Final Milestone

12/7/2022

CS Capstone I

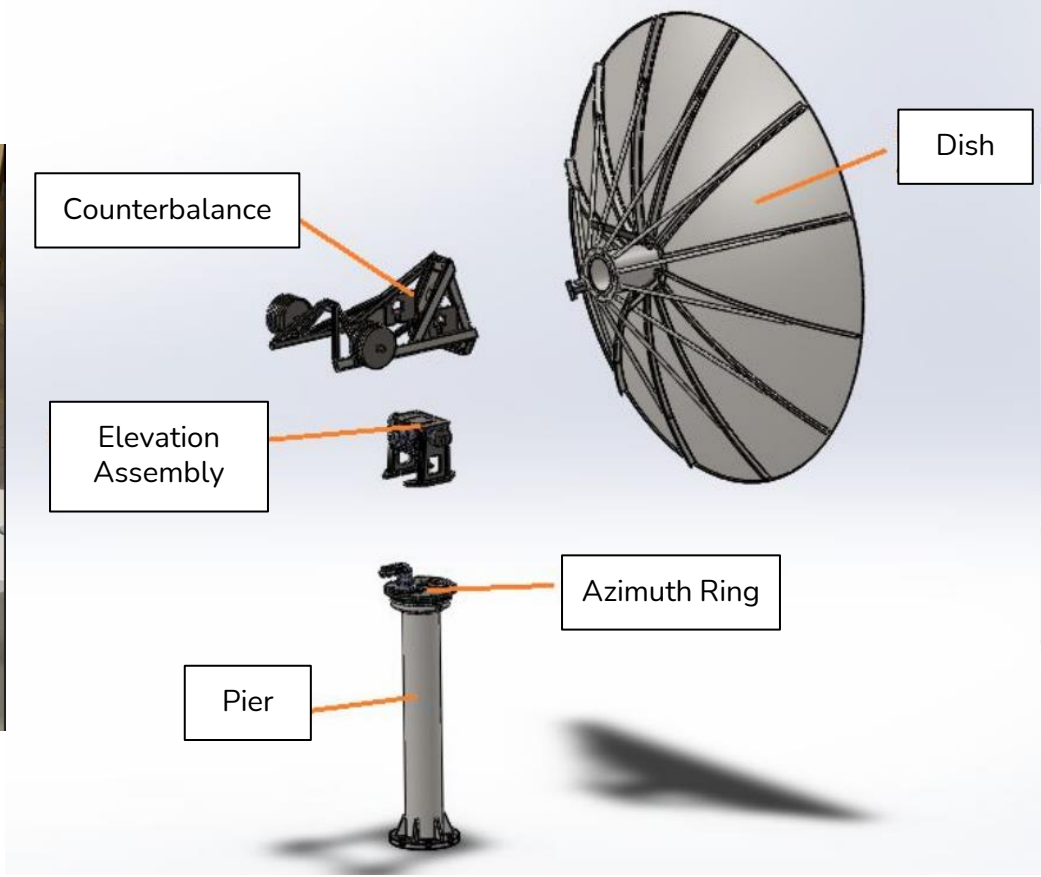
Table of Contents

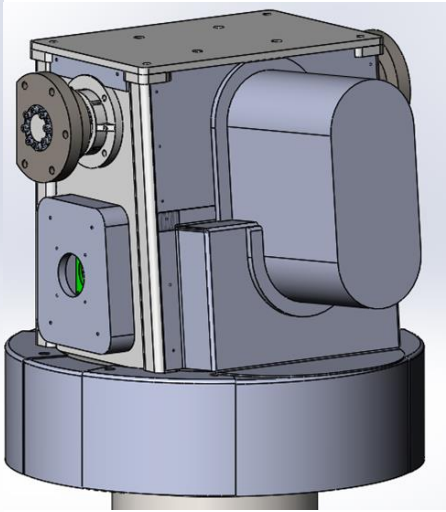
YCAS Radio Telescope Project Overview

What Is The Radio Telescope?

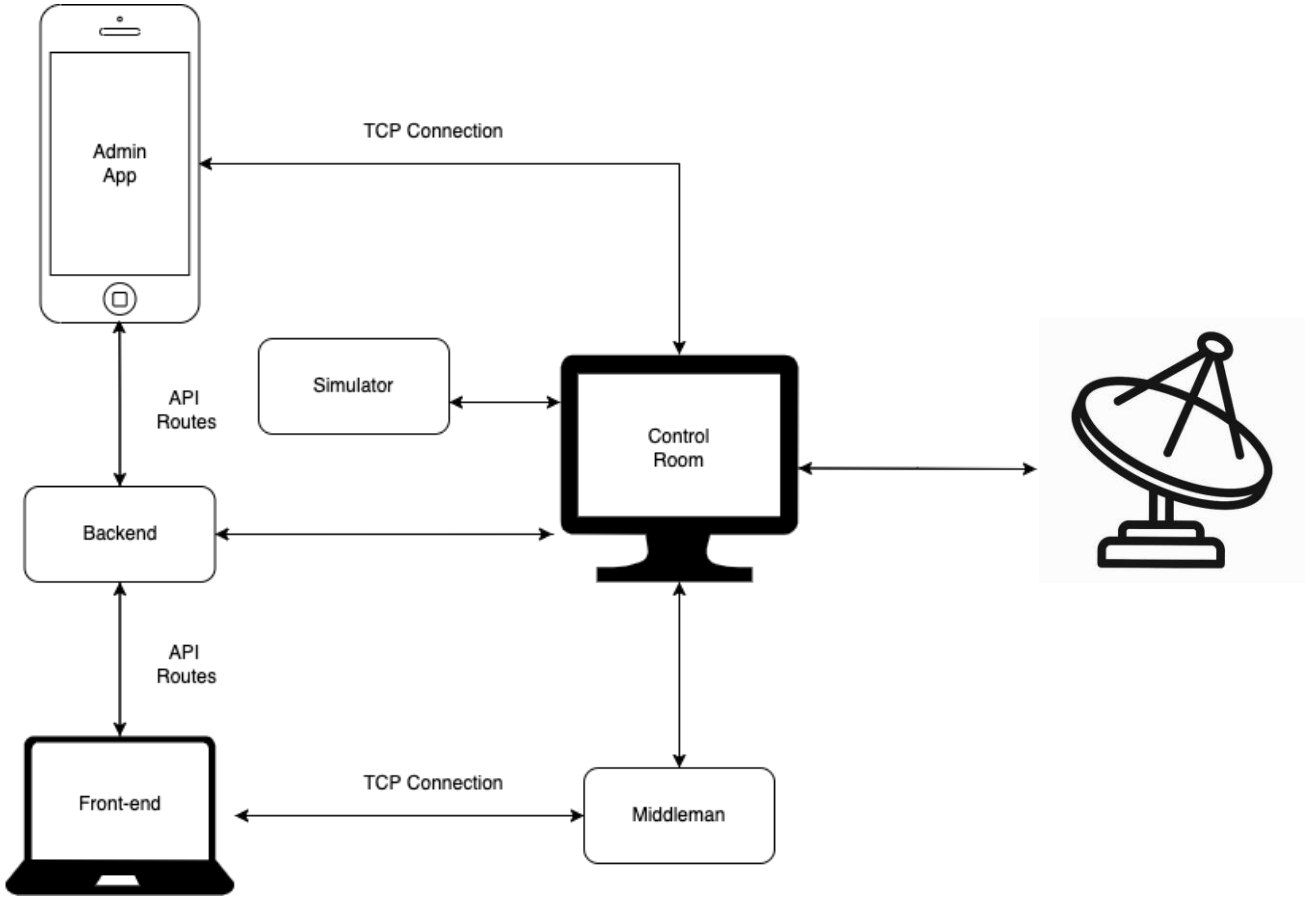
- 4.5m, remote controlled, auto-tracking, auto-locating Radio Telescope capable of scanning 1.42Ghz radio signals
- 5 years ago, YCP was contracted to build a radio telescope for the York County Astronomical Society to be placed in John C. Rudy County Park, York
- The telescope is for educational and amateur astronomy research
- This has been an ongoing project, delayed by COVID-19, worked on by over 80 students throughout 5 years
- The telescope is planned to be installed in the park

Major Components

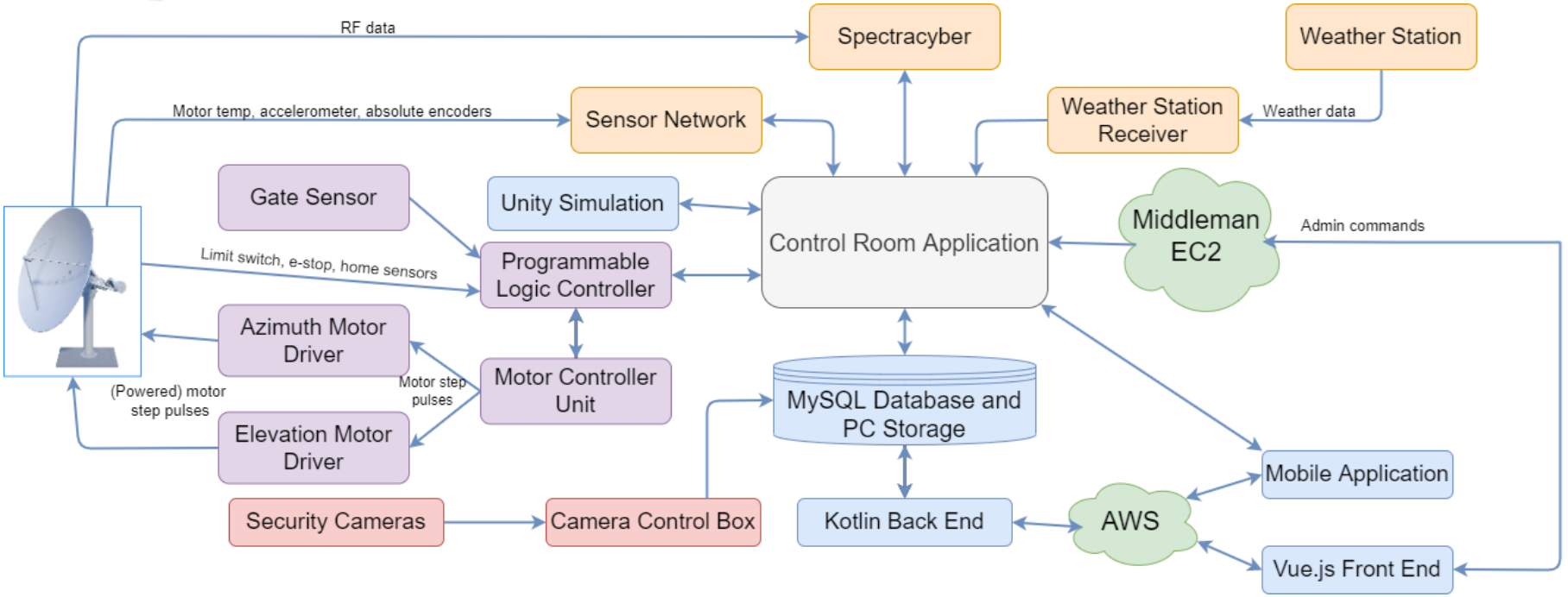




Telescope Overview



Detailed Telescope Overview



How will this be used?

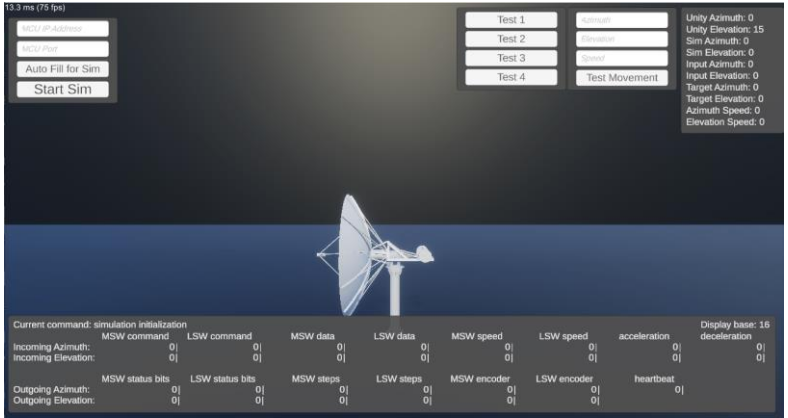
- YCAS will have a website that the general public can create accounts and set appointments
- YCAS admins can use the control room software to monitor and move the telescope
- YCAS admins will have access to the mobile app allowing them to control the telescope remotely
- YCAS will have access to a VR game version of the park and telescope that can be shown off to students

The VR Team

Derek Herr, Jordan King

Purpose of the VR Team

- Simulated environment similar to the end product of the telescope
  - Can be shown for educational purposes and to potential business partners
- Provides a portable service that simulates the functionality of the telescope when access isn't available

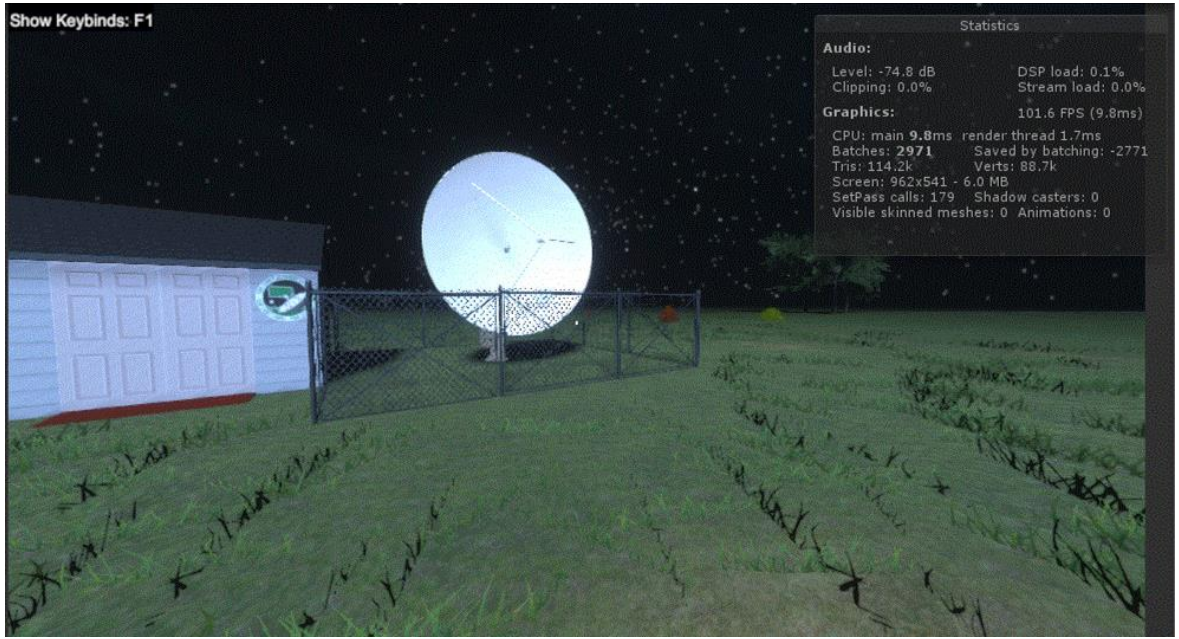


What We Accomplished

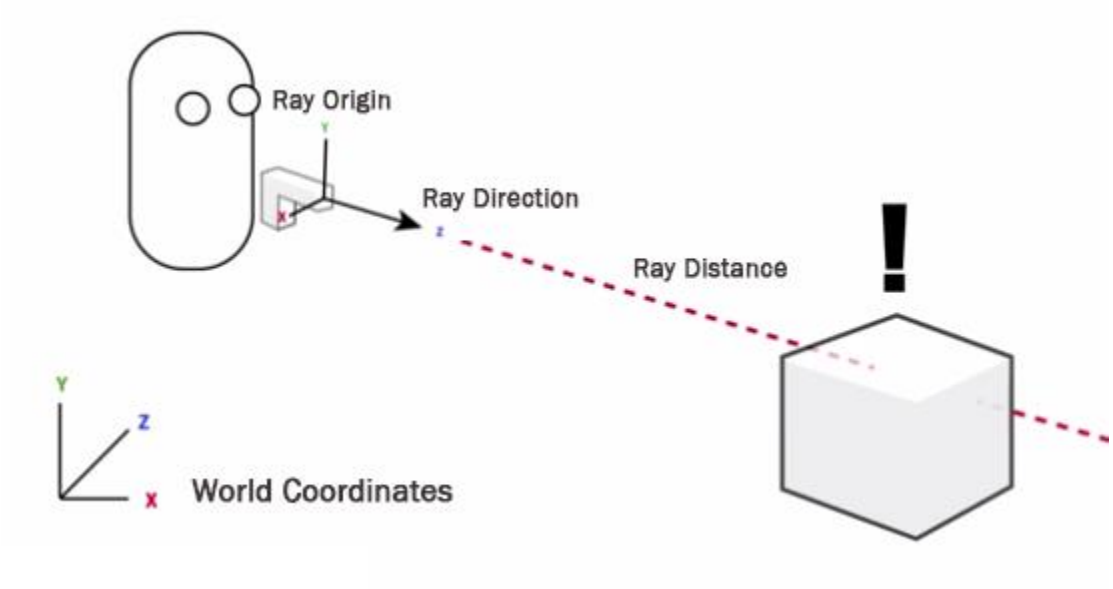
- Working star positioning system that allows user to see star alignment at a specific time
- A modified star interaction system that allows the inclusion of multiple data objects per interaction point
- A completely new cinematic that helps explain the purpose of the telescope
- Full functionality of the console that can change the date/time in the simulation
- Full functionality of a open/close gate system

Gate open/close interaction

- The player is able to open/close the gate

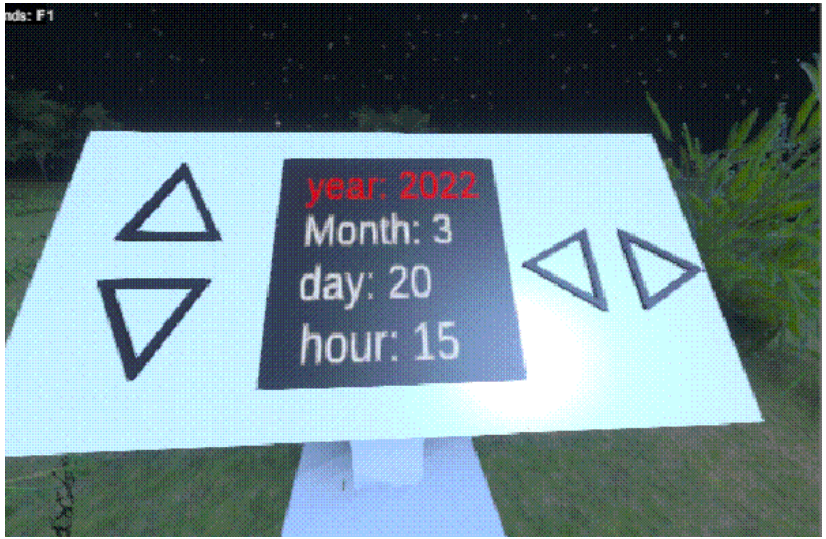


Ray casting


Ray Origin
Ray Direction
Ray Distance
!
Y
Z
X
World Coordinates

Star Console interaction

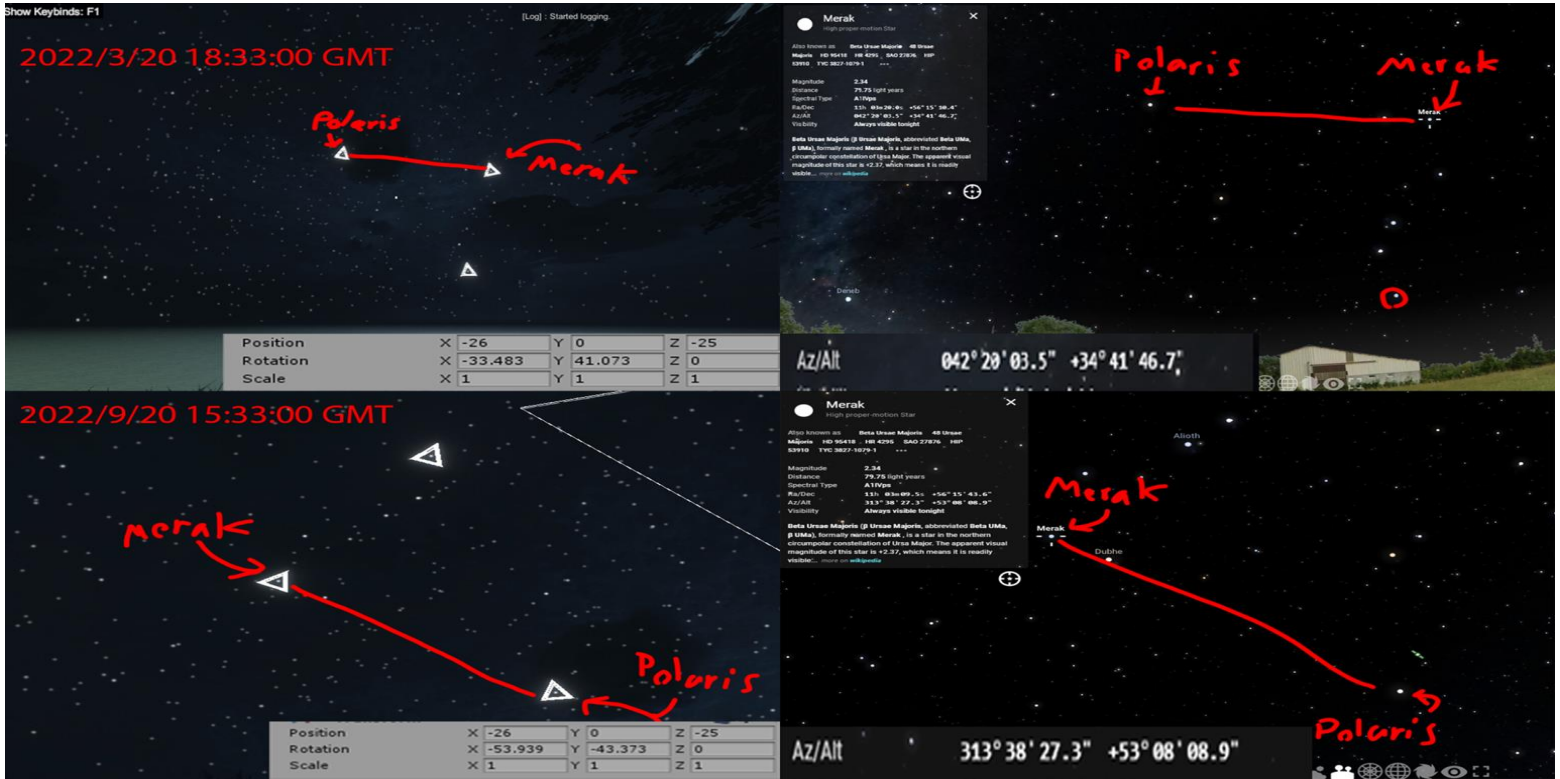
- Lets the player change the date/time of the simulation
  - One set of arrows increments/decrements
  - The other switches between date units
- Displays the date/time and shows what unit can be changed



Accurate Star System

Star Positioning

- Is controlled by the control panel
- Accurate to 3 degrees within 30 years from 2022

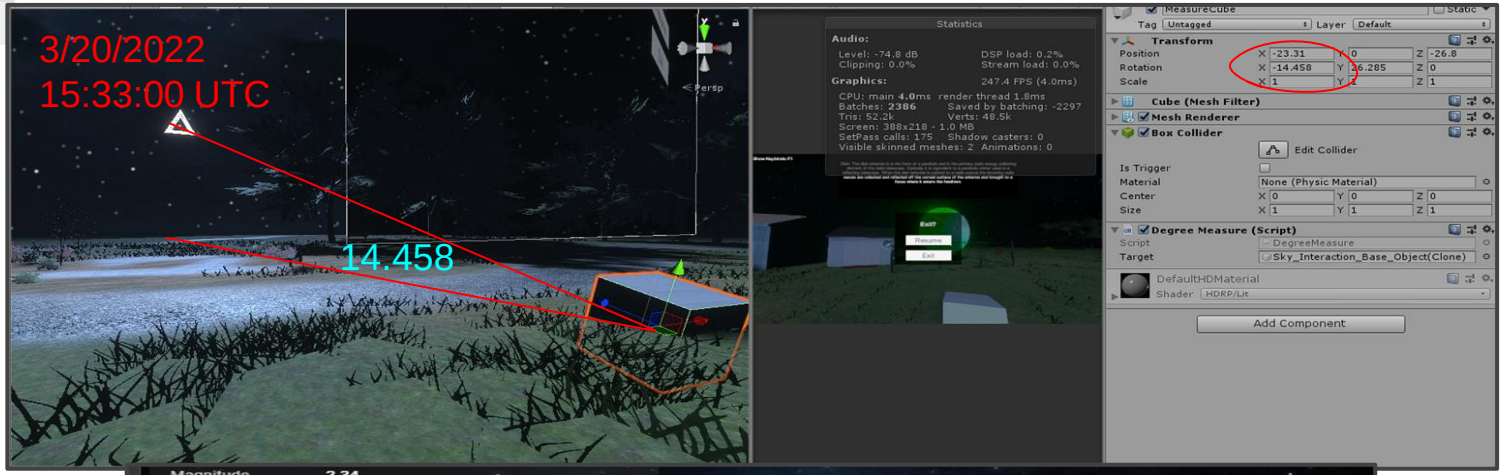


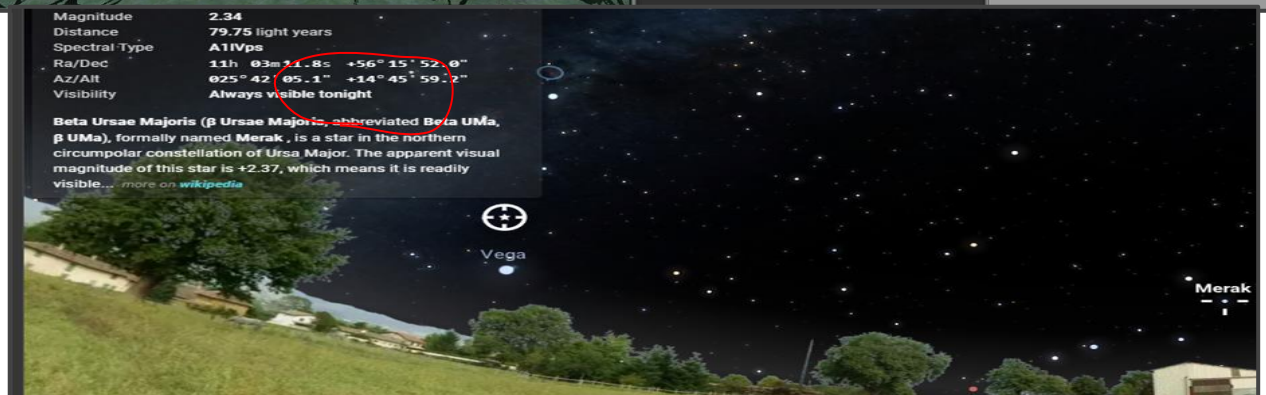
Star Positioning Calc

- Set Reference point (Vernal Equinox of 2022)
- Calculate difference in time
- Normalize to only one year of calc
- Calculate Leap Year offset
- Multiply difference by angle of rotation per minute
- Rotate Star System

Star Positioning Testing

- Is controlled by the control panel
- Accurate to 3 degrees within 30 years from 2022





Star Positioning Usage

- User enters a date with the control panel
- User confirms date on the control panel
- Star positioning script is called which calculates where the stars should be and rotates the system accordingly

Star Interaction System Updates

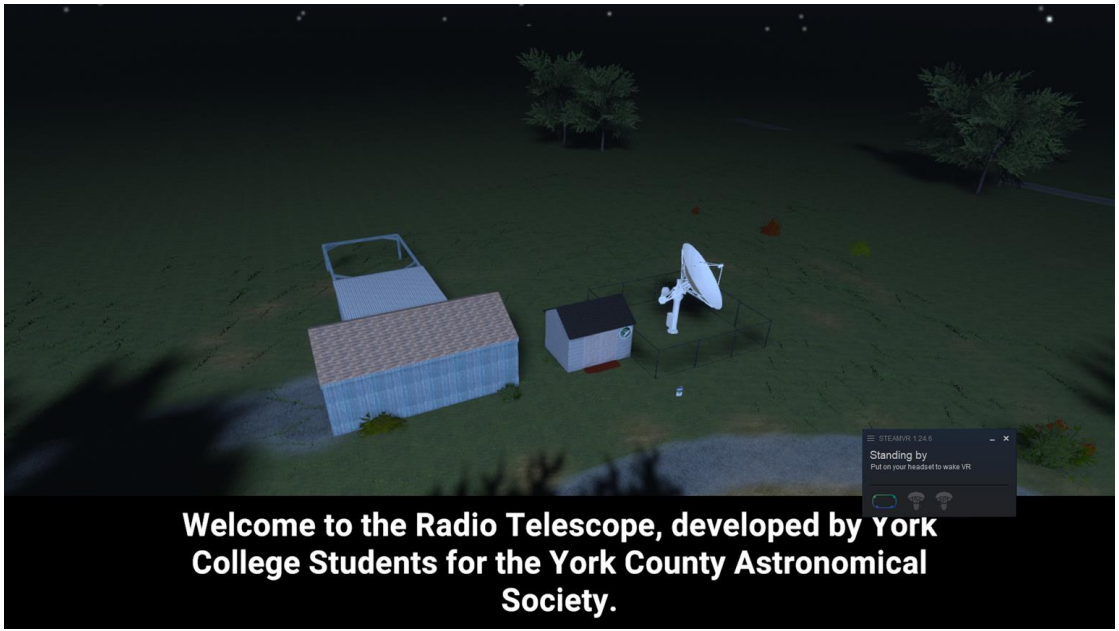
- User interacts with sky objects to see telescope data
- Mostly operational from last semester
- Needed to be updated to handle multiple data objects on one data point
- .CSV format changes
- Restructuring of the entire data model from last semester

Star Interaction System Old

Star Interaction System New

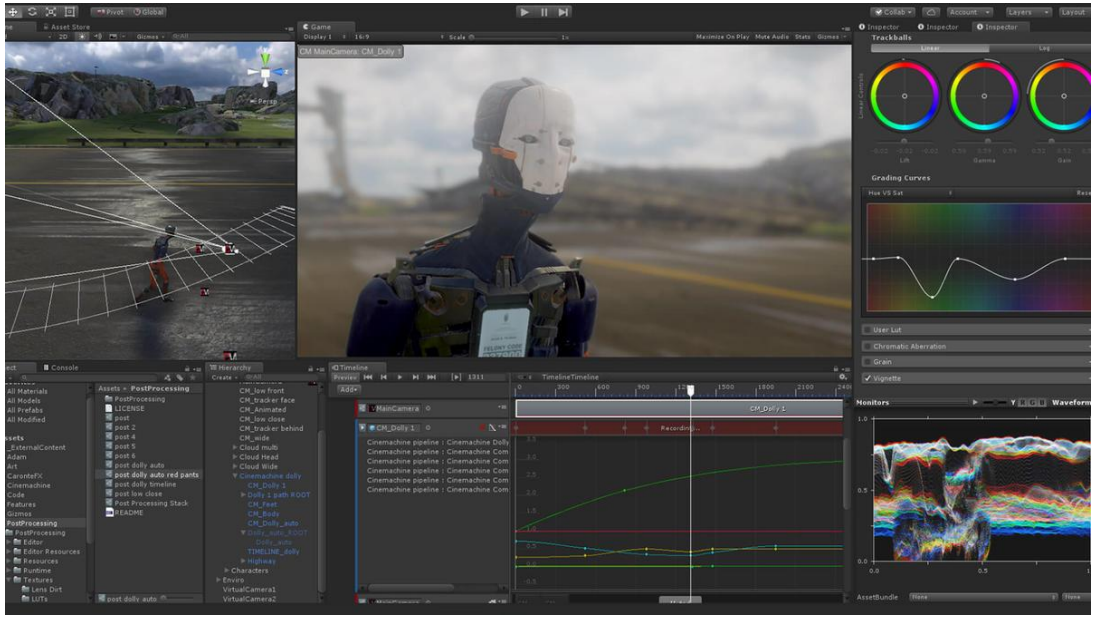
Cinematic

- A short cinematic that would explain the functions and development for the telescope
- Started development late in semester
- Structure of cinematic built
- Fully operational on MK, needs to be added on VR



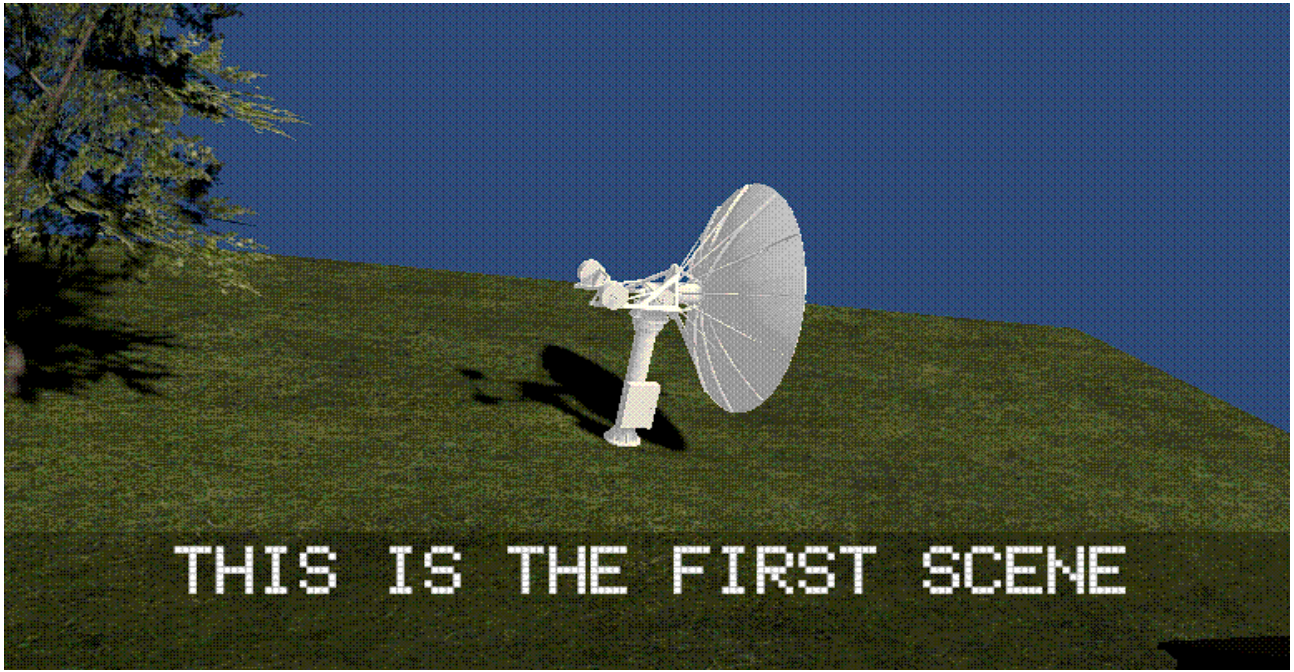
Unity Cinemachine

- Chosen as the framework for the cinematic
- Is Unity's main framework for cinematics



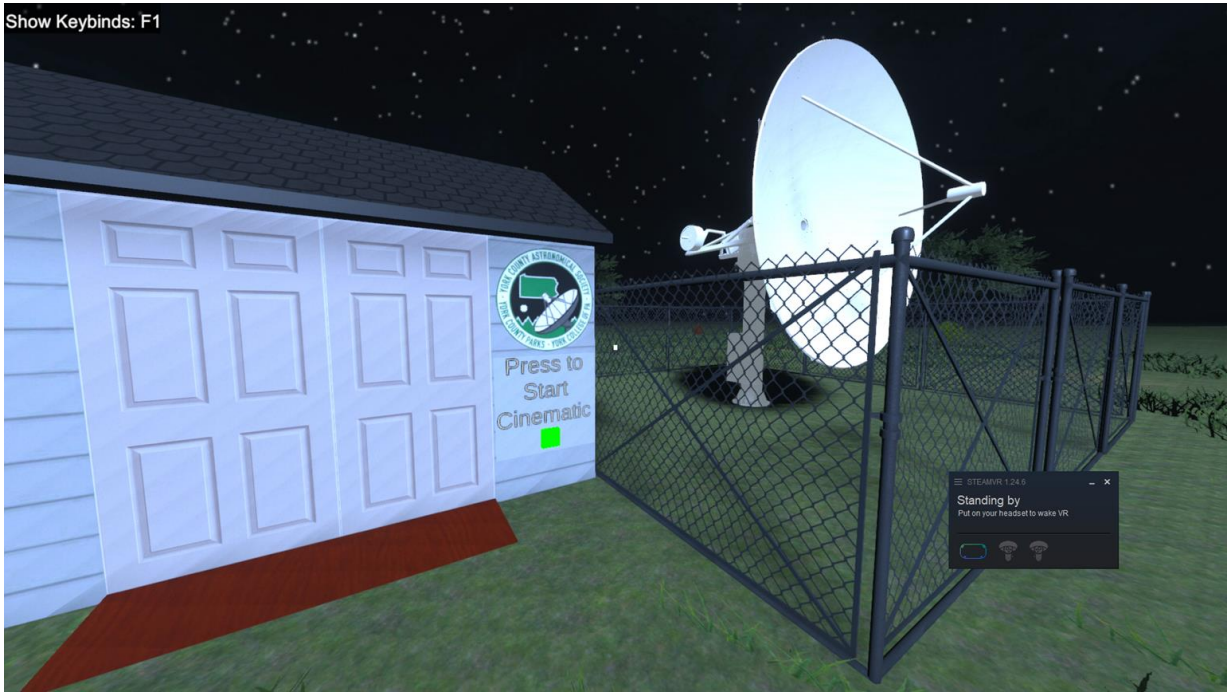
Cinematic Testing and Development

- Small demo scene created to test capabilities of cinemachine
- Perfect on mouse and keyboard
- Vr requires modifications and longer development



Cinematic Usage

- User hits cinematic button in main scene
- Cinematic plays
- User returns to scene on completion



Demo

The Mobile App Team

Kevin Tanzosh

Purpose of the Mobile App

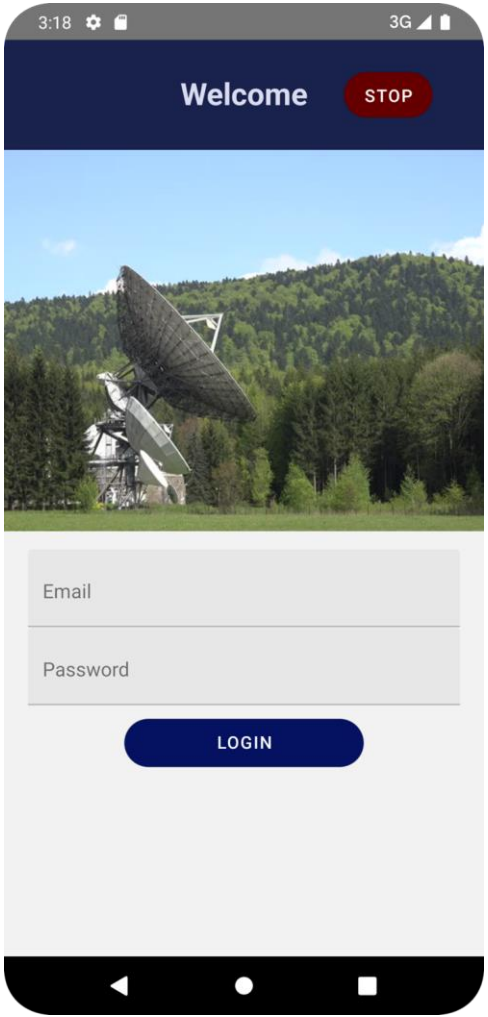
- The purpose of the mobile application is for the administrators to be able to remotely
  - Monitor the telescope’s status in real-time
  - Operate it remotely
  - Receive notifications on telescope health
  - Check the health of the telescope from sensors and override them
  - View local weather conditions from a weather station co-located with the telescope

What We Started With

- Functioning app that mostly relied on encrypted TCP for communication
- API system that was set up but not entirely functional
- Various updates needed for some older functionalities
- Various updates to dependencies and React Native

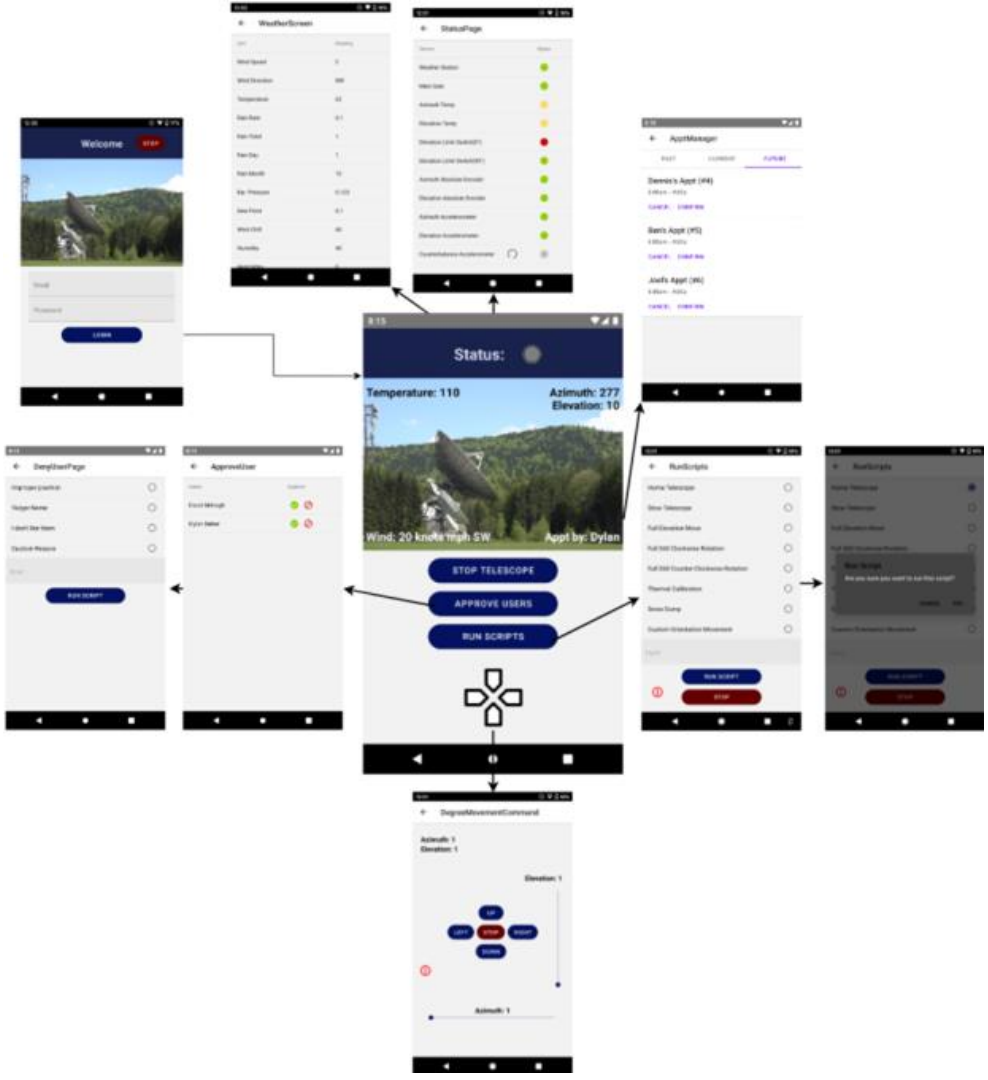
Login

- Admins will be given a login to be used when entering the app
- Their information will be stored for future use
- This login will create a token to authorize API calls, will be stored with previous information
- A stop button is added for safety
- After login the admin will be directed to the home page



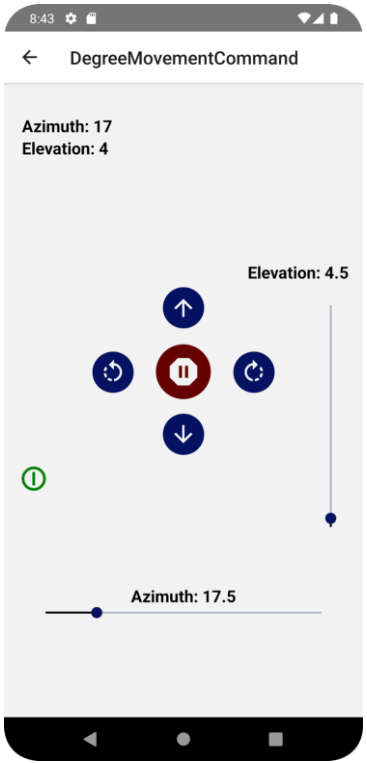
Home Page

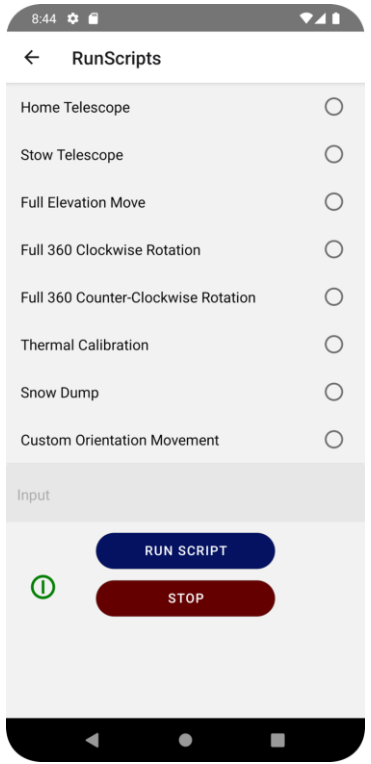
- Main page that leads to all other pages
- Weather, current azimuth and elevation location are shown here, gathered from TCP messaging



Movement Pages

- Degree Movement Page allows for precise movements to be made
- Run Scripts allows for control room scripts to be run
- Both allow for MCU bit to be reset if the icon is red.
- All information is sent or received through encrypted TCP Messaging





TCP Messaging

- TCP Encryption encrypts a message in the form of
  - <Version> | <Type> | <Name <values>> | UTC TIME
- Receives in the form of
  - <Version> | <ENCRYPTED COMMAND>
- EncrytoJS is used for encryption and decryption
- This is used to safely communicate with the control room for movement and other actions to take place, with an encrypted return message sent back to the mobile app

The Control Room Team

Liam Bradley, Josh Snyder, Robby Weaver

Purpose of the Control Room

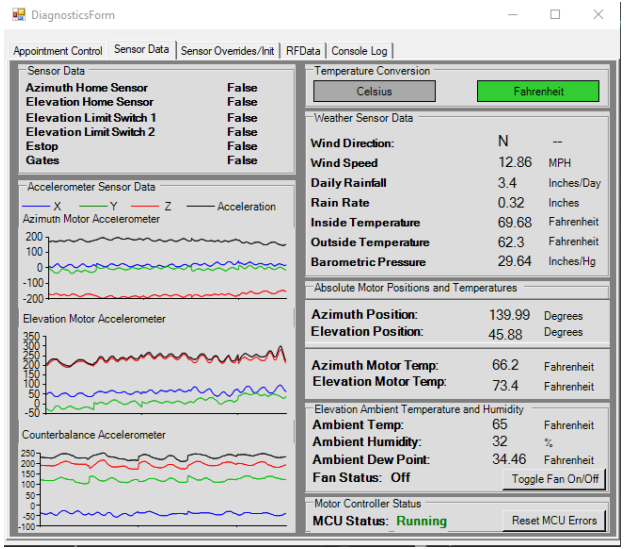
- Critical component to the operation of the telescope
  - Indirectly or directly connects to each component in some way
- Directly controls the Radio Telescope hardware
- Monitors sensors and health of the telescope
  - Save sensor data to the database for retrieval later on
  - Notifies administrators if something goes wrong
- Queues up appointments that users add through the website
- Displays data from different types of scans
  - Sends data to users after their appointment has complete

What We Started With

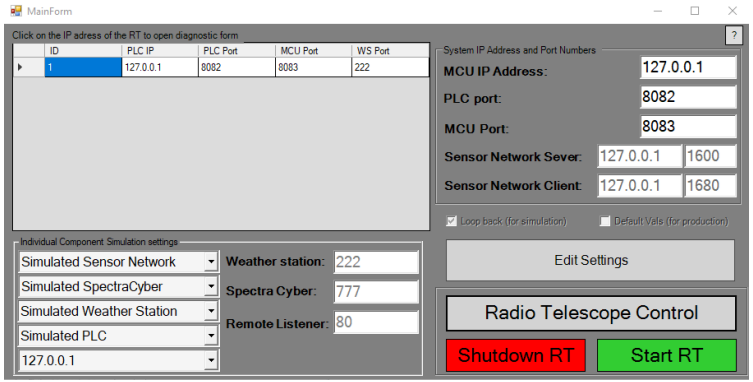
- The telescope was in a working state at the end of Spring 2022
- Various hardware issues needed addressed:
  - Unreliable data from accelerometers and encoders
  - “Encoders too far apart” bug
- Various software issues needed addressed:
  - Handling overdue appointments
  - Slow encoder updates & graph skipping on accelerometer data
- Various features needed implemented:
  - Limit switch overrides
  - Disconnect → Stow routine
  - Adding users/appointments from the Control Room Application

Control Room Overview

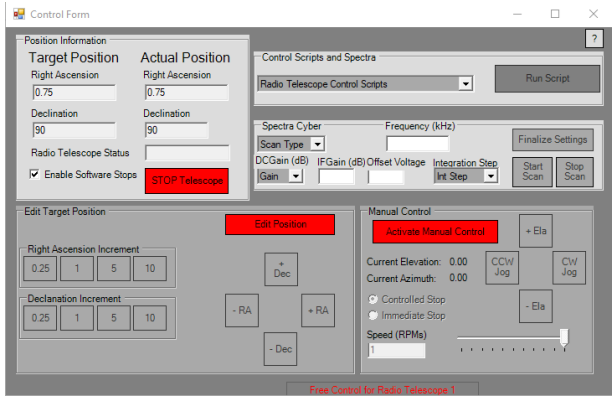
- “Brains” of the telescope
- 4 Main Forms
  - Main Form
  - Diagnostics Form
  - Control Form
  - Appointment Form



Diagnostics Form



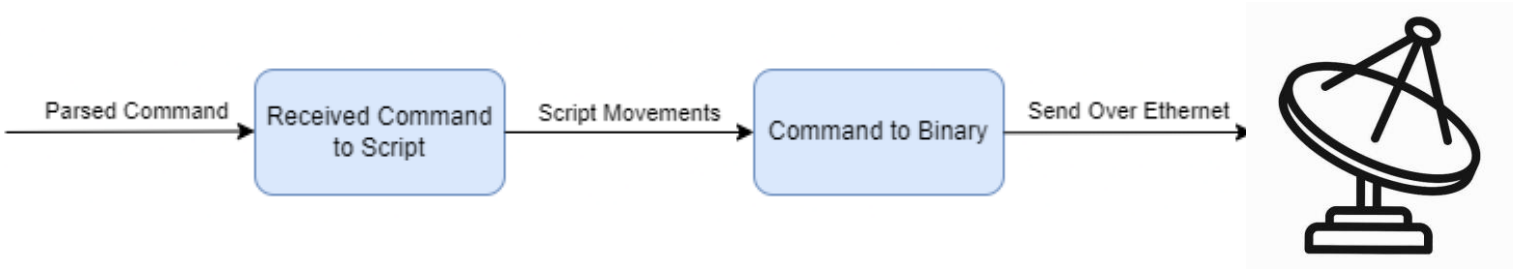
Main Form



Control Form

Commanding the Telescope

- Commands are interpreted as scripts
- Scripts are a series of movements
- Different movements types to send to MCU
  - Absolute movements
  - Relative movements
  - Jog movements
- Binary instructions are assembled for the move and sent via ethernet



Hardware

MCU:

- Controls the elevation and azimuth motors
- Capable of relative moves, homing, and jogs
- Configured via Modbus TCP Commands

PLC:

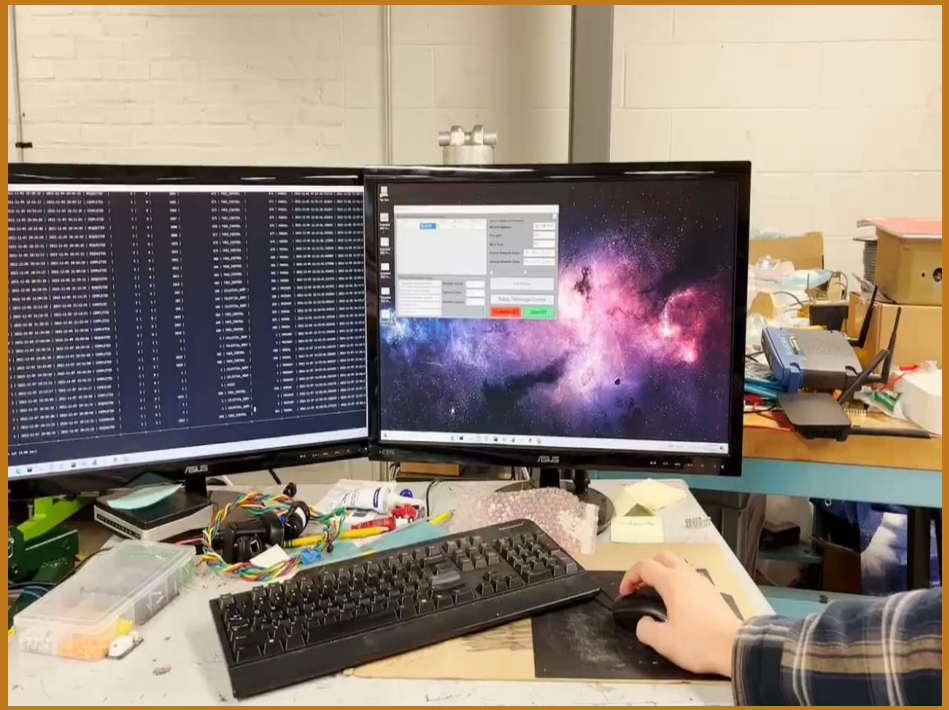
- Communicates via Modbus TCP Commands between Control Room and MCU
- Monitors various system flags and prevents / allows / initiates movements in these scenarios (limit switch overrides, Control Room Disconnect, EStops, etc.)

ESS:

- Collects acceleration / encoder data to the Control Room
- Control Room processes data and makes decisions based on data
    - Ex: Encoders too far apart, too windy, etc.

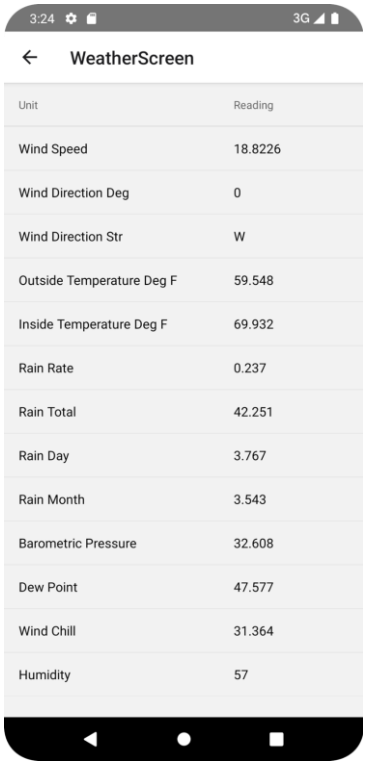
Mobile App/Movement Demo

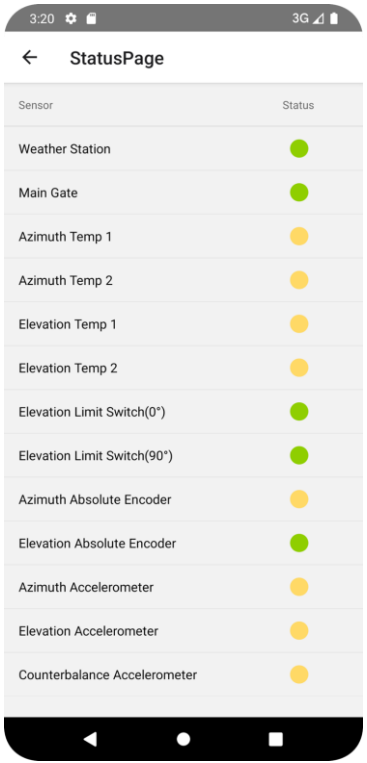
PLC Stow Routine



Weather Data/Sensor Page

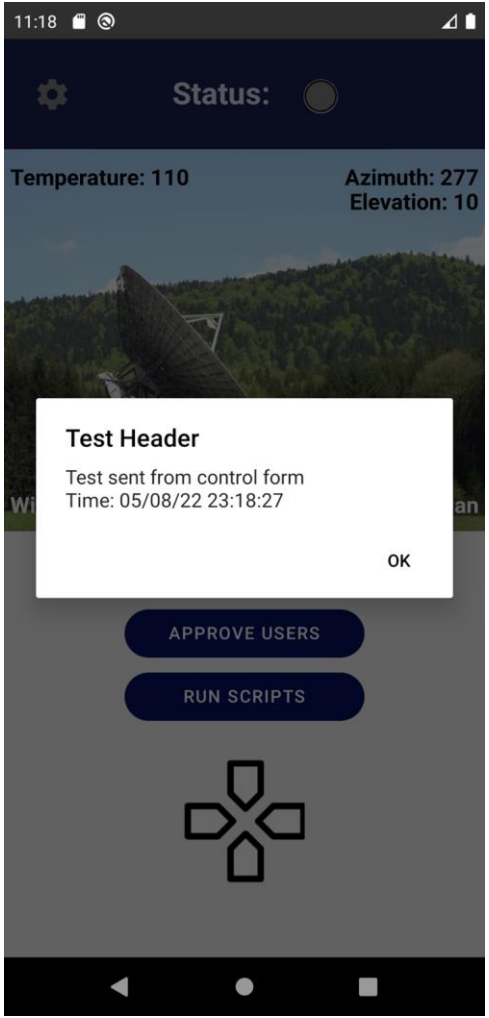
- Our Two Pages that currently use API Requests to get the most recent info on the Weather and the state of the Sensors
- Resend API call and update on click
- Set to load when data is still initial values





Firebase/ Push Notifications

- Push notifications are sent from the Control Room via Google's Firebase Notification Service APIs
- Push notifications are sent when:
  - Sensors override
  - Sensors become critical from API call
  - Disconnect from ESS
  - Gate is opened



API Requests

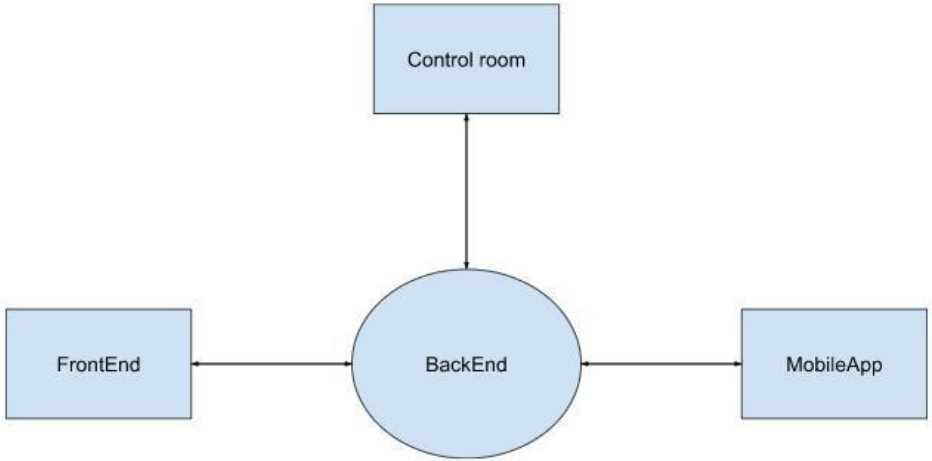
- In conjunction with the back end team, a request is sent to them to gather the necessary information for both weather data and sensors
- Parsed and properly displays the necessary information from that request
- Allows for important information to be seen anywhere

The Backend Team

Jordan King

Purpose of the Backend

- Supports systems such as control room, frontend, and mobile app.
    - Providing JSONs that contain data through API requests
- Stores data within the repository

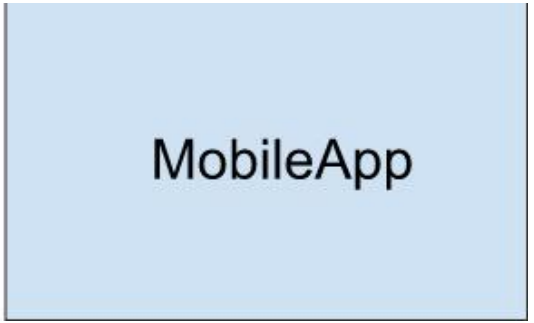


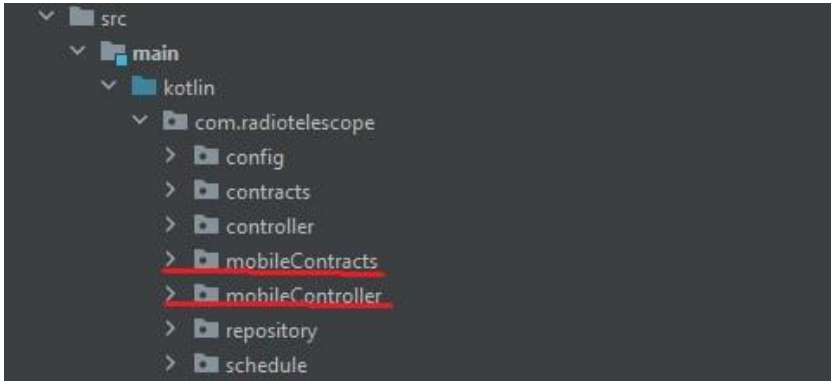
What we started with

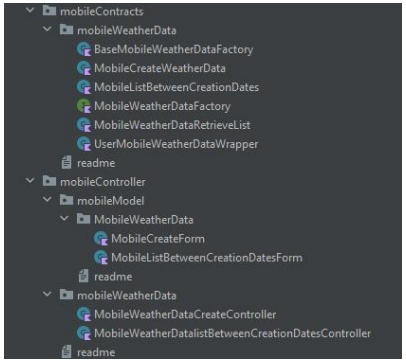
- Functional backend with working APIs for the Front-end
- The Database for the backend was fully functional with the Front-end APIs
- The security of the product was very poor with only an encryption process for the password

Original Design

- The API calls for the MobileApp were going to be separated from the FrontEnd.
- Was a terrible idea since nothing would be changed about the APIs

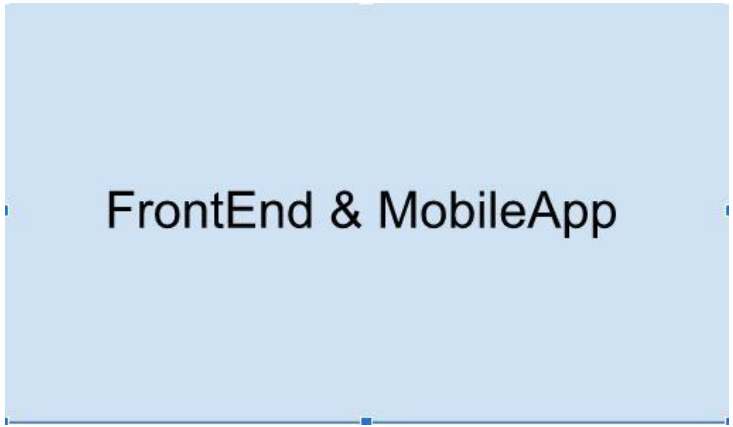


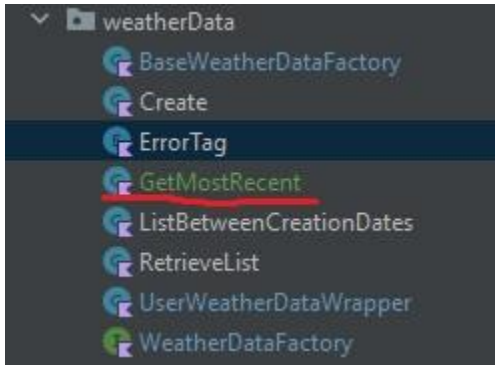




New Design

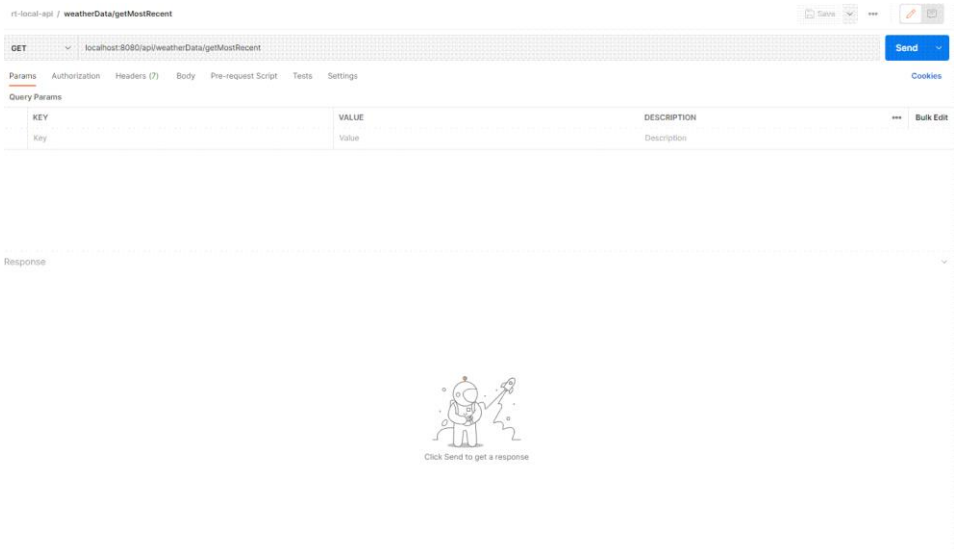
- Therefore the team decided to scrap the effort to and adopt a better plan of fully utilizing the API calls from FrontEnd into the MobileApp





mobile WeatherDataMostRecent support

- Lets the mobile app request the most recent weather data from the backend

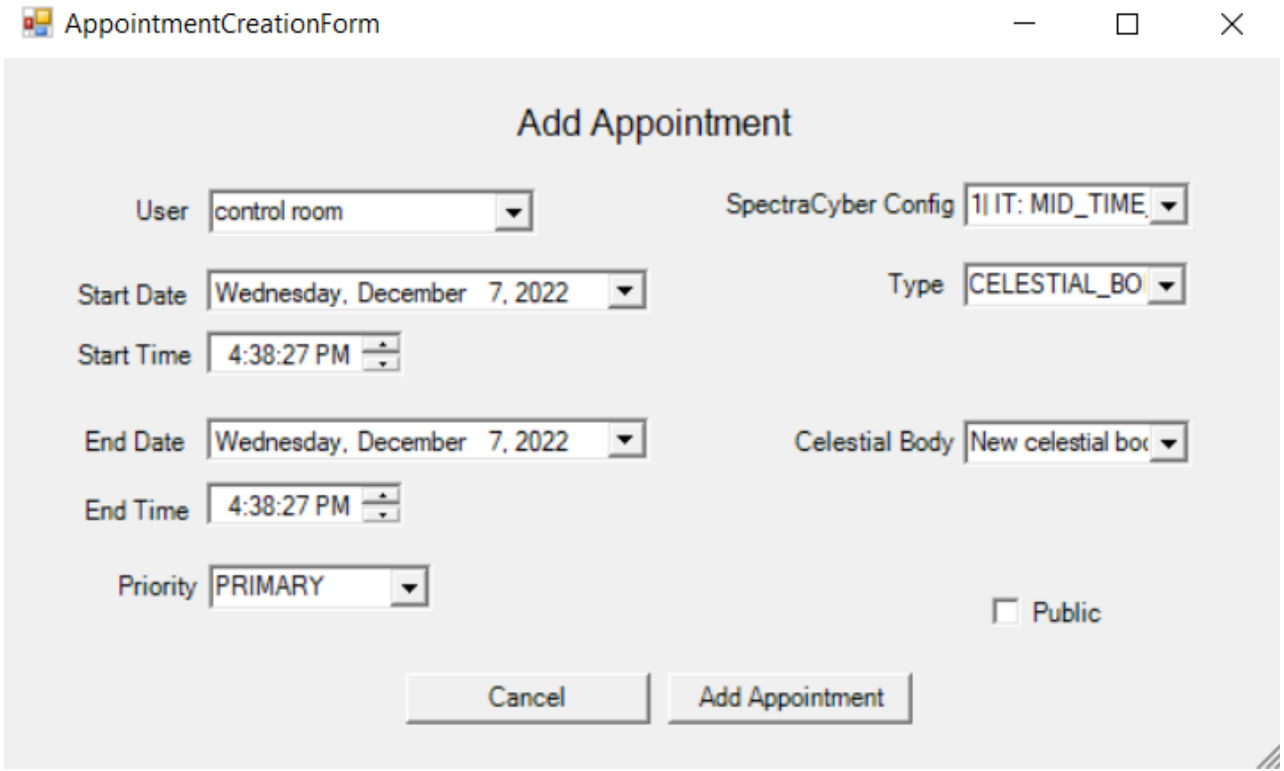


	id	wind_speed	wind_direction_deg	wind_direction_str	outside_temperature_degF	inside_temperature_degF
395	395	15.0495	0	NNE	60.619	68.97
396	396	26.1178	0	NE	61.512	70.002
397	397	18.1628	0	N	48.639	69.644
398	398	10.1534	0	NNE	57.495	69.154
399	399	23.8986	0	NE	58.68	69.878
400	400	21.4645	0	ENE	56.14	69.828
401	401	29.504	0	E	56.011	71.58
402	402	17.7446	0	ESE	59.273	70.154
403	403	16.6433	0	SE	61.766	69.721
404	404	21.8459	0	SSE	62.045	69.916
405	405	18.6596	0	S	56.629	69.067
406	406	16.2805	0	SSW	52.755	69.149
407	407	19.309	0	SW	53.925	69.8
408	408	24.6213	0	WSW	62.557	68.373
409	409	23.7408	0	W	48.479	72.027
410	410	25.1721	0	WNW	60.882	67.082
411	411	19.7524	0	NW	47.81	70.074
412	412	24.5829	0	N	58.851	70.869
413	413	19.0543	0	NNE	67.576	69.221
414	414	17.2153	0	NE	55.515	67.974
415	415	20.4365	0	ENE	56.051	69.313
416	416	23.5522	0	E	67.889	70.931

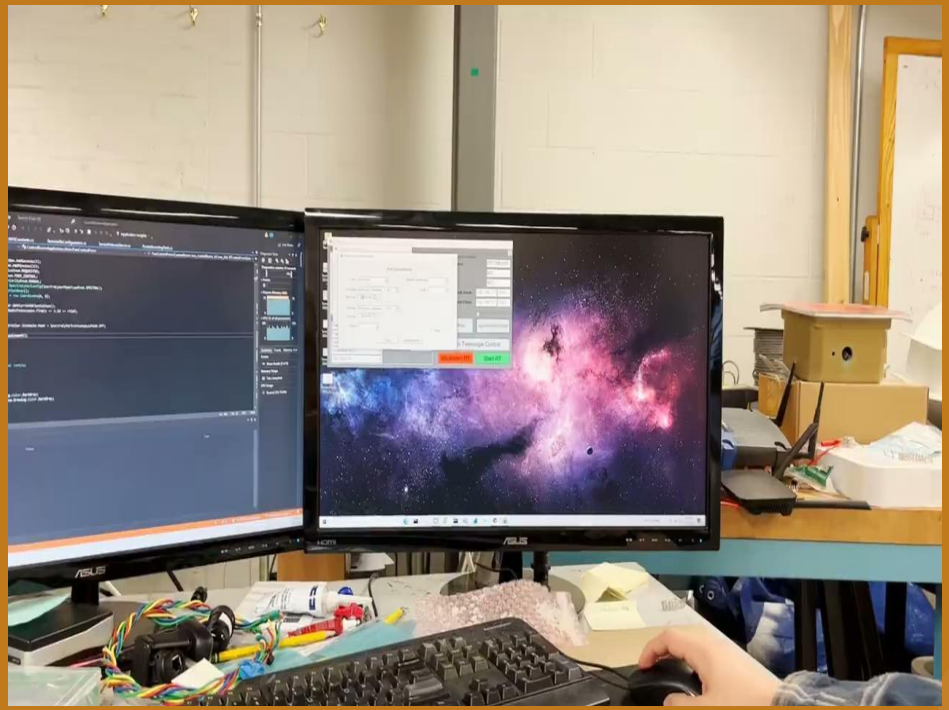
The Life of an Appointment

A story of love, war, tragedy, and perseverance

Appointment Creation



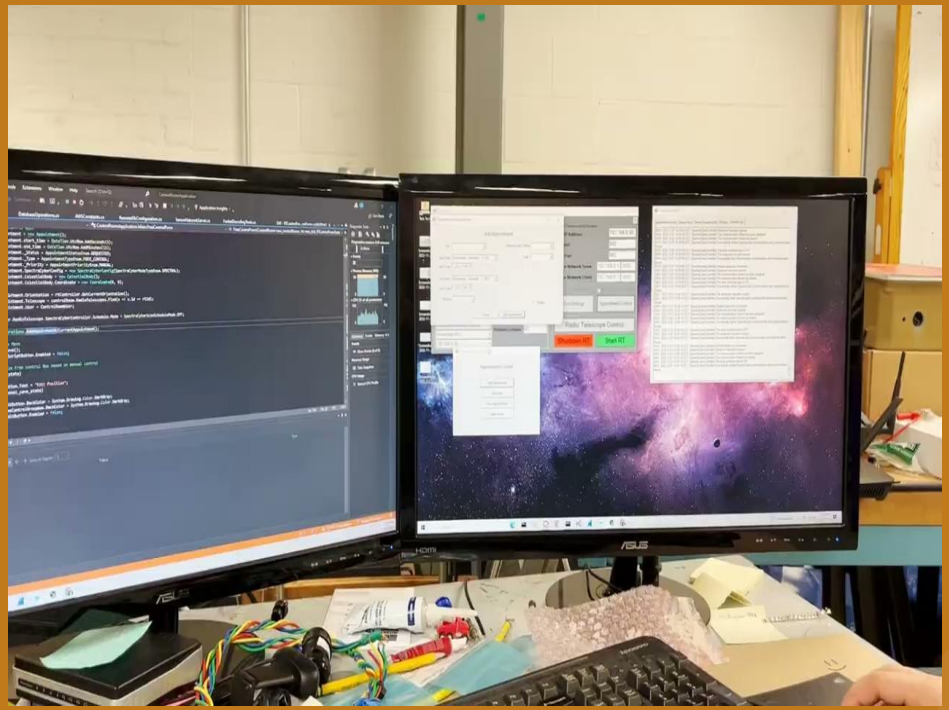
On-time Appointment



Overdue Appointments

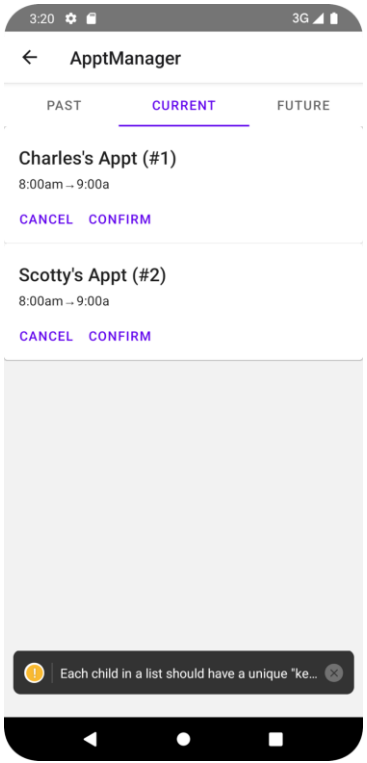
- Appointments are handled one of three ways depending on their start & end times.
- Case 1: Start & End Times have not passed
  - Appointment runs as usual
- Case 2: Start Time has passed; End Time has not passed
  - Appointment starts and will end at the end time (might end early)
  - User is notified
- Case 3: Start & End Times have passed
  - Appointment is cancelled
  - User is notified

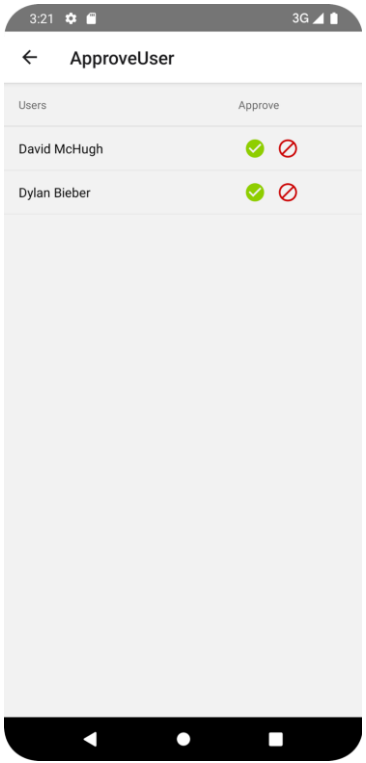
Overdue Appointment



Appointment and Approval Page

- Base functionally is present to display all past, present, and future appointments, or users
- Future work to check their results, approve or deny appointments or ban the users.





Future Work!

Future Work-VR

- Improve the console
  - Improve the representation of the console (add highlight to the buttons when hovered over)
  - Improve the scripting of the console(regarding decrementing/incrementing date)
- Make star positioning more accurate
  - Still inaccurate to extreme distances from reference
- Add the cinematic to the VR
  - Needs accommodations for virtual cameras
  - Polishing and adding animations
- Create the Sun and Moon

Future Work - MA

- Continued work to switch from TCP to API calls
- Appointments Overhaul
- iOS Visual Updates
- Store admin Requests
- Continued updates to React native and dependencies
- Live photos of Radio Telescope

Future Work - Control Room

- Before Installation:
  - Updating the Number of Steps in Disconnect → Stow Routine
  - Email Notification for Overdue Appointments
  - ESS Watchdog and Ethernet Connection issues
  - Spectral Scan to CSV
  - Verify Overall Functionality
- After Installation:
  - Remote Desktop Setup
  - Security Camera Installation

Future work - Backend

- Create support for other APIs within the mobile app
  - Get upcoming appointments
  - Get past appointments (Last month, who what when, png from results)
  - Get current appointments
  - Put approve/deny/deny and ban upcoming appointments
  - Get users awaiting approval
  - Put approve users
  - Put sensor override
- Security of the BackEnd could be improved with HTTPS

Questions?