FIELD OF VISION COMPANION APP
Development Report

FIELD OF VISION

Group 21
Andrew McDonald
Andrew Meehan
Luke McGrath
Aislinn Smyth
David Deneher
John Kommala

INDEX

Index

1 Introduction **2**
- 1.1 Background and Problem Statement 2
- 1.2 Technical Approach 2

2 Requirements **3**
- 2.1 Functional Requirements 3
- 2.2 Non-functional Requirements 3
- 2.3 User Interface Scenarios 6

3 Design **7**
- 3.1 Architecture Diagram 7
- 3.2 UML Class and Sequence Diagrams 9

4 Implementation **10**
- 4.1 Tools, Libraries, Platforms 10
- 4.2 User Interfaces 11
- 4.3 Algorithms 12

5 Conclusions **13**
- 5.1 Design and Implementation 13
- 5.2 Project Objectives 13
- 5.3 The Team 14
- 5.4 Algorithms 14

Appendix I: Requirements Document **15**

Appendix II: Software Design Specification **23**

CS2013 – CS3013: Development Report

1. Introduction

1.1.Background & Problem Statement

The project that our group received was Project Number 40: Field of Vision - Project 1: NFC or Bluetooth enabled mobile app for visually impaired users to communicate with a touch-based tablet.

Field of Vision is a mobile application created for a touch-based tablet that our client has created. The project is to enable visually impaired sports fans to experience football matches, they do this by feeling the position of the ball along with every shot, tackle, and goal in real-time. It is an app which is accessible and easy to use. The app controls the tablet device, it gathers the balls coordinates from Amazon Web Servers and sends those coordinates directly to the tablet to direct the ball. The app also plays the audio sound of the match and on the controls page gives options to alter the volume and the play/pause button.

1.2.Technical Approach

The group itself consisted of six members, so we decided in order to get the application done as efficiently as possible we would split the group into the front end and the back end. Both Andrews and David worked on the back end and Aislinn, John and Luke worked on the front end. We found it a lot easier to communicate within our smaller groups and therefore made splitting up the work a lot easier. About twice a week both the back end and front end teams would come together to discuss progress. We chose to develop and work on the application using flutter and Android Studio. The front end of the project itself consisted of a cross-platform app that contains the interface as a whole. The app was developed using the language DART, we were able to collaborate on the code by uploading the code to Github to allow all members to access and modify the code as they wish.

2. Requirements

2.1.Functional requirements

- Display a selection of highlights.
- Allow the user to select a highlight for viewing.
- Send signals to the device in real time containing data related to the highlight that is being viewed.
- Allow the user to play, pause, rewind and toggle other settings related to the playback of the highlights.

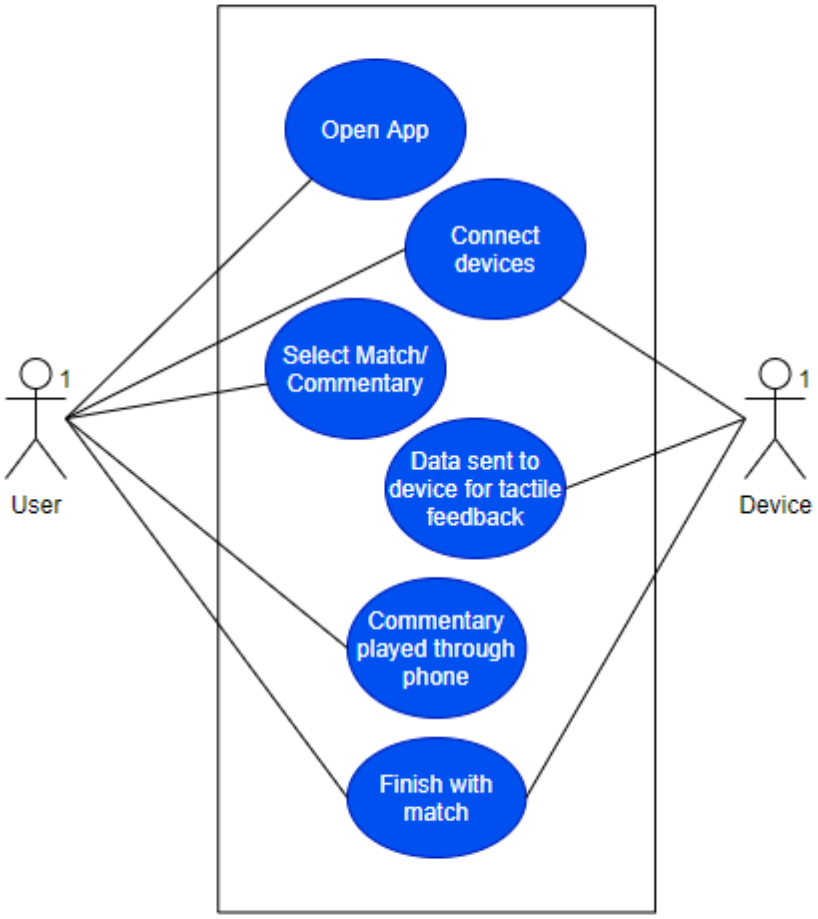
2.2.Non-functional requirements

- User interface should be minimal but accessible and aesthetically pleasing.
- Signals should be sent from the app to the device as quick as possible in order to provide real time updates of the highlight being displayed.
- User data should be GDPR compliant.

2.3.User Interaction Scenarios

Use Cases

Diagram



Textual Descriptions

Use Case Name	Select Match/Commentary
Actors	User's phone
Preconditions	User has the app opened, they're connected to the device, and they have an internet connection
Descriptions	User has started the app and needs to select which match they would like to experience
Postconditions	User is able to experience the game using the device they're connected to
Alternate flows and exceptions	User closes the app and does not select a game to be played

Use Case Name	Data Sent to Device for Tactile Feedback
Actors	Device
Preconditions	The app is open on the phone, a match has been selected, and playback has started
Descriptions	The game data is sent to the device and is outputting the sensory information to allow the user to experience the games
Postconditions	The user has "watched" the match using the information sent to the device
Alternate flows and exceptions	There is a transmission error or packet loss that results in interrupted transmission of the game

Use Case Name	Connect Devices

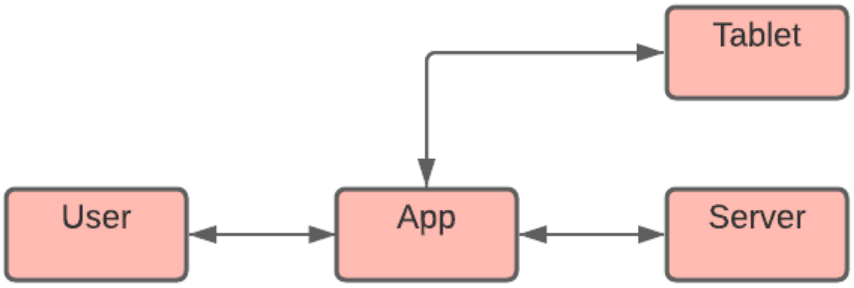
Actors	User’s phone and Device
Preconditions	Both devices are turned on and in range of each other
Descriptions	Uses Bluetooth to allow for connectivity between the phone which communicates the match information to the paired device
Postconditions	The information is then transmitted to the device and the user is able to experience the match
Alternate flows and exceptions	No exceptions

3. Design

Our project consists of a mobile application, compatible with both android and iOS, built using flutter that is meant to be used as a controller for the Tablet from Field Of Vision. Our application allows a user to connect to a tablet, browse games in range and then play the corresponding audio and be able to control the playback. The application consists of three pages, a main screen, a screen that displays all the available matches(games), and a control screen to control the audio playback. The application communicates with the server-side part of the project using an AWS (Amazon Web Services) database.

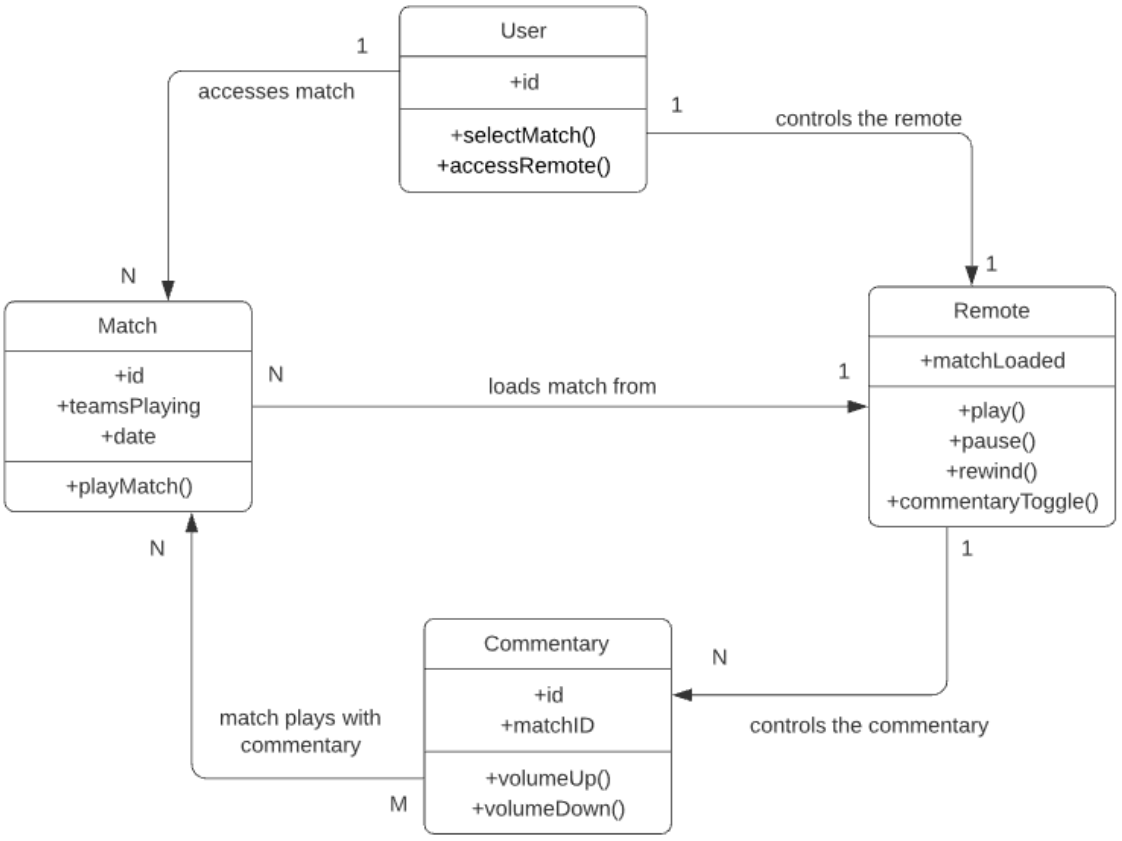
3.1.Architecture Diagram

Simplified Architecture Diagram

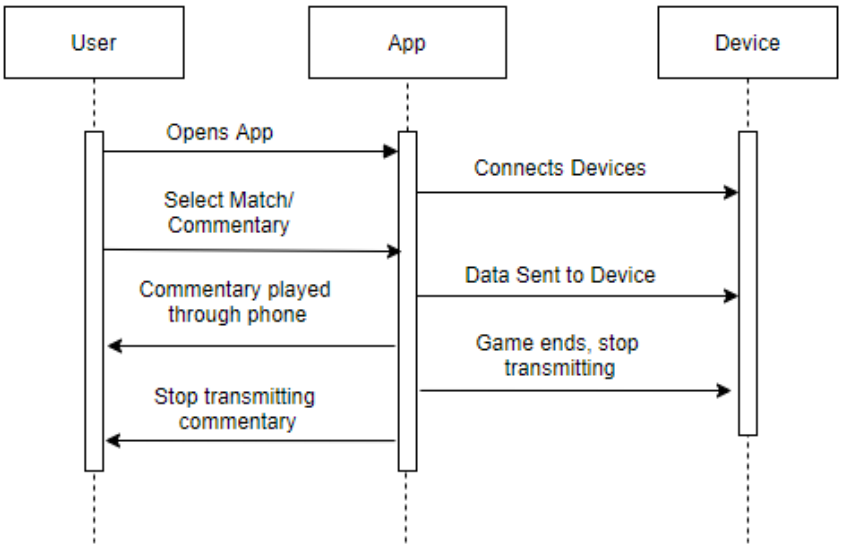


3.2.UML Class and Sequence Diagrams

3.2.1 Class Diagram



3.2.2 Sequence Diagram



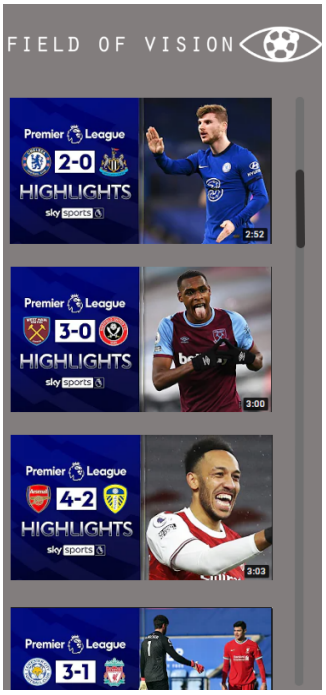
4. Implementation

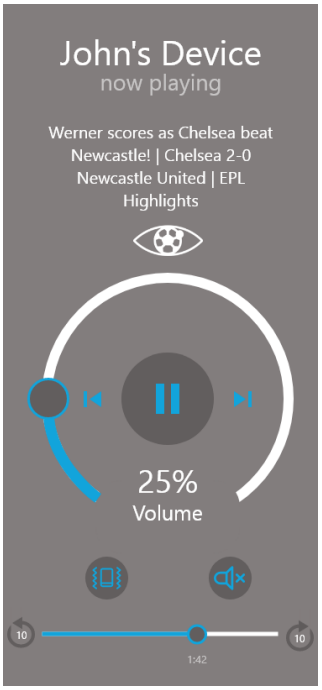
4.1.Tools, Libraries, Platforms

- We used Android Studio as the primary development platform. This was ideal as it linked with Flutter and allowed us to use the android and ios emulator to trial our app. All team members had to familiarize themselves with the platform before undertaking their designated section to complete.
- We used Amazon Web Services (AWS) for the back-end when testing the server communication with the device. Using AWS we could set up the ESP32 chip that the device used with the Internet Of Things (IOT) functionality of AWS. We used this to test sending coordinates from the server to the device itself.
- Firebase was used for testing the backend communication within the application. It was easy and to set up and test a database and it linked well with Flutter so allowed us to focus on the accessibility and front end.
- GitHub was used for version control, and the primary code repository. It was necessary that the repository was configured to allow all other teams involved in the various aspects of the project to integrate accordingly. We had to ensure a correct method of version control was adhered to, in order to deliver and communicate the project to the standard agreed with the client, and to ensure a log of all work and changes were recorded in the event of backreferencing.
- We used Discord to communicate as a team. Discord was used as it has all the necessary tools needed for collaboration. We could video call for meetings with each other and with clients. We could share our screen to collaborate with each other on code and we had separate channels for general info, back-end and front-end to ensure no one missed any relevant info.
- Arduino was used to communicate with the device itself. In order to make the device follow the position of the ball in real time we had to write Arduino code that moved it accordingly. We also needed to write code to let the device communicate with the servers. This code could then be uploaded to the chip inside the device and controlled via our app.
- We used Adobe XD for any mockups needed for the app. We used this as it is very intuitive and easy to use and is the industry standard for mockups and designs for apps and websites.
- Fantasy Premier League was used to host our team's fpl league to boost team-building. David is winning by a large margin as of writing this.

4.2.User Interfaces

We initially designed mockups based on the clients specifications using Adobe XD. We





designed it to be simple, yet accessible to visually impaired users.

Whilst the design for the video selection scene was fine this initial control screen proved to be too complicated to implement and more importantly too complex for visually impaired fans

to use. We

simplified this screen for our final design, and added a home screen for ease of use. Here we have displayed our interfaces on both an Apple (first two photos) and Android device (last photo).

4.3.Algorithms

The main algorithm we needed to work on for our project was the algorithm used to move the position of the ‘fingerpiece’ on the device itself. This algorithm had to take the coordinates of the position of the ball that were collected every second and move the device accordingly. To do this we used a for loop that multiplies the coordinates by the height and width of the device and used a formula to calculate the speed at which the device should move between these two coordinate based on when they were recorded in the game.

5. Conclusions

5.1.Design and Implementation

We encountered a few aspects that we needed to address in the implementation of the design. The initial problem we encountered was that none of us had any experience with flutter or with the language DART, so we had to take a few tutorials on YouTube and on LinkedIn.

The main problem we encountered throughout this project was making sure our application was a cross-platform app. At the beginning we only tested the application on an android emulator but later once we tested it on an IOS we quickly realised it did not fit the screen and had to alter our code to suit any size screen and any operating system. Tying into that, we also had pixel overflow issues on certain emulators which showed to us we had hard coded dimensions and had to go back and fix that also.

Another aspect we encountered during the design and implementation of the project was that we needed to be designing an application from a completely different perspective. The target audience for our app were visually impaired individuals, therefore when designing we had to keep in mind that although the app should be aesthetically pleasing, it also had to be practical toward the individuals it was aimed toward. For the homepage, we used contrasting colours in order to make the “Get Started” button stand out. The background of the screen does not vary in colour so to not over stimulate the users eyes.

5.2.Project Objectives

Overall, we did a fantastic job at reaching our project objectives. We had weekly meetings and milestones set to keep ourselves from falling behind and we managed to meet all the milestones successfully. We worked very well as a team, we managed to fulfill all the requirements our client set and we set for ourselves at the beginning of this module and in the end we managed to bring all the components of the project together to create our end product. We are very content with the finished product of our project and believe we worked well as a collective team.

5.3. The Team

We all benefited massively from doing this project, it gave us an opportunity to develop and improve our communication skills. It also forced us to become comfortable with relying on each other and developing trust in each other and really helped with our team work. We worked well collectively, if anyone had any issues or problems there was always someone online to help them figure it out. Due to the current circumstances of COVID19 we faced a few challenges such as poor connection, time differences and just the general lack of in person meetings. Luckily, due to how well we communicated and everybody's will to work, we managed to reach our weekly milestones. Although our circumstances were strange and different, we still were able to keep in contact with each other, our demonstrator and our client.

5.4. Algorithms

Overall, the implementation of our algorithms was of a very high-level. We experienced a few issues related to cross-platform implementation issues. Our lack of parameter coding caused issues when testing the app among different operating systems and screen sizes. We had a lot of Pixel overflow as we used an android emulator more so than an IOS emulator. We solved this issue by passing parameters as opposed to hard coding a set size and therefore the efficiency of our algorithm went up greatly. Finally, the algorithms we implemented adhered to the efficiency standard we set and satisfied the clients needs, we believe our algorithms outperformed our expectations in regard to the efficiency and performance within the interface.

CS2013/CS3013

Requirements Document

Group 21

David Deneher
John Kommala
Aislinn Smyth
Andrew Mc Donald
Luke Mc Grath
Andrew Meehan

Introduction

1.1 Overview

The purpose of the system we are designing is to provide a way for visually impaired football fans to experience live sporting events. Our app should be able to be used in conjunction with the physical device made by Field Of Vision to help them to achieve their goal of making football accessible to the visually impaired fanbase.

1.2 Scope

For the Field of Vision application our scope is to design and implement a companion app to control the current Field of Vision product, focusing on the accessibility of the app, bluetooth functionality, the ability to play, pause and rewind the match and the ability to access past matches.

1.3. Objectives and success criteria

The main objective of this project is to create an application for the visually impaired, which allow visually impaired people to watch or re-watch any football match of their choosing, and to implement key features into our application which allow us to correctly and efficiently implement this. The key features to have achieved by the completion of this project are that the application must be compatible with both IOS and Android, accessible, bluetooth functionality, and other extra key features previously agreed with the client.

To ensure we meet each deadline and we deliver our project to the highest quality we possibly can in the designated ten weeks we have outlined a few key components each team member has agreed to follow:

1. Constant communication between team members.
2. Implementing features previously agreed with the client.
3. Following our timeline closely.
4. Realistic expectations.
5. Weekly milestones.
6. Individual work ethic.
7. Dedicated and reliable team members.

2. Current system

The current system was developed by the FOV team using Arduino. The system features an Arduino tablet with a physically moving cursor. In the current system the coordinates are manually inputted which will determine the movement of the physical cursor in the tablet.

3. Proposed System

3.1. Overview

The proposed system will essentially be an NFC or Bluetooth enabled mobile app that allows visually impaired users to communicate with a touch-based tablet device. The app will have an accessible design and will be very easy to use and navigate. There will be a menu for selecting highlights. Once a match is selected there will be a remote that has buttons to allow the user to play, pause, rewind, toggle vibrations and toggle commentary as they wish. The app will send signals to our device via bluetooth.

3.2. Functional Requirements

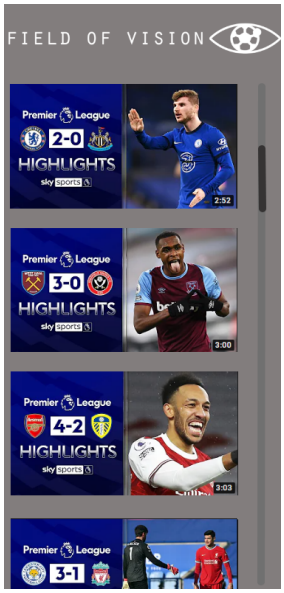
- Display a selection highlights
- Allow the user to select a highlight for viewing
- Send signals to the device in real time containing data related to the highlight that is being viewed
- Allow the user to play, pause, rewind and toggle other settings related to the playback of the highlights

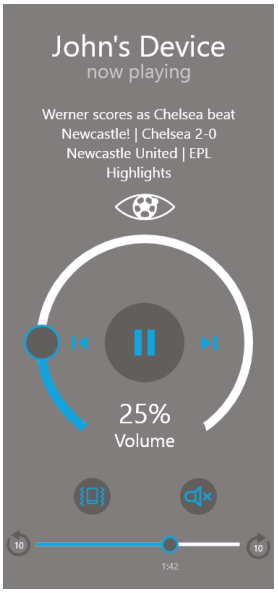
3.3. Non-functional requirements

- User interface should be minimal but accessible and aesthetically pleasing
- Signals should be sent from the app to the device as quick as possible in order to provide real time updates of the highlight being displayed
- User data should be GDPR compliant

3.4. System prototype (models)

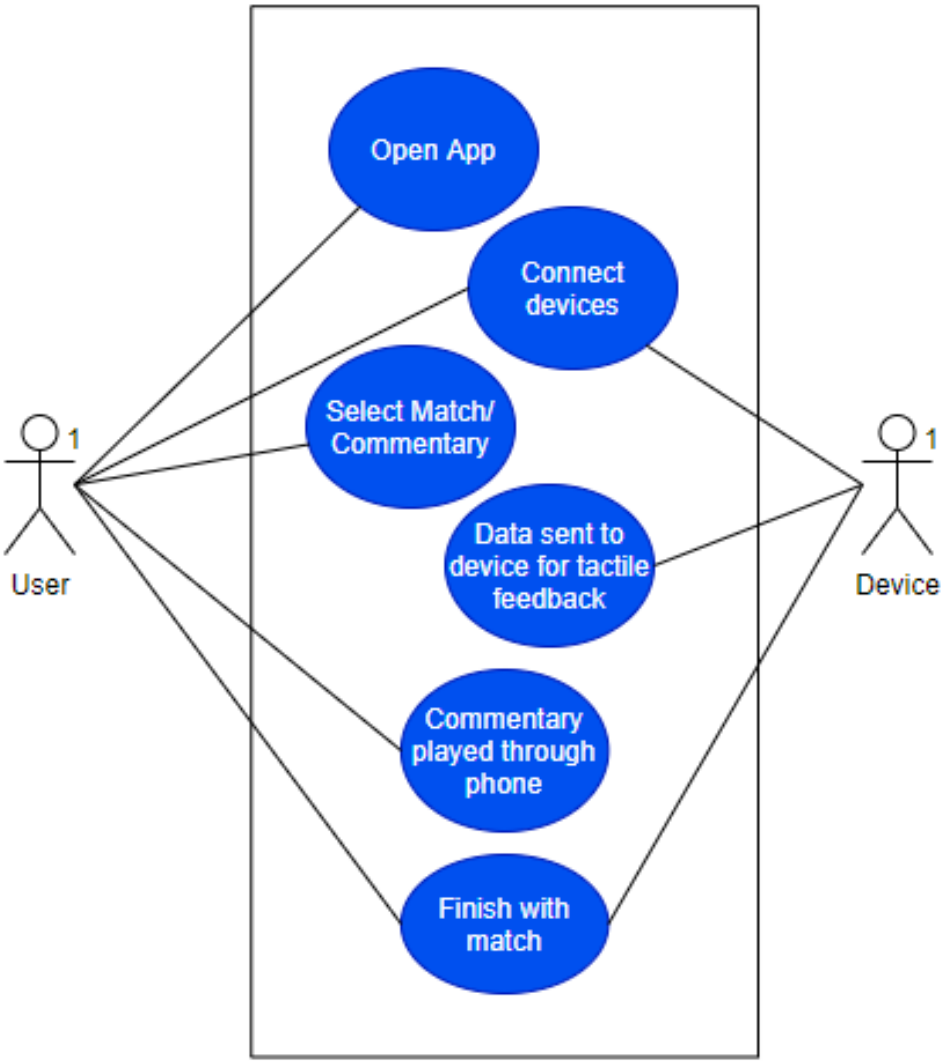
3.4.1.User interface mockups





For the user interface mockups of the app I used Adobe XD, the industry standard for user experience design for web and mobile apps. I wanted to design a simple interface that would be accessible for visually impaired users, but still was simple to use for non-visually impaired users. I made the home page a simple scroll page where the user could pick a highlight from the selection. Once they select the highlight I designed the next page, the controller. Here I based the design off the Google Home controller and added a volume and progress bar and buttons for commentary, pause/play, fast forward/rewind and vibration.

3.4.2.Use cases (including text narratives)

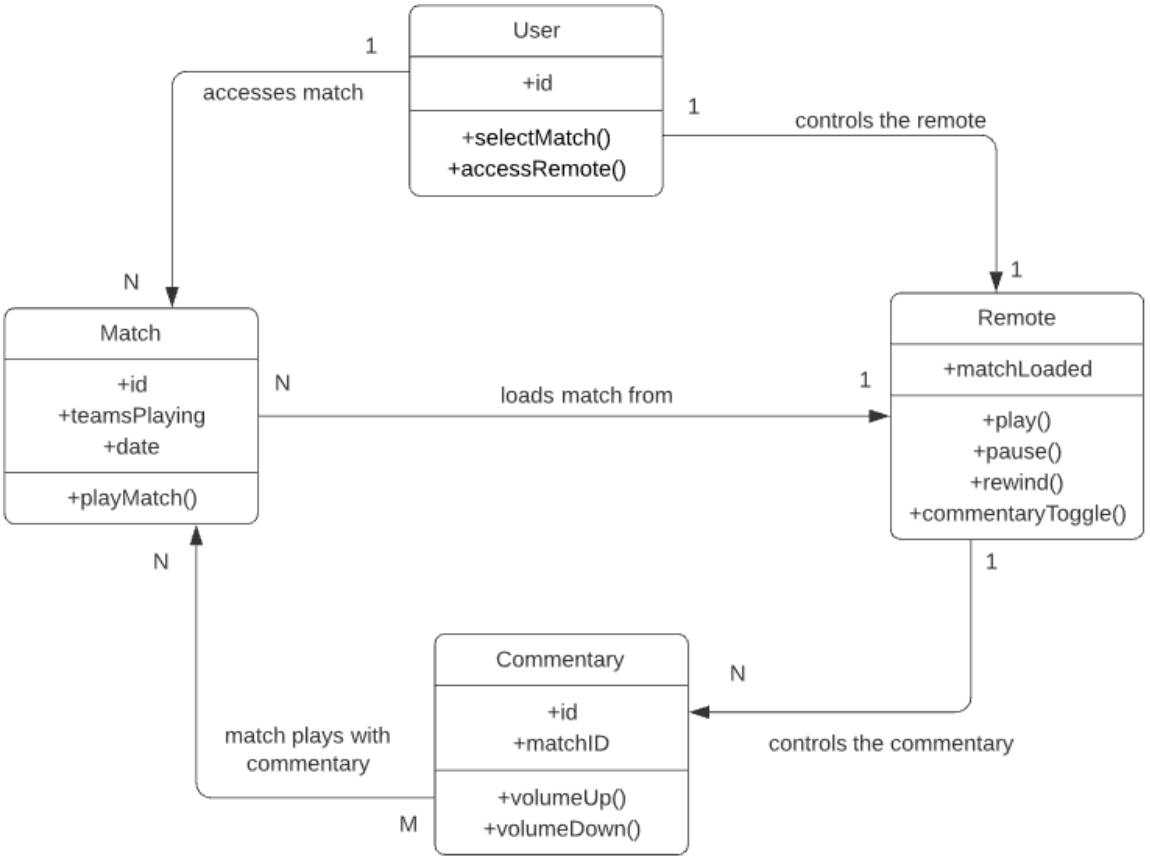


Use Case Name	Select Match/Commentary
Actors	User's phone
Preconditions	User has the app opened, they're connected to the device, and they have an internet connection
Descriptions	User has started the app and needs to select which match they would like to experience
Postconditions	User is able to experience the game using the device they're connected to
Alternate flows and exceptions	User closes the app and does not select a game to be played

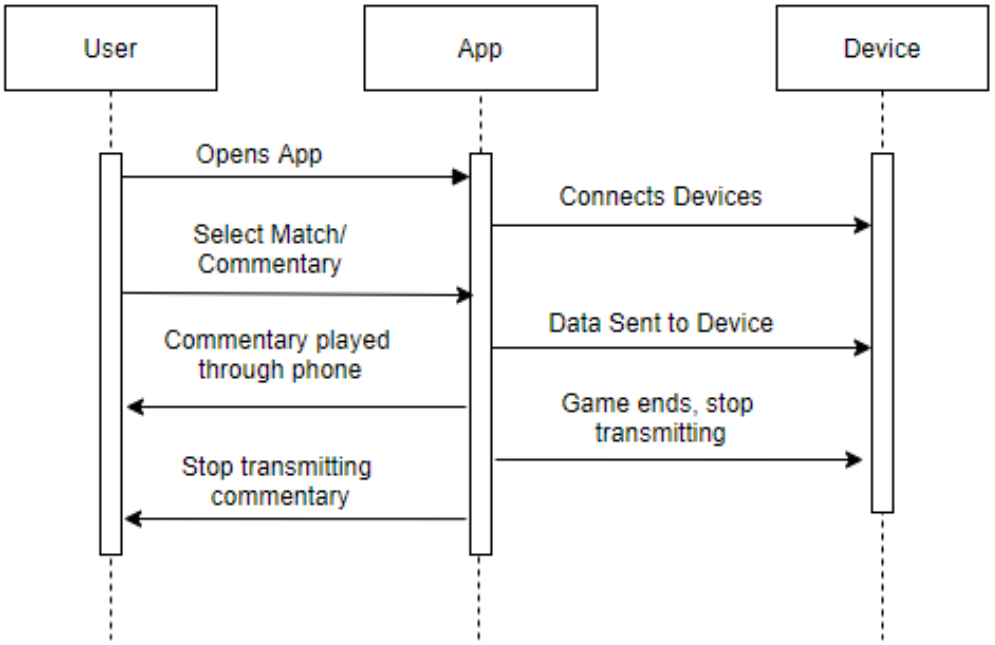
Use Case Name	Data Sent to Device for Tactile Feedback
Actors	Device
Preconditions	The app is open on the phone, a match has been selected, and playback has started
Descriptions	The game data is sent to the device and is outputting the sensory information to allow the user to experience the games
Postconditions	The user has "watched" the match using the information sent to the device
Alternate flows and exceptions	There is a transmission error or packet loss that results in interrupted transmission of the game

Use Case Name	Connect Devices
Actors	User’s phone and Device
Preconditions	Both devices are turned on and in range of each other
Descriptions	Uses Bluetooth to allow for connectivity between the phone which communicates the match information to the paired device
Postconditions	The information is then transmitted to the device and the user is able to experience the match
Alternate flows and exceptions	No exceptions

3.4.3.Object model



3.4.4.Dynamic model



CS2013 – CS3013: Software Design Specification Outline

1. Introduction

1.1. Overview - Purpose of system

The purpose of the system we are designing is to provide a way for visually impaired football fans to experience live sporting events. Our app should be able to be used in conjunction with the physical device made by Field Of Vision to help them to achieve their goal of making football accessible to the visually impaired fanbase.

1.2. Scope

For the Field of Vision application our scope is to design and implement a companion app to control the current Field of Vision product, focusing on the accessibility of the app, bluetooth functionality, the ability to play, pause and rewind the match and the ability to access past matches.

1.3. Definitions, abbreviations

- AWS - Amazon Web Services
- IoT - Internet of Things

1.4. References

Flutter Documentation

AWS IoT Documentation

Arduino IDE Documentation

2. System Design

2.1. Design Overview

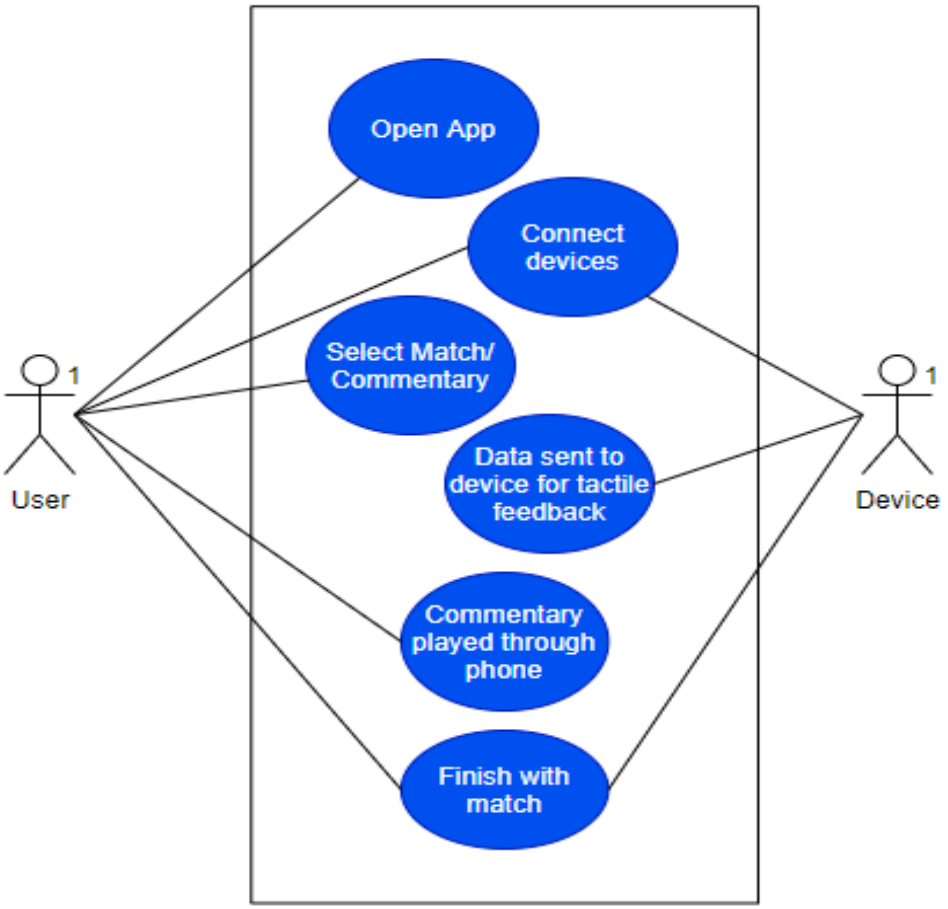
High-level overview of how the system is implemented, what tools, frameworks and languages are used etc.

The system will essentially be an NFC or Bluetooth enabled mobile app that allows visually impaired users to communicate with a touch-based tablet device. The app will have an accessible design and will be very easy to use and navigate. There will be a menu for selecting highlights. Once a match is selected there will be a remote that has buttons to allow the user to play, pause, rewind, toggle vibrations and toggle commentary as they wish. The app will send signals to our device via bluetooth. To build the app for both iOS and android, we have been using flutter, which uses the programming language

dart. To design mockups and other related things we use Adobe XD. To have a shared code-base we have been using git as our VCS and github. Some of the other tools we have been using are VS code as an ide/text editor, Android Studio or Xcode for simulation, discord for communication. We use Arduino IDE to work on the communication and testing with the device. We use AWS IOT to send the coordinates from the servers to the device, in order to make it move.

2.2. System Design Models

The user communicates with the app which in turns communicates with the device. The app is connected to an AWS server which contains the coordinates of the ball and the data on the app like the thumbnails and the names of the games. The app uses bluetooth to control the device so the user needs to enable bluetooth as well as wifi.



Use Case Name	Select Match/Commentary
Actors	User’s phone
Preconditions	User has the app opened, they’re connected to the device, and they have an internet connection
Descriptions	User has started the app and needs to select which match they would like to experience
Postconditions	User is able to experience the game using the device they’re connected to
Alternate flows and exceptions	User closes the app and does not select a game to be played

Use Case Name	Data Sent to Device for Tactile Feedback
Actors	Device
Preconditions	The app is open on the phone, a match has been selected, and playback has started
Descriptions	The game data is sent to the device and is outputting the sensory information to allow the user to experience the games
Postconditions	The user has “watched” the match using the information sent to the device
Alternate flows and exceptions	There is a transmission error or packet loss that results in interrupted transmission of the game

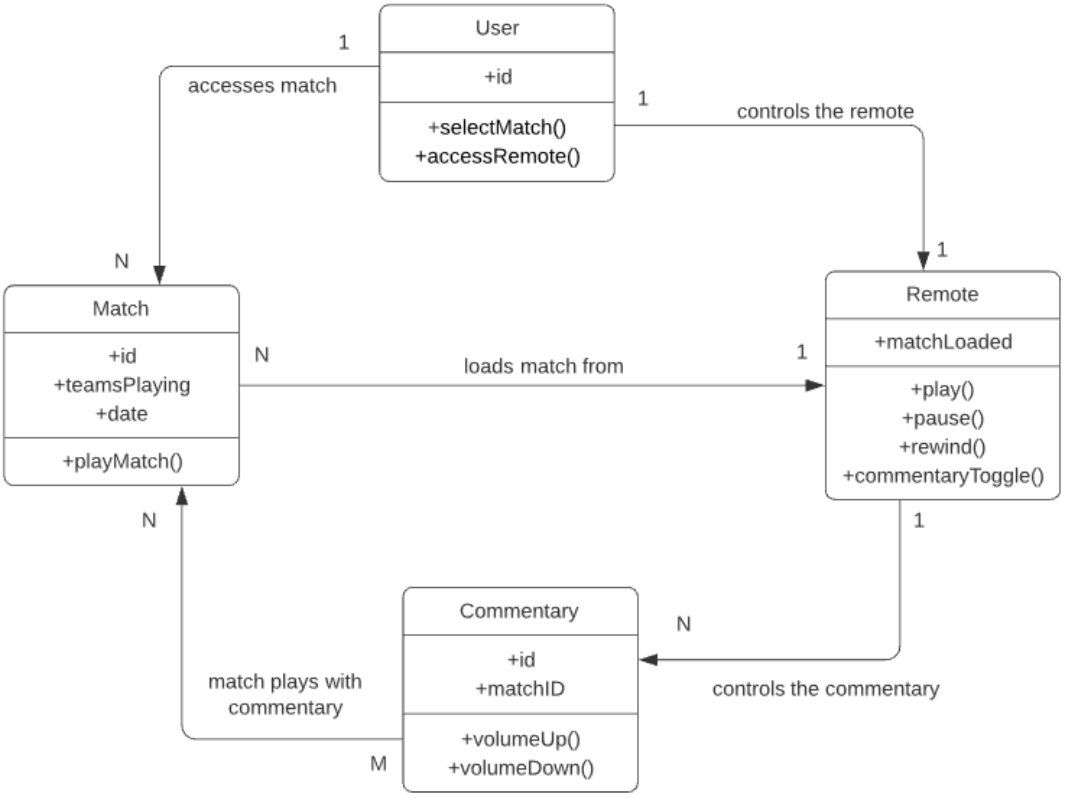
Use Case Name	Connect Devices
Actors	User’s phone and Device

Preconditions	Both devices are turned on and in range of each other
Descriptions	Uses Bluetooth to allow for connectivity between the phone which communicates the match information to the paired device
Postconditions	The information is then transmitted to the device and the user is able to experience the match
Alternate flows and exceptions	No exceptions

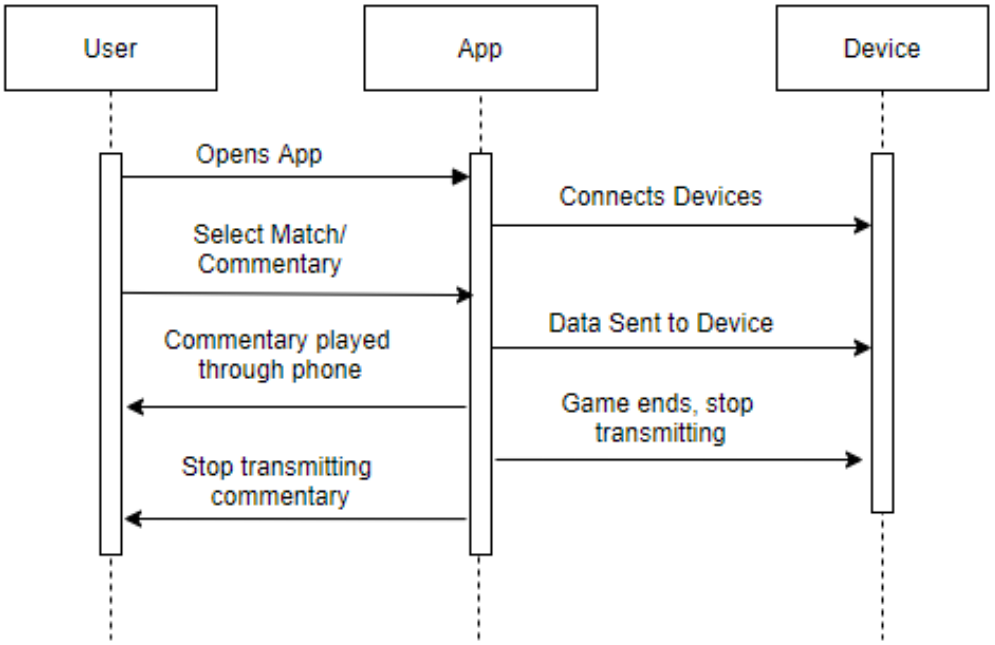
System Architecture

For the flutter app, we have a separate file for every screen, we invoke them all in the main file, keeping the main file clean and simple. For the backend we have different Arduino files that each do a different action with the device such as controlling it and connecting to the wifi, as well as receiving and sending messages to the server.

Class Diagrams



Sequence Diagrams



State Diagrams