the last saved point.
- **Alternative Flow:** Starts new game
    - If no saved game exists, the system prompts the student to start a new game.
- **Exceptions:** If the saved game file is corrupted or missing, the system notifies the student and suggests starting a new game.
- **Related Requirements:** FR-1

Use Case 4: Save Game

- **Actors:** Student
- **Preconditions:** The student is logged into the game and has completed part of a level.
- **Postconditions:** The game progress is saved, and the student can resume the game from where they left off in future sessions.
- **Mainflow:** Autosave
    - The student plays through the game and completes a portion of a level.

    - The student either logs out or closes the game.
    - The system will then navigate to the main menu with the students account info
    - When the student logs back in, the system presents an option to resume the game from the saved state.
- **Alternative Flow:** Manual save
    - The student manually selects the save option from the in-game menu.
- **Exceptions:** If the system fails to save the progress due to an error, the student is notified, and the system attempts to retry.
- FR-1

Use Case 5: View leaderboard

- **Actors:** Student, Teacher
- **Triggers:** Actor selects ‘Leaderboard’ option
- **Preconditions:** Students have completed levels in the game, and scores are logged.
- **Postconditions:** The student views the leaderboard, showing top scores based on level completion and performance.
- **Mainflow:**
    - The student or teacher navigates to the leaderboard from the main menu.
    - The system displays the leaderboard, showing students’ ranks based on their score.
    - The student can view their rank and compare it with other students.
- **Alternative Flow:**
    - The student can look at a more detailed page about their own stats specifically
- **Exceptions:** If no leaderboard data is available, the system informs the user that the leaderboard is empty or still being updated.
- **Related Requirements:** FR-12

Use Case 6: Mini-game

- **Actors:** Students
- **Triggers:** Student navigates to the multiplayer or minigame option in the main menu.
- **Preconditions:** Student is logged in.
- **Postconditions:** The students are in a multiplayer minigame.
- **Mainflow:** Host Lobby
    - The student has logged in and selects minigames on the start screen.
    - The user then selects the host lobby.
    - The system then creates a lobby with a code for the host to share with friends.
    - The student can then select start once other students have joined.
- **Alternative Flow:** Join Lobby
    - The student has logged in and selects minigames on the start screen
    - The student then selects join lobby
    - The student is then asked to enter in a room code inorder to join the lobby, this is shared by the host.
    - Once the student is in, they must wait for the host to begin the game.
- **Exceptions:** If the lobby host is no longer active, pick a new host or abandon all…
- **Related Requirements:** FR-11

Use Case 7: Options

- **Actors:** Student
- **Triggers:** Student selects ‘Options’ option.
- **Preconditions:** The student is logged into the game and has accessed the options menu.
- **Postconditions:** The student customizes the game settings, such as music volume and sound effects volume.
- **Mainflow:**
    - The student navigates to the options menu from the main menu or in-game menu.
    - The system presents various settings such as music volume controls and sound effects volume controls.
    - The student adjusts the settings.
    - The system saves the student’s settings.

- **Alternative Flow:**
    - The student restores default settings.
- **Exceptions:** If the settings fail to save, the system notifies the student and retries saving.
- **Related Requirements:** FR-6, 13

Traceability Matrix

Functional Requirement	Use Case	User Story	Priority
FR-1: Save progress	UC-2: Start new game	As a student, I want my progress to be saved so that when I return to the game, I can continue where I left off without losing my previous progress.	Level 0
FR-1: Save progress	UC-3: Continue game	As a student, I want my progress to be saved so that when I return to the game, I can continue where I left off without losing my previous progress.	Level 0
FR-1: Save progress	UC-4: Save Game	As a student, I want my progress to be saved so that when I return to the game, I can continue where I left off without losing my previous progress.	Level 0
FR-5: Log in to resume save file	UC-1: Log in	As a student, I would like to play the game at home from where I left off at school.	Level 0
FR-6: Difficulty Adjustment and Hints	UC-7: Options	The system must provide an option for students to adjust the game difficulty or access hints when they are struggling to understand the material.	Level 2

FR-11: Multiplayer minigames	UC-6: Mini-game	As a player, I would like to be able to interact with other players while learning about Macbeth	Level 1
FR-12: Leaderboard	UC-5 View leaderboard	As a student, I want to compete with other students to make me want to learn more	Level 2

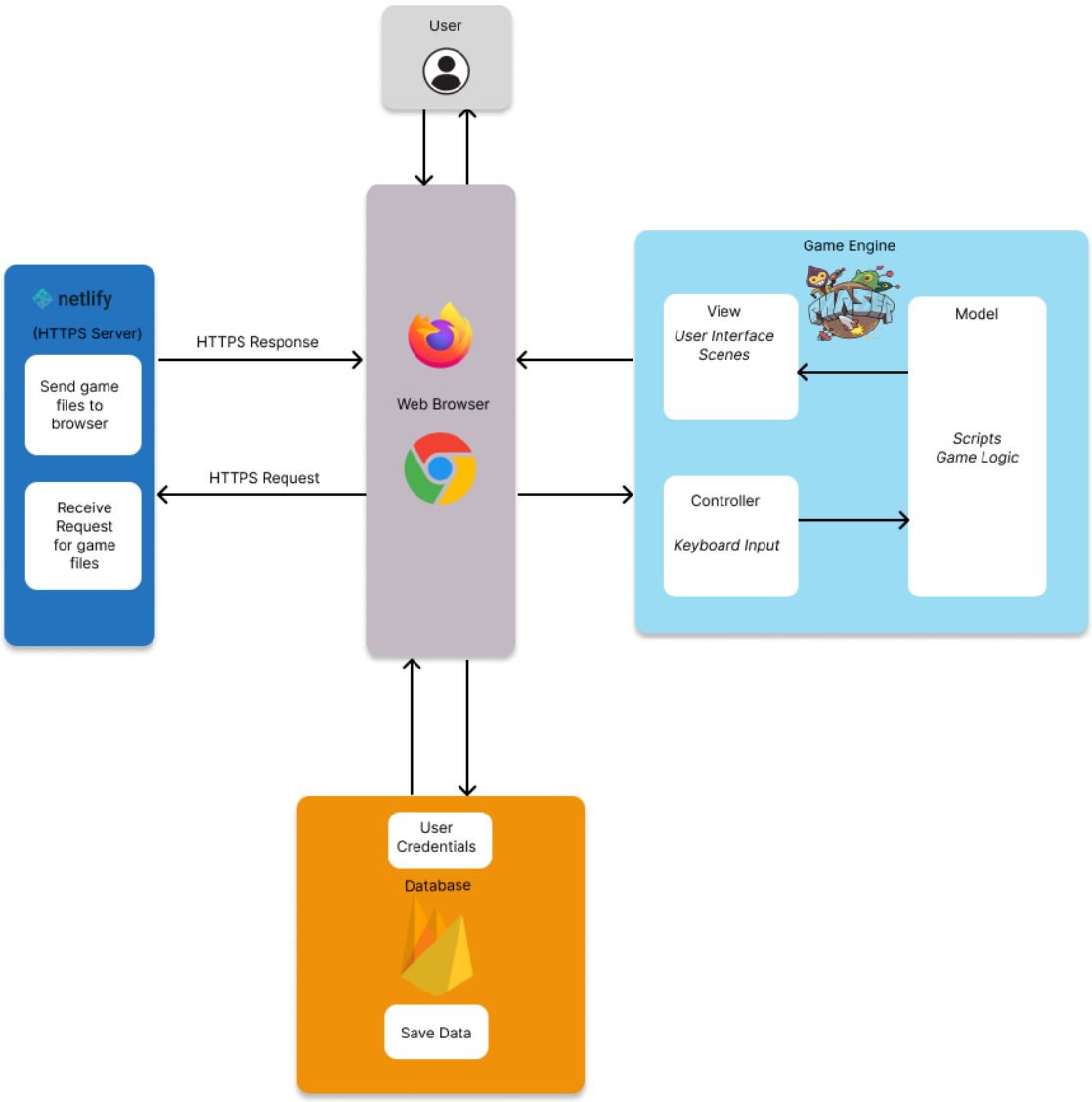
Evaluation of Existing Solutions

The only gamified versions of Shakespearean plays like Macbeth, exist in the form of quizzes/trivia questionnaires. The problem with these current methods of learning Macbeth is that they are not very different from just taking a practice test, which we know students already take plenty of. We are creating an entirely unique method of gamifying the learning and conceptual understanding of Macbeth by turning it into a web based playable RPG.

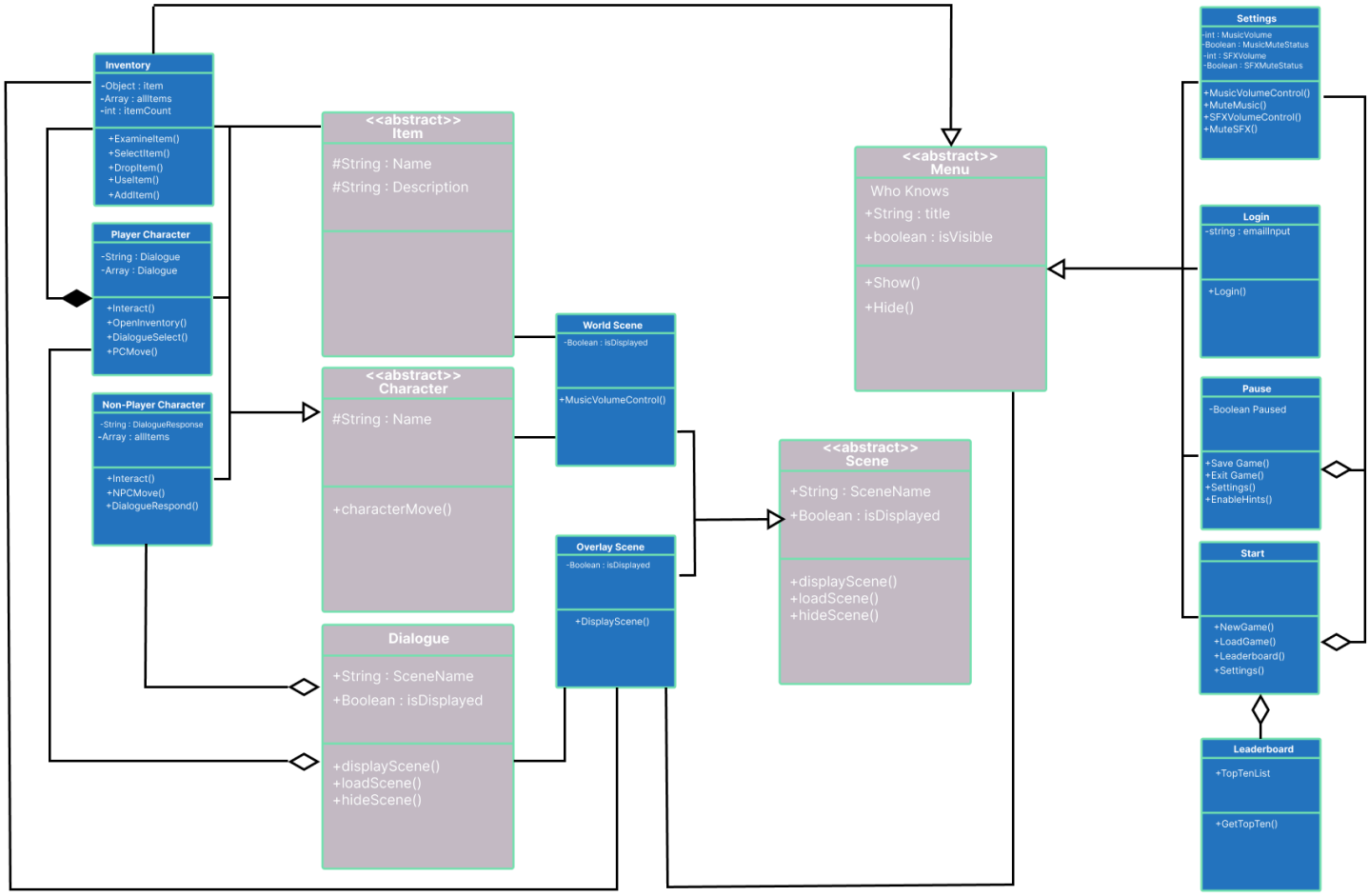
Individual Contributions Summary (Sprint1)

- **Matt:** Worked on use cases, use case diagrams, eval of existing solutions, and nonfunctional requirements.
- **Nathan:** Created Traceability Matrix, filled out functional requirements and use cases, and made modifications to the use case diagram.
- **Justin:** Created documents, setup figma board, worked on use case diagram, worked on evaluation of existing solutions, and user stories.
- **Trevor:** Filled out functional requirements, use cases, and modified eval of existing solutions. Also added a missing user story for login and touches to use case diagram.

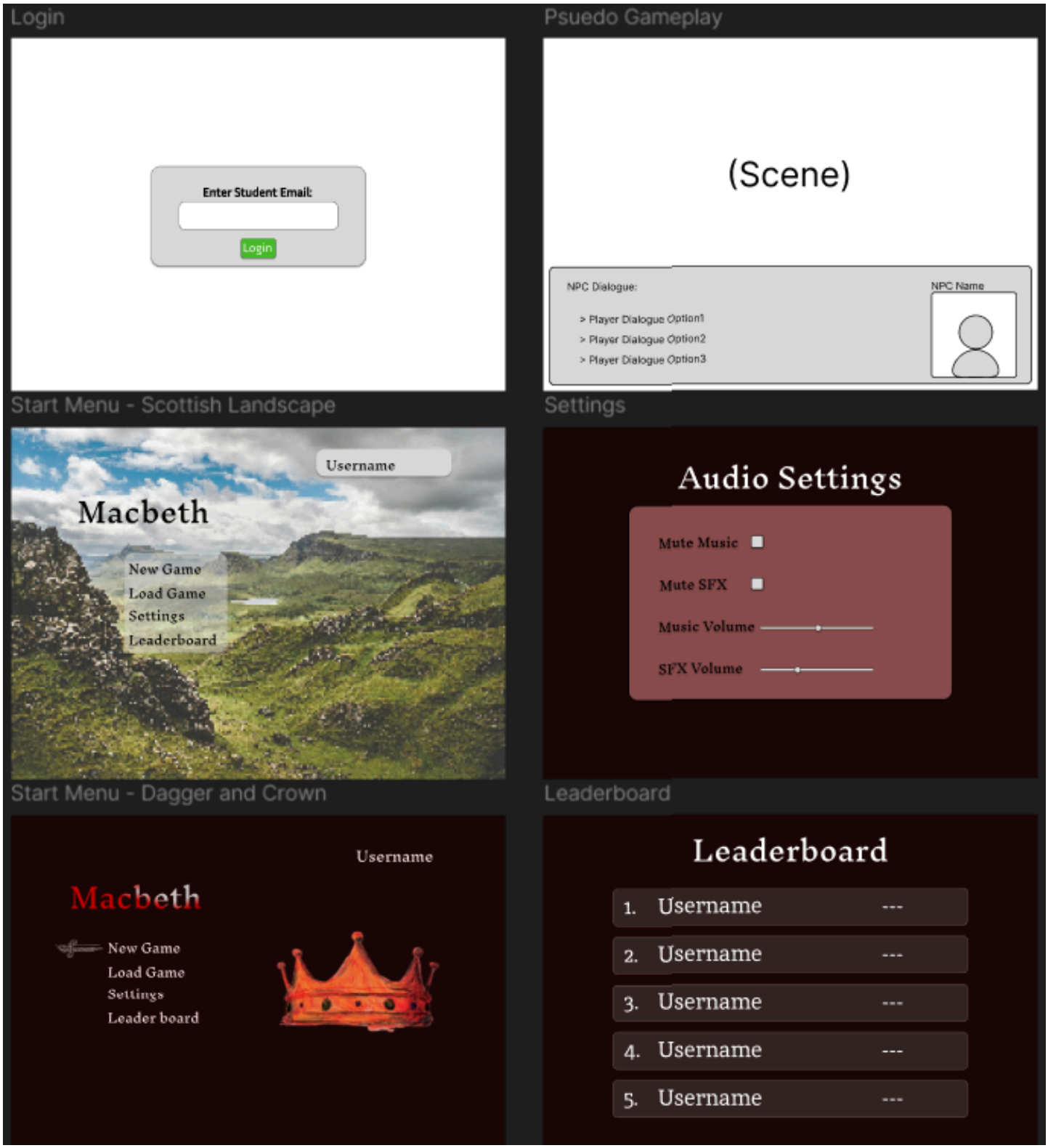
System Architecture



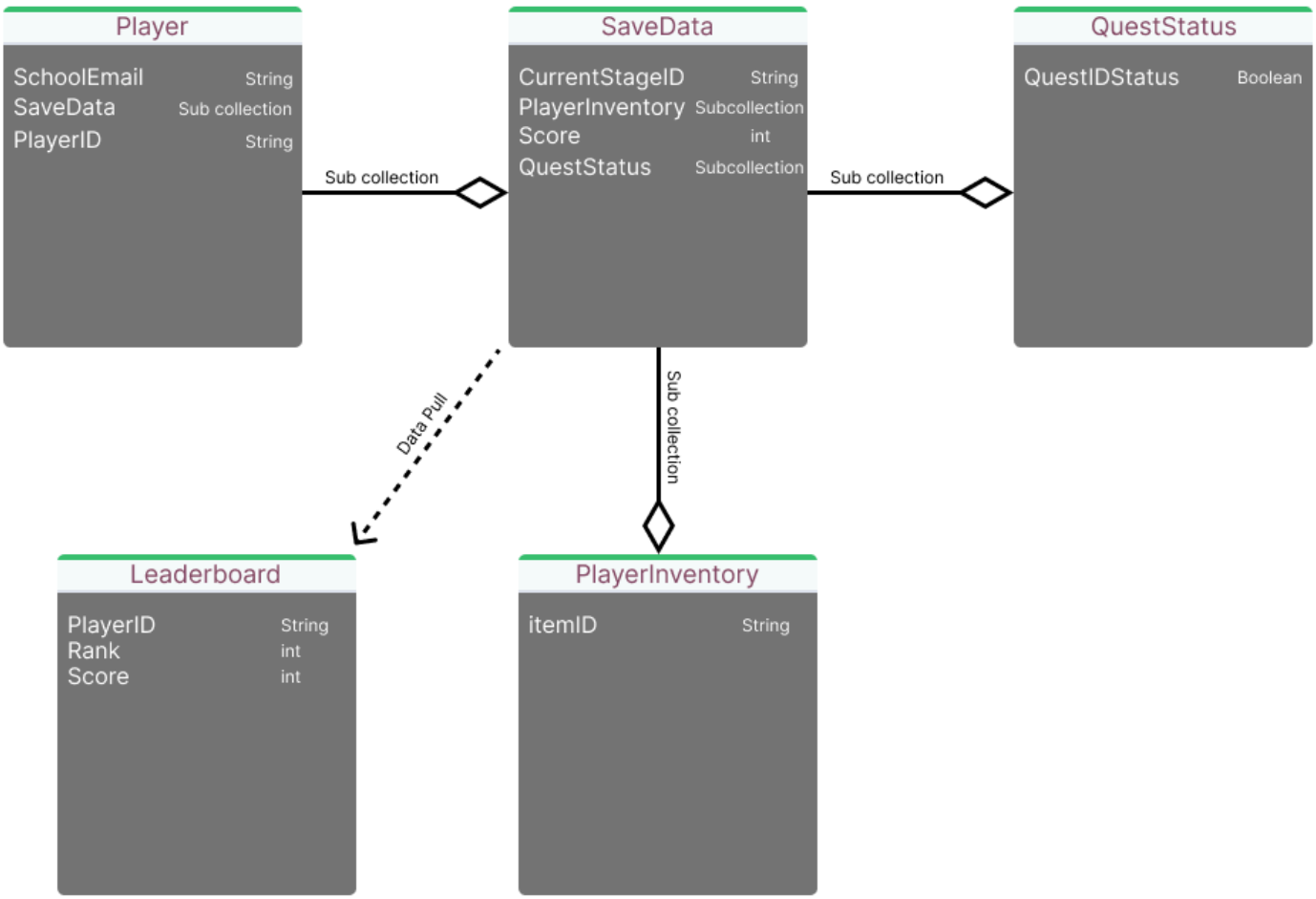
UML Diagram


Inventory
-Object : item
-Array : allItems
-int : itemCount
+ExamineItem()
+SelectItem()
+DropItem()
+UseItem()
+AddItem()
Player Character
-String : Dialogue
-Array : Dialogue
+Interact()
+OpenInventory()
+DialogueSelect()
+PCMove()
Non-Player Character
-String : DialogueResponse
-Array : allItems
+Interact()
+NPCMove()
+DialogueRespond()
<<abstract>>
Item
#String : Name
#String : Description
<<abstract>>
Character
#String : Name
+characterMove()
Dialogue
+String : SceneName
+Boolean : isDisplayed
+displayScene()
+loadScene()
+hideScene()
World Scene
-Boolean : isDisplayed
+MusicVolumeControl()
Overlay Scene
-Boolean : isDisplayed
+DisplayScene()
<<abstract>>
Scene
+String : SceneName
+Boolean : isDisplayed
+displayScene()
+loadScene()
+hideScene()
<<abstract>>
Menu
Who Knows
+String : title
+boolean : isVisible
+Show()
+Hide()
Settings
-int : MusicVolume
-Boolean : MusicMuteStatus
-int : SFXVolume
-Boolean : SFXMuteStatus
+MusicVolumeControl()
+MuteMusic()
+SFXVolumeControl()
+MuteSFX()
Login
-string : emailInput
+Login()
Pause
-Boolean Paused
+Save Game()
+Exit Game()
+Settings()
+EnableHints()
Start
+NewGame()
+LoadGame()
+Leaderboard()
+Settings()
Leaderboard
+TopTenList
+GetTopTen()

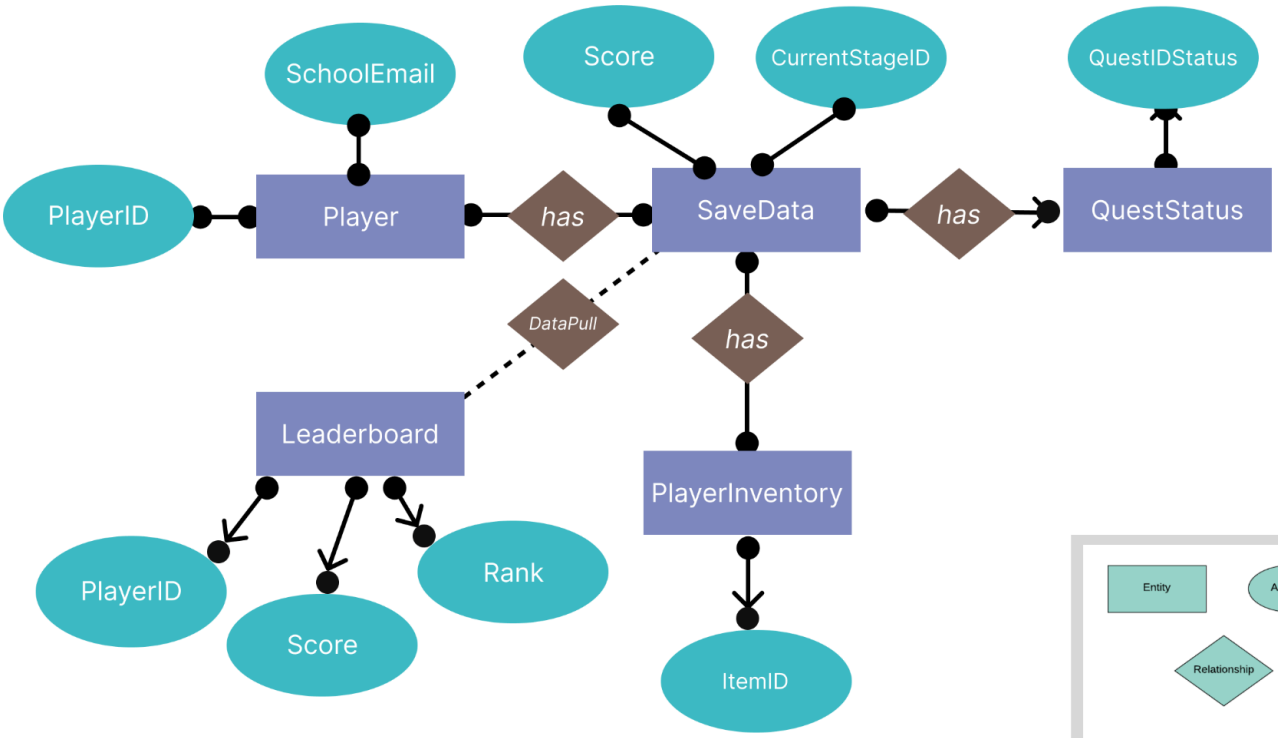
UI Design

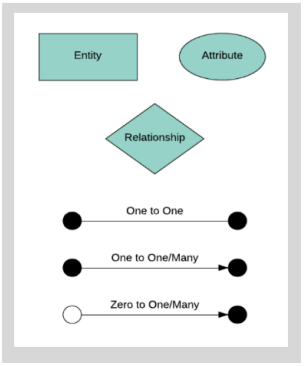

Login
Enter Student Email:
Login
Psuedo Gameplay
(Scene)
NPC Dialogue:
NPC Name
> Player Dialogue Option1
> Player Dialogue Option2
> Player Dialogue Option3
Start Menu - Scottish Landscape
Username
Macbeth
New Game
Load Game
Settings
Leaderboard
Settings
Audio Settings
Mute Music
Mute SFX
Music Volume
SFX Volume
Start Menu - Dagger and Crown
Username
Macbeth
New Game
Load Game
Settings
Leader board
Leaderboard
Leaderboard
1. Username ---
2. Username ---
3. Username ---
4. Username ---
5. Username ---

Data Model

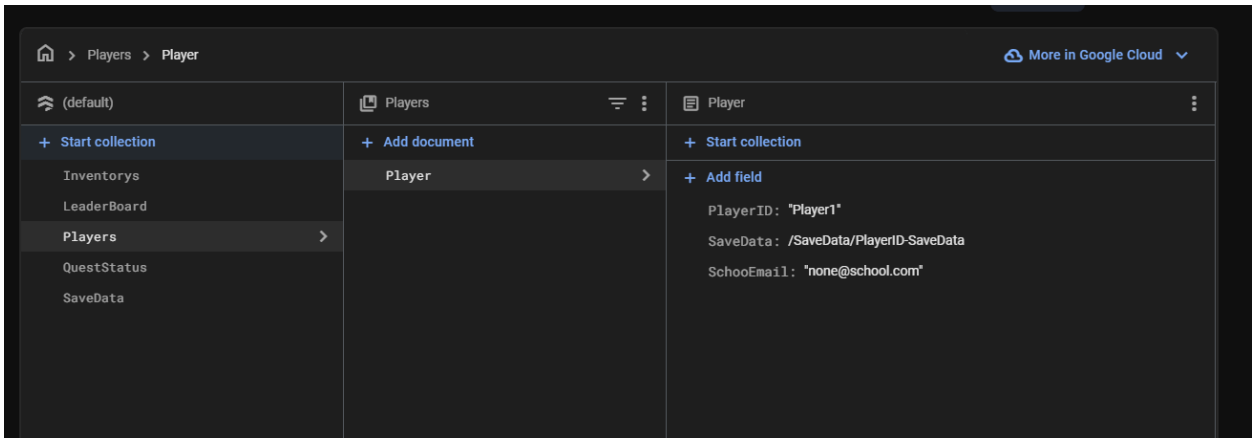

Player
SchoolEmail String
SaveData Sub collection
PlayerID String
Sub collection
SaveData
CurrentStageID String
PlayerInventory Subcollection
Score int
QuestStatus Subcollection
Sub collection
QuestStatus
QuestIDStatus Boolean
Data Pull
Sub collection
Leaderboard
PlayerID String
Rank int
Score int
PlayerInventory
itemID String

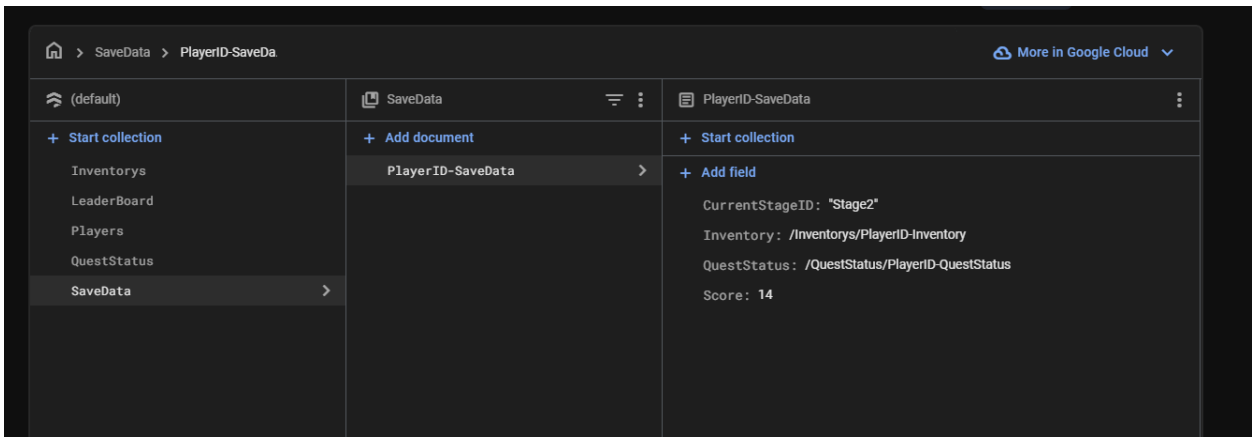
ER Diagram

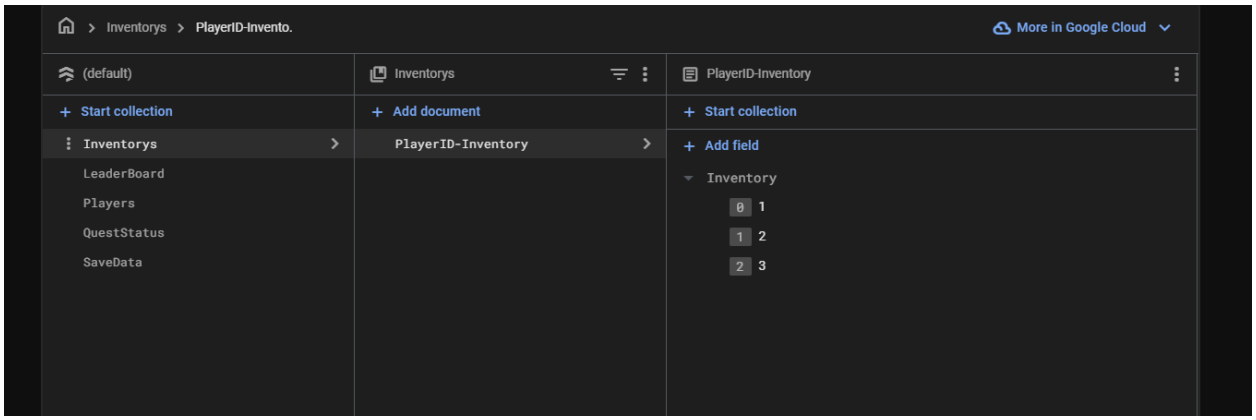

SchoolEmail
Score
CurrentStageID
QuestIDStatus
PlayerID
Player
has
SaveData
has
QuestStatus
DataPull
has
Leaderboard
PlayerInventory
PlayerID
Rank
Score
ItemID

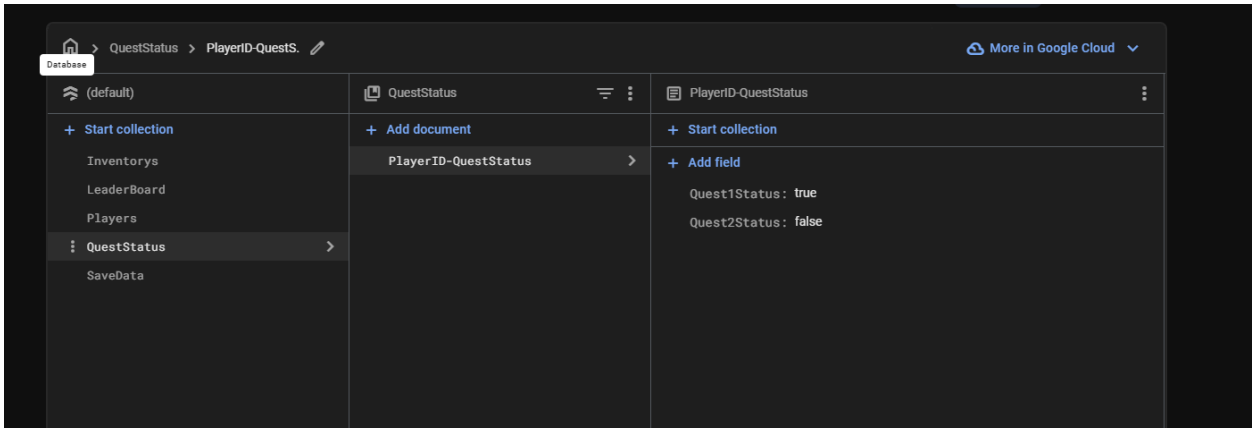

Entity
Attribute
Relationship
One to One
One to One/Many
Zero to One/Many

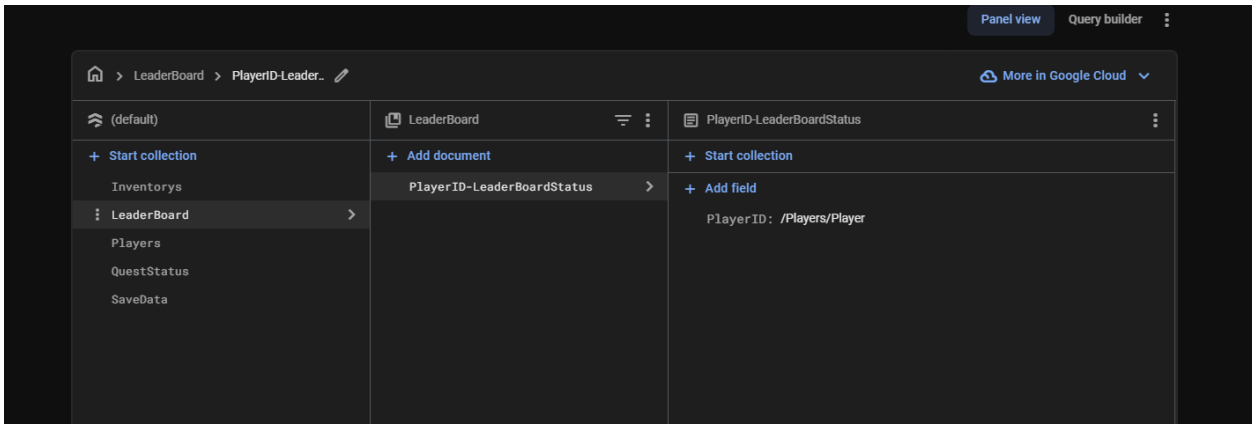
Sample Data


Players > Player
More in Google Cloud
(default)
Start collection
Inventorys
LeaderBoard
Players
QuestStatus
SaveData
Players
Add document
Player
Player
Start collection
Add field
PlayerID: "Player1"
SaveData: /SaveData/PlayerID-SaveData
SchooEmail: "none@school.com"


SaveData > PlayerID-SaveDa
More in Google Cloud
(default)
Start collection
Inventorys
LeaderBoard
Players
QuestStatus
SaveData
SaveData
Add document
PlayerID-SaveData
PlayerID-SaveData
Start collection
Add field
CurrentStageID: "Stage2"
Inventory: /Inventorys/PlayerID-Inventory
QuestStatus: /QuestStatus/PlayerID-QuestStatus
Score: 14


Inventorys > PlayerID-Invento.
More in Google Cloud
(default)
Start collection
Inventorys
LeaderBoard
Players
QuestStatus
SaveData
Inventorys
Add document
PlayerID-Inventory
PlayerID-Inventory
Start collection
Add field
Inventory
0 1
1 2
2 3


Database
QuestStatus
PlayerID-QuestS.
More in Google Cloud
(default)
QuestStatus
PlayerID-QuestStatus
Start collection
Add document
Start collection
Inventorys
LeaderBoard
Players
QuestStatus
SaveData
PlayerID-QuestStatus
Add field
Quest1Status: true
Quest2Status: false


Panel view
Query builder
LeaderBoard
PlayerID-Leader..
More in Google Cloud
(default)
LeaderBoard
PlayerID-LeaderBoardStatus
Start collection
Add document
Start collection
Inventorys
LeaderBoard
Players
QuestStatus
SaveData
PlayerID-LeaderBoardStatus
Add field
PlayerID: /Players/Player

Individual Contributions Summary (Sprint 2)

- **Matt**: Worked on UML, ERD, and system architecture
- **Nathan**: Worked on the UML, the UI, and the system architecture.
- **Justin**: Worked on UML Diagram, System Architecture, Data Design, and setup the environment
- **Trevor**: Helped construct UML Diagram and System Architecture, threw together all the UI mockups.