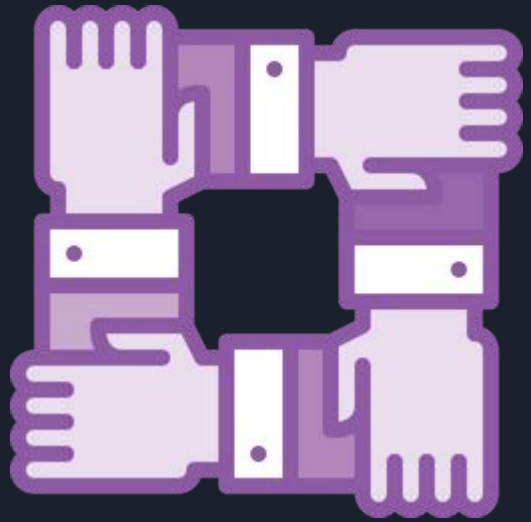
Admin Portal

Team members

- William English
- Daniel Litt
- Cody Miner
- Brian Ringer



Business Requirements

- BR1: We need a system that can create lessons for the official training game
- BR2: We would like to be able to get lessons in the official training game without going through a developer
- BR3: We want to get analytics based on the users of the training software
- BR4: We want to gamify the training software in order to engage users

Use Cases

- UC1: Admin uploads video
- UC2: Admin puts videos into a “lesson” object
- UC3: Admin saves lesson into database
- UC4: User can download lessons into unity application
- UC5: Users view and play lessons with progress tracked
- UC6: Users will complete something and earn an achievement
- UC7: User’s data is collected for analytics
- UC8: Admin can analyze user data
- UC9: Admin can add “calls” to each video

Web App Functional Requirements

- FR1: Web App can upload videos to the database -- HIGH -- BR1

- FR2: Web App can upload “lessons” to the database -- HIGH -- BR1

- FR3: Admin can add videos through the Web App into the database -- HIGH -- BR1

- FR4: Admin can order videos through the Web app into a “lesson” -- HIGH -- BR1

- FR5: Admin can add in the "calls" need to be made on each video -- HIGH -- BR1

- FR6: Admin can log into the Web App -- MEDIUM -- BR1

- FR7: Admin can view tracked analytics from users -- LOW -- BR3

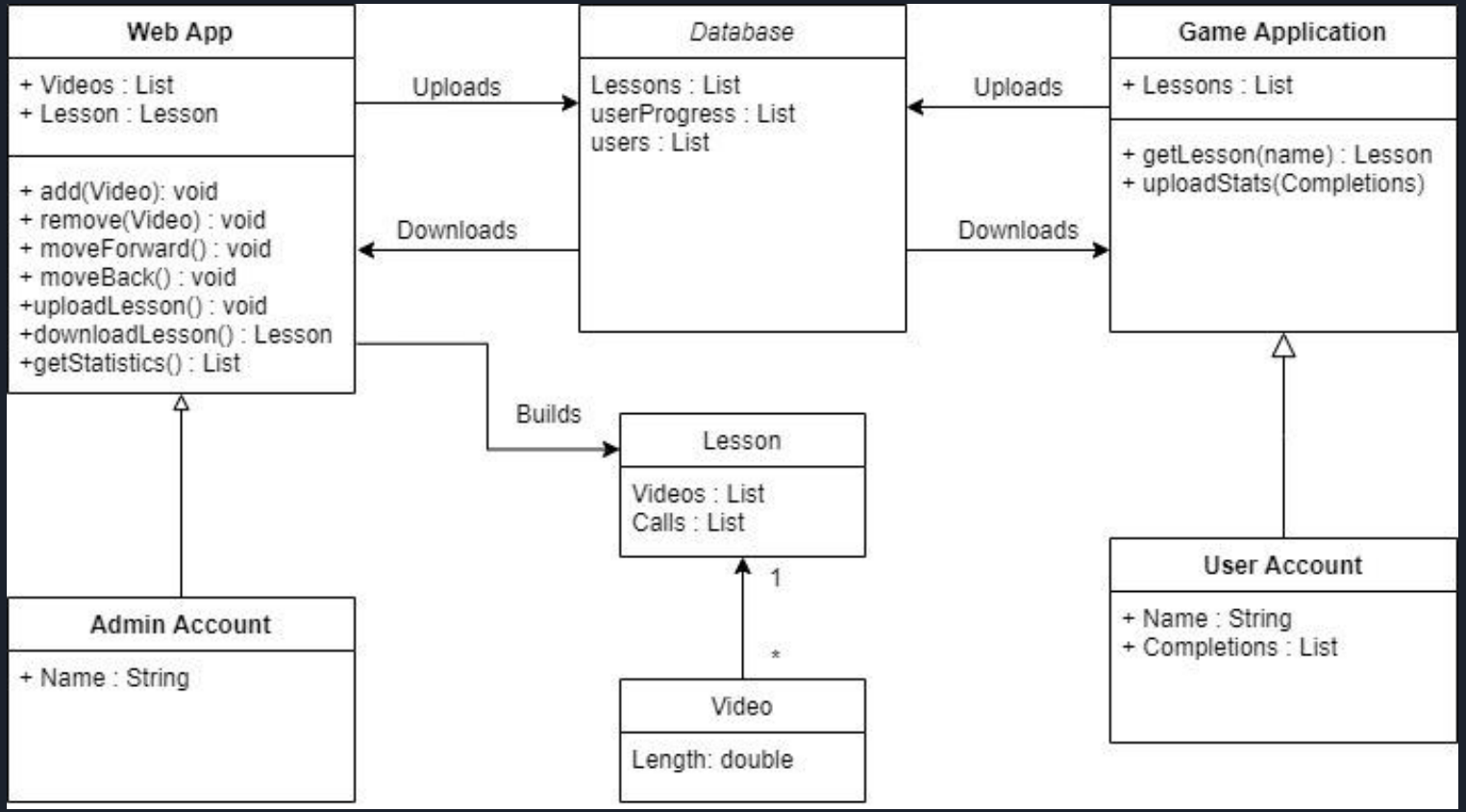
Game App Functional Requirements

- FR8: Users can create an account for the Game Application -- MEDIUM -- BR3
- FR9: Users can log into the Game Application locally -- MEDIUM -- BR3
- FR10: Game Application can download videos -- HIGH -- BR2
- FR11: Game Application can download “lessons” from the database -- HIGH -- BR2
- FR12: Users can download videos to Game Application -- HIGH -- BR2
- FR13: Users can download “lessons” to Game Application -- HIGH -- BR2
- FR14: User progress is stored locally -- HIGH -- BR4
- FR15: User progress is stored in the cloud -- MEDIUM -- BR3
- FR16: User analytics are tracked -- LOW -- BR3
- FR17: Game Application can upload tracked user data -- MEDIUM -- BR3
- FR18: Game Application can display incomplete achievements to users -- LOW -- BR4
- FR19: Game Application can track completed achievements -- LOW -- BR4
- FR20: Game Application can display completed achievements -- LOW -- BR4
- FR21: User’s local analytics are push when they are internet -- LOW -- BR4

Non-Functional Requirements

- PER1: Video quality 1080p and 4k and individual videos capped at 500 Mb (should not exceed 2 min) -- MEDIUM -- BR1
- PER2: Video can be uploaded and processed in 15 minutes -- HIGH -- BR1
- USE1: Training should take less than 30 minutes, all in-app -- LOW -- BR1

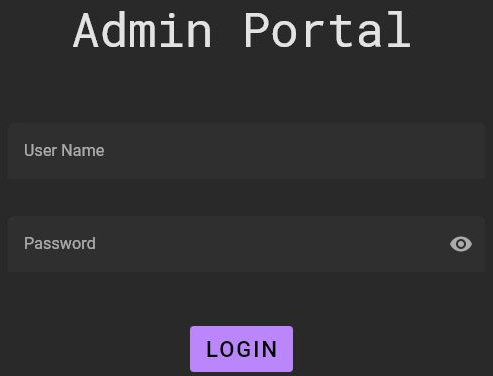
Domain Model

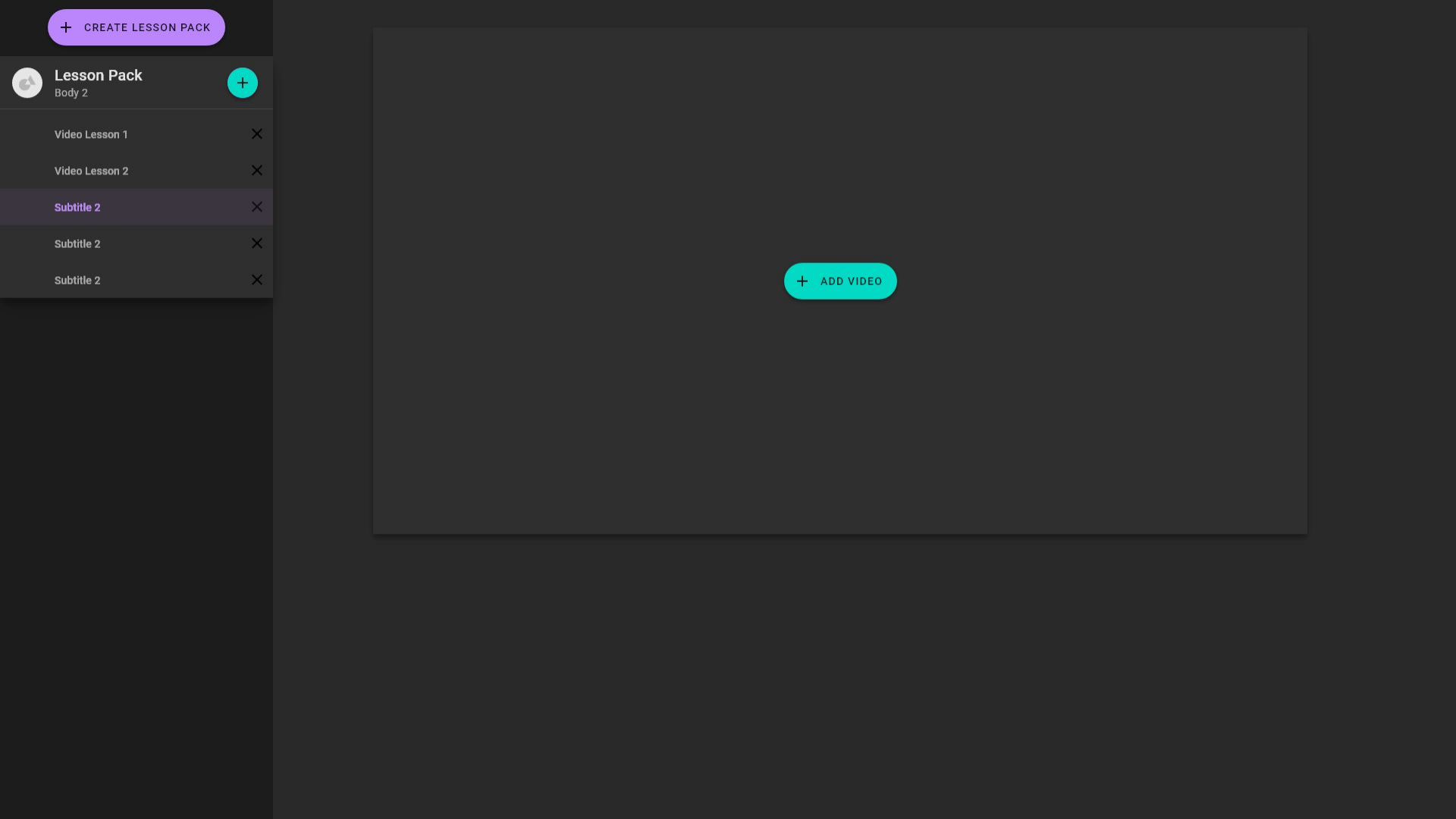


Tech Stack

- **Unity** - We are building off of an existing product that has been built with Unity.
- **React JS** - The functionality of the lesson maker and the data visualizations will require more than regular HTML, CSS, and JS, so we believe that this is the best solution for those requirements.
- **Firebase** - Firebase is a cloud computing solution with competitive pricing for applications without much data or many read/writes.
- **Github** - For version control, we will take advantage of Git/Github. This is the version control we are all used to and has a track record of working.
- **Freedcamp** - https://freedcamp.com/ - This is the client's preferred project management tool.

https://xd.adobe.com/view/b1275bf7-866c-4221-8389-899349c1f0ba-e474/




CREATE LESSON PACK
Lesson Pack
Body 2
Video Lesson 1
Video Lesson 2
Subtitle 2
Subtitle 2
Subtitle 2
ADD VIDEO

CREATE LESSON PACK

Lesson Pack
Body 2

- Video Lesson 1
- Video Lesson 2
- Subtitle 2
- Subtitle 2
- Subtitle 2

Call Options

- Correct call
- Call 2
- Call 3

First Iteration Features

- Web App can upload point-of-view videos to the database -- HIGH
- Admin can log into the Web App -- MED
- Game Application can download point-of-view videos -- HIGH

What we changed because of our mentor

- Ask more about what they are going to do with the data collect
- Wanted to go more in depth about how the data is going to be stored and whether it will be uploaded
- Get clarification of what training entails for USE1.
- Domain model was changed
- External application should handle server communication
- Suggested not to use JSON but just use SQL
- Thought first iteration features might be excessive -- make sure to add most value (agile) or make the database local in first iteration
- We should stop using Zoom because it keep cutting us off

What we changed because of our client

- Need to add instruction video to prototype otherwise he enjoyed the look
- We forgot a use case where the admin can add “calls” to the videos
- Planning on better defining the terms used for his domain
- Agreed to scale back first iteration features like our mentor said
- The specifics of what data will be collected
- How to discuss the lesson and what language to use internally


Pluto