

# Real-Time Map Data Visualizer

DESIGN DOCUMENT

Team 21

Dr. Joseph Zambreno

Dr. Joseph Zambreno

Scott Frank, Ami Ikanovic, Isaac Littler, Benjamin Kelly, Zahydee

Machado, Elizabeth Nelson, Parth Padmanabhan

sdmay22-21@iastate.edu

<https://sdmay22-21.sd.ece.iastate.edu/index.html>

Revised: Dec 6, 2021

# Executive Summary

## Development Standards & Practices Used

- Microsoft Hololens 2 Development Standards
- Unity Engine Development Standards
- IEEE 802.11 Standard for Information Technology - Telecommunications and Information Exchange between systems

## Summary of Requirements

- Track physical objects to map the (virtual) globe's location
- Scale and/or rotate the 3D globe at the user's command
- Globe remains stationary in the room independent from the user's location
- 6'x6' unobstructed space (constraint)
- Use of Microsoft's Hololens 2 (constraint)
- Own API layer to standardize incoming data streams
- Built using Unity Engine
- Accepts three (3) different types of geographical data streams as input
- Maintains stable sixty (60) fps (constraint)
- One (1) hour of use on the device's onboard battery
- "Plug & Play"/"Pick up and go" ease of usability
- Maintains 720p resolution

## Applicable Courses from Iowa State University Curriculum

- COM S 227: Object-Oriented Programming
- COM S 228: Introduction to Data Structures
- COM S 309: Software Development Practices
- COM S 336: Introduction to Computer Graphics
- COM S 437: Computer Game and Media Programming
- SE 317: Introduction to Software Testing
- SE 319: Construction of User Interfaces
- SE 329: Software Project Management
- SE 339: Software Architecture and Design

## New Skills/Knowledge acquired not taught in courses

- Flask
- GraphQL
- Unity Game Engine
- Mixed Reality Toolkit
- AR development

# Table of Contents

<b>List of Figures/Tables/Symbols/Definitions</b>	<b>6</b>
x.1 Figures	6
x.2 Tables	6
x.3 Definitions	6
<b>1 Team</b>	<b>7</b>
1.1 Team Members	7
1.2 Required Skill Sets for Your Project	7
1.3 Skill Sets covered by the Team	7
1.4 Project Management Style Adopted by the team	7
1.5 Initial Project Management Roles CHANGE	7
<b>2 Introduction</b>	<b>8</b>
2.1 Problem Statement	8
2.2 Requirements & Constraints	8
2.3 Engineering Standards	8
2.4 Intended Users and Uses	9
<b>3 Project Plan</b>	<b>10</b>
3.1 Project Management/Tracking Procedures	10
3.2 Task Decomposition	10
Figure 1 - Task Decomposition Diagram	10
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria	11
3.4 Project Timeline/Schedule	12
Figure 2 - Gantt Chart	13
3.5 Risks And Risk Management/Mitigation	14
3.6 Personnel Effort Requirements	15
Table 1 - Personal Efforts	15
3.7 Other Resource Requirements	15
<b>4 Design</b>	<b>16</b>

4.1 Design Context	16
4.1.1 Broader Context	16
Table 2 - Design Context	16
4.1.2 User Needs	17
4.1.3 Prior Work/Solutions	17
4.1.4 Technical Complexity	18
4.2 Design Exploration	18
4.2.1 Design Decisions	18
4.2.2 Ideation	19
4.2.3 Decision-Making and Trade-Off	20
Table 3 - Decision Making	20
4.3 Proposed Design	20
4.3.1 Design Visual and Description	21
Figure 3 - Flow of Data Diagram	21
Figure 4 - Swimlane Diagram	21
4.3.2 Functionality	22
4.3.3 Areas of Concern and Development	22
4.4 Technology Considerations	22
4.5 Design Analysis	22
4.6 Design Plan	23
<b>5 Testing</b>	<b>24</b>
5.1 Unit Testing	24
5.2 Interface Testing	25
5.3 Integration Testing	25
5.4 System Testing	26
5.5 Regression Testing	26
5.6 Acceptance Testing	26
5.7 Security Testing (if applicable)	27

5.8 Results	27
Figure 5 - Test Flow Diagram	27
Figure 6 - High-level Test Flow Diagram	27
<b>6 Implementation</b>	<b>28</b>
<b>7 Professionalism</b>	<b>29</b>
7.1 Areas of Responsibility	29
7.2 Project Specific Professional Responsibility Areas	31
7.3 Most Applicable Professional Responsibility Area	32
<b>8 Closing Material</b>	<b>33</b>
8.1 Discussion	33
8.2 Conclusion	33
8.3 References	33
8.4 Appendices	33
8.4.1 Team Contract	33

# List of Figures/Tables/Symbols/Definitions

## X.1 FIGURES

Figure 1 - Task Decomposition Diagram

Figure 2 - Gantt Chart

Figure 3 - Flow of Data Diagram

Figure 4 - Swimlane Diagram

Figure 5 - Test Flow Diagram

Figure 6 - High-level Test Flow Diagram

## X.2 TABLES

Table 1 - Personal Efforts

Table 2 - Design Context

Table 3 - Decision Making

## X.3 DEFINITIONS

Unity - Game development engine, used to create 3D visualizations for the Microsoft HoloLens 2

Flask - Python web framework, used to implement our backend service for data processing

GraphQL - Alternative API architecture to REST, used for our internal data calls from backend to Unity client

Mixed Reality Toolkit - Cross platform toolkit for building AR experience, providing us with components and features to accelerate our development process (AR functionality)

# 1 Team

## 1.1 TEAM MEMBERS

Scott Frank	Ami Ikanovic	Isaac Littler
Benjamin Kelly	Zahydee Machado	Elizabeth Nelson
Parth Padmanabhan		

## 1.2 REQUIRED SKILL SETS FOR YOUR PROJECT

Computer / Electrical / Software Engineering

## 1.3 SKILL SETS COVERED BY THE TEAM

Computer Engineering: Isaac Littler

Electrical Engineering: Scott Fank

Software Engineering: Benjamin Kelly, Zahydee Machado, Ami Ikanovic, Parth Padmanabhan, Elizabeth Nelson

## 1.4 PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM

Agile-Waterfall hybrid

## 1.5 INITIAL PROJECT MANAGEMENT ROLES CHANGE

Team Lead: Elizabeth Nelson

Scheduling Lead: Isaac Littler

Client Communication Lead: Scott Fank

Team Communication Lead: Ami Ikanovic

Facilities Manager: Benjamin Kelly

Documentation Lead: Parth Padmanabhan

Delivery Manager: Zahydee Machado



## 2 Introduction

### 2.1 PROBLEM STATEMENT

Geographical data is often difficult to interpret and visualize. It can be challenging to portray data in a manner that conveys a sense of global scale to users.

### 2.2 REQUIREMENTS & CONSTRAINTS

- The application will be able to track physical objects to map the globe's location
- The application will be able to scale the three-dimensional globe to the user's liking
- The application will be able to rotate the globe to give users a full 360-degree view of all geographical-based data
- The globe will be able to remain stationary in the "world" independent from the user's location
- This application will require a 6' x 6' unobstructed space (empty 6 x 6 space) (Constraint)
- This development and use of the application will remain under a \$1000 dollar budget (Constraint)
- This application will require the use of a Microsoft HoloLens (Constraint)
- This application will have its own API layer that standardizes incoming data streams (Compatibility layer)
- This application will use the implementation of real-time data as well as historical data
- This application will be able to take 3 different types of geographical data streams as input
- This application will be able to layer different types of data on the globe visualization for a multi-type data visualization
- This application will be built using the Unity engine
- This application will maintain a stable 60 fps through the HoloLens visualization (Constraint)
- This application will be able to run for an hour on one HoloLens battery charge
- This application will have a "pick up and go" ease of use where no technical developer is needed for a user to understand how to run the application
- This application will create a 720p resolution image as specified in HoloLens standards
- The project implementation needs to be completed within 14 weeks. (time constraint)
- This application will need to run smoothly utilizing the limited HoloLens hardware performance (Constraint)
- This application will need to run using the Unity engine under the constraint of the system requirements needed for Unity (Constraint)

### 2.3 ENGINEERING STANDARDS

- Microsoft HoloLens Development Standards
- To meet a threshold of usability and performance for the user
- Unity Engine Development standards
- To meet a threshold of usability and performance for the user
- To ensure ease of development

- IEEE 802.11 Standard for Information Technology - Telecommunications and Information Exchange between systems
- To maintain reliable and stable connection between hardware devices and software application

## 2.4 INTENDED USERS AND USES

- Benefits and Stakeholders
  - General Users, will be able to see visualization of geographical data and experience a practical application of augmented reality
  - Client, will be able to check weather patterns, import custom streams of historical and real time data and showcase the ability of senior engineering students to faculty
  - Developers, will be able to showcase their abilities and develop a fully usable and finished product
- Use Case
  - User will be able to see weather data visualized on a three dimensional globe
  - Users will be able to scale the size of the three dimensional globe
  - Users will be able to rotate the globe to see all geographical based data
  - Users will be able to move around the space while the globe remains stationary in relation to the space
  - Users will be able to select from available data streams they want to visualize on the globe
  - Users will be able to select and see historical data visualization
  - Users will be able to pick up the Hololens, put it on and begin to see the application without further technical knowledge needed
- Benefits visitors and prospective students of Iowa State to see the potential abilities they may learn
- Benefits any user wishing to visualize a data stream using AR
- Client
  - Check weather patterns
  - Import custom streams of historical and real time data
  - Experience practical applications of augmented reality

## 3 Project Plan

### 3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

The Real Time Data Visualization project will follow a hybrid management style of Agile and Waterfall. The software has parts that can be developed in parallel and parts that will need to be developed sequentially. Some parts of the software will be developed to a usable extent, moved on from and then returned to to be fleshed out and fully completed. For example, when the core functionality of our Unity application needs to make API calls the custom API layer will send mock data until real data is received from external API's and standardized by the custom API layer. Agile will allow us this flexibility and dynamic development while Waterfall will cover our well defined sequential and parallel order of development.

The progress of this project will be tracked using Gitlab Issues. This feature will allow us to express all planned “issues” and milestones in an organized fashion in the same location as our code base. All additional communication is and will continue to be done using a team discord server.

### 3.2 TASK DECOMPOSITION

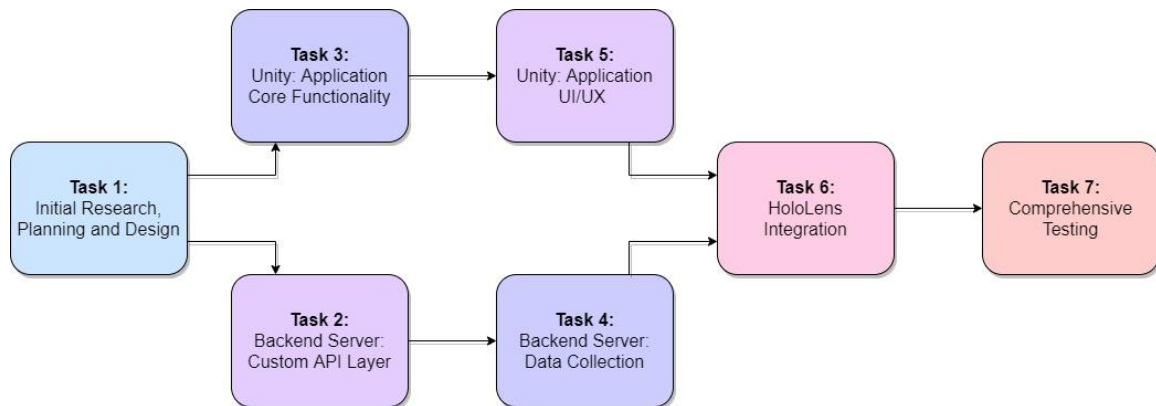


Figure 1 - Task Decomposition Diagram

### 3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

1. Initial Research, Planning and Design
  - 1.1. Research three competing applications already on the market
  - 1.2. Research five available APIs for data
  - 1.3. Research five available hardware platforms for visualizer
  - 1.4. Research three available software platforms for design and development
  - 1.5. Create three initial project solutions based on completed research to present to client
2. Backend Server: Custom API Layer
  - 2.1. Server is able to send and receive custom format data to and from Unity application
  - 2.2. Data collected is organized into a singular custom API format for use in Unity
3. Unity: Application Core Functionality
  - 3.1. Application can call custom made API to receive data
  - 3.2. Implementation three dimensional globe
  - 3.3. Single data stream is geographically coordinated and displayed on globe
  - 3.4. Multiple data streams are geographically coordinated and displayed on globe
  - 3.5. Implementation of zooming in on specific location
  - 3.6. Implementation of scaling the globe model
  - 3.7. Implementation of multiple data streams layered over one another
4. Backend Server: Data collection
  - 4.1. Server calls and receives historical data from one API
  - 4.2. Server calls and receives realtime data from one API
  - 4.3. Server calls and receives data from multiple API's
5. Unity: Application UX/UI
  - 5.1. implementation of custom pattern visualization for individual data streams
  - 5.2. Implementation of custom Legend's for individual data streams
  - 5.3. Implementation of menu selection to display data streams
6. Hololens Integration
  - 6.1. HoloLens default hand gestures integrated with menu selection
  - 6.2. HoloLens default hand gestures integrated with scaling and zooming
7. Comprehensive Testing
  - 7.1. User can navigate all primary program features within 10 minutes
  - 7.2. Data retrieval and command results need to be processed within 1 seconds of user input
  - 7.3. Application will maintain 720p resolution image
  - 7.4. Globe is scalable to an approximate two feet tall without pixelation
  - 7.5. Full-functionality visualization running at a stable 60 fps

### 3.4 PROJECT TIMELINE/SCHEDULE

Our project schedule follows as:

- Semester 1
  - Week 1-4: Research & Planning
  - Week 5-12: Design
- Semester 2
  - Week 1-2: Backend Server: Custom API Layer
  - Week 1-6: Unity: Application Core Functionality
  - Week 3-6: Backend Server: Data Collection
  - Week 7-9: Unity: Application UX/UI
  - Week 10-11: HoloLens Integration
  - Week 12: Comprehensive Testing

Second semester marks the beginning of our implementation stage. We start off the first two weeks developing “Backend Server: Custom API Layer” and “Unity: Core Application” in parallel. This will allow the core application to make mock calls to the custom API layer and receive mock data. Week three will mark the start of “Backend Server: Data Collection” which will continue to be developed to give the unity core application real data. The custom API layer will be returned to at various points to be fleshed out and corrected correspondingly to the data collection development. Once we complete the data collection layer we can begin with “Unity: Application UX/UI” development where we spend the next two weeks making the UX/UI presentable and cohesive with data visualization layers. Once that is completed we will begin to integrate everything into the HoloLens and perform comprehensive testing of the application.

To view spreadsheet:  Gantt chart

## GANTT CHART

Smartsheet Tip → A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE	3D Data Visualizer	GROUP NAME	SDMAY-22-21
PROJECT MANAGER	Joseph Zambreno	DATE	10/10/21

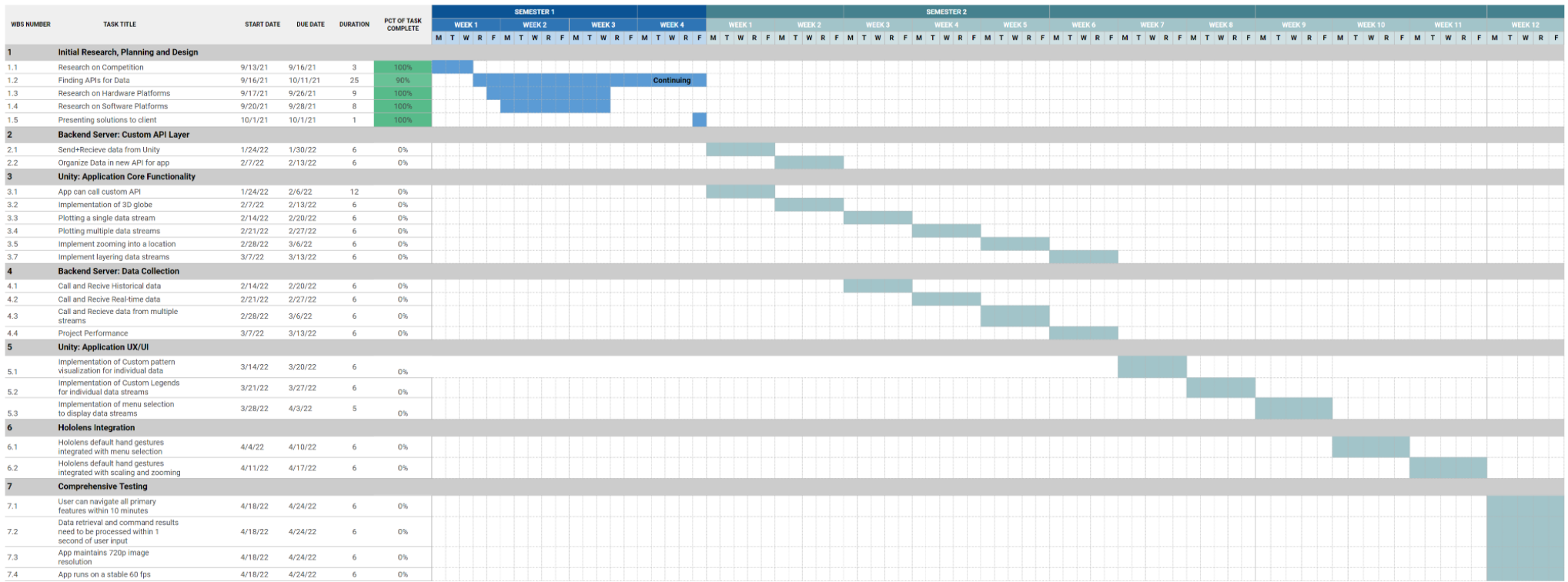


Figure 2 - Gantt Chart

### 3.5 RISKS AND RISK MANAGEMENT/MITIGATION

- Initial Planning and Research
  - Resources found aren't compatible/usable for our project (0.3)
- Backend Server: Custom API
  - Not able to abstract out multiple external API's into one custom layer (0.5)
    - Risk Mitigation: Limit API support for similar API sources
- Unity: Application Core Functionality
  - Zooming functionality is unachievable (0.2)
  - Scaling functionality is unachievable (0.2)
  - Layering multiple data streams functionality is achievable (0.3)
- Backend Server: Data collection
  - External API changes or deprecation of free data sources
    - Risk Mitigation: Use the paid version of the OpenWeather API service
- Unity: Application UX/UI (UI, modeling, etc.)
  - UI design isn't intuitive to new users (0.3)
  - Layering of multiple custom patterns create incoherent or displeasing visualization (0.6)
    - Risk Mitigation: Limit the amount of layers a user can enable at once
  - Layering of multiple custom patterns causes performance issues (0.6)
    - Risk Mitigation: Limit the amount of layers a user can enable at once
- HoloLens Integration
  - HoloLens default hand gestures are not intuitive for scaling and zooming (0.5)
    - Risk Mitigation: Create custom hand gestures or use a device (Controller) for UI/UX application interaction
- Comprehensive Testing
  - 60 fps target not consistently met (0.1)

### 3.6 PERSONNEL EFFORT REQUIREMENTS

Tasks	Person-hours
Research, Planning and Design	84 (3 hours/week * no. of weeks * no. of members)
Backend Server: Custom API Layer	70 (5 hours/week * no. of weeks * no. of members)
Unity: Application Core Functionality	245 (5 hours/week * no. of weeks * no. of members)
Backend Server: Data collection	140 (5 hours/week * no. of weeks * no. of members)
Unity: Application UX/UI (UI, modeling, etc.)	105 (5 hours/week * no. of weeks * no. of members)
Hololens Integration	70 (5 hours/week * no. of weeks * no. of members)
Comprehensive Testing	140 (5 hours/week * no. of weeks * no. of members)

*Table 1 - Personal Efforts*

### 3.7 OTHER RESOURCE REQUIREMENTS

- Microsoft HoloLens 2
- Unity Hub
- Unity Asset Store
- Weather APIs



## 4 Design

### 4.1 DESIGN CONTEXT

#### 4.1.1 Broader Context

This 3D data visualization application is being designed as a showcase piece for the Department of Electrical and Computer Engineering at Iowa State. It will provide an educational experience with one of the newest trends in visualization technology and hardware in order to highlight the university's commitment to research and innovation. The application is intended to attract and communicate to prospective students what their future at Iowa State can look like.

Area	Description	Examples
Public health, safety, and welfare	This project can allow the various users to learn and become aware of the circumstances and challenges that populations around the globe are facing.	Depending upon the data called by the program, we could alert users to dangerous weather patterns, virus hotspots or environmental pollutants.
Global, cultural, and social	The project will reflect the goals and values of the Iowa State University community. It will aim to showcase the potential of the education provided at the Department of Electrical and Computer Engineering and their research goals.	It will showcase the commitment to innovation and research of the ECpE by incorporating new technologies and ideas.
Environmental	Due to the small scale and software-based nature of the project, the environmental impact is minimal and limited to that of manufacturing the single unit Hololens 2 that will be used to run the project.	This project will require a manufactured Hololens 2 device to run and energy to recharge it periodically.
Economic	This project might factor into prospective students' decisions about whether to pursue their studies at Iowa State. This could potentially influence the university's enrollment numbers and therefore have an economic impact on it.	Product creates opportunities for the economic advancement of Iowa State University by targeting prospective students.

*Table 2 - Design Context*

#### 4.1.2 User Needs

- Prospective students need a way to see what sort of projects and education lies ahead if they were to come to Iowa State University and enroll in Electrical / Computer Engineering
- EE/CPRE faculty need a way to showcase their department's achievements and innovative practices to prospective students as well as visitors interested in the department
- Visitors need a way to physically see examples of what goes on in the EE / CPRE department because depending on who they are they could want to create business or educational connections with the department
- Current students need a way to learn about ongoing development of projects to make decisions on research, classes, and areas of study they want to pursue.

#### 4.1.3 Prior Work/Solutions

The majority of visualization systems of global scale are 2D and only compatible with traditional computer monitors or touch screen displays. Those visualizers generally are limited to displaying one type of data such as weather, population, pollution, to serve a specific purpose. There are some 3D visualizers, such as Google Earth, that focus on displaying details of the earth and its geography. The application we found most similar to our application goal is MeteoEarth. This application displays a 3D globe on a 2D display with various weather data types. Through our research we were unable to find a Hologens application that will do the same real time data visualization that we intend to do.

The following is a Pros/Cons list describing how our target solution will stand against others.

##### Pros

- "Real" 3D as our solution will use the Microsoft Hologens 2. This will allow users to see the globe as it is in real life and see geographical phenomena on a proper and global scale.
- Multiple types of data streams visualized at once layering one over the other in a cohesive manner
- Extensive user interactivity involving the ability to move the hologens(display) around a stationary globe
- Scalable globe
- The ability to input any data type following our API standard to visualize on the globe
- Global scale and localized scale of visualization in the same application

##### Cons

- Using the HoloLens 2 as our primary display creates an expensive initial cost for a user
- The use of the Hologens 2 only allows one user at a time to see the visualization if a group doesn't have multiple headsets

#### 4.1.4 Technical Complexity

The architecture of the 3D data visualization application has multiple subsystems that connect to create the cohesive whole. Each of these subsystems will involve different scientific, mathematical and engineering principles as part of the development solution.

##### Backend: Data Collection

- This component will require an understanding on efficient manipulation of large datasets to ensure that an acceptable level of system performance is maintained. It will also require investigation to find reputable data sources that meet the project criteria.

##### Backend: Custom API Layer

- Because the function of this subsystem is to standardize incoming data into a single source and format, it will require a strong understanding and analysis of scientific weather data and weather patterns. The creation of a custom API layer that accepts the importation of external data will require a lot of efficient and intuitive API design principles

##### Unity: Application Core Functionality

- This subsystem is a large percentage of the overarching system. The unity application needs to take geological data (coordinate-based data) and correlate that data to a 3d model of the globe. This will require mathematical principles in the field of geometry. It will also require an understanding of computer graphics for converting the individual data points into a cohesive visualization. We will need to create visual meshes that adhere to current standards in weather data visualizations, but also represent it in a 3D manner.

##### Unity: Application UI/UX

- This subsystem handles the interaction between the user and the program. Good UI/UX design will be required to make the program viable for our intended user base. This will need to involve principles in the areas of accessibility and usability.

## 4.2 DESIGN EXPLORATION

### 4.2.1 Design Decisions

1. Display as a Hololens 2 AR application
2. Use the Unity Game Engine for development
3. Use of OpenWeather API

#### 4.2.2 Ideation

When choosing our platform, we considered the following options:

- VR (Virtual Reality Interface)
- AR (Augmented Reality Interface)
- Sphere Display (Gakken WorldEye)
- 3D Monitor / Projection
- Touch Screen Device/Computer Monitor

We individually did research for APIs, display hardwares, and design softwares. We compiled all the information we found and discussed them. This allowed us all to share our own ideas and learn about more opportunities available to us.

### 4.2.3 Decision-Making and Trade-Off

We discussed all of our options and listed out our pros and cons for each. We then narrowed our options down to three and presented them to our client. Based on his recommendations, we chose an AR interface: the Microsoft HoloLens 2.

<p><b>VR</b></p> <ul style="list-style-type: none"> <li>● Pros <ul style="list-style-type: none"> <li>○ No physical bounds for visualization</li> <li>○ Within given budget</li> </ul> </li> <li>● Cons <ul style="list-style-type: none"> <li>○ Limited to one user at a time</li> <li>○ Software heavy/does not take advantage of other skill sets</li> <li>○ Not providing a physical device</li> </ul> </li> </ul> <p><b>AR (Augmented Reality Interface)</b></p> <ul style="list-style-type: none"> <li>● Pros <ul style="list-style-type: none"> <li>○ No physical bounds for visualization</li> <li>○ Easier access for the everyday man</li> <li>○ Within given budget</li> </ul> </li> <li>● Cons <ul style="list-style-type: none"> <li>○ Not as innovative as other options</li> <li>○ Restricted to device resolution and specifications</li> <li>○ Software heavy/does not take advantage of other skill sets</li> </ul> </li> </ul> <p><b>Sphere Display (Gakken WorldEye)</b></p> <ul style="list-style-type: none"> <li>● Pros <ul style="list-style-type: none"> <li>○ Unique displays allows for global viewing</li> <li>○ Includes interesting hardware</li> </ul> </li> <li>● Cons <ul style="list-style-type: none"> <li>○ Low Resolution (on the Gakken display)</li> <li>○ Details/Data hard to display</li> <li>○ Out of budget (on the vast majority of solutions)</li> </ul> </li> </ul>	<p><b>3D Monitor / Projection</b></p> <ul style="list-style-type: none"> <li>● Pros <ul style="list-style-type: none"> <li>○ Large scale viewing options</li> <li>○ Multi-user interactable</li> <li>○ No glasses/headset required</li> </ul> </li> <li>● Cons <ul style="list-style-type: none"> <li>○ Limited viewing angles w/ some equipment</li> <li>○ Not as innovative as other options</li> <li>○ Software heavy/does not take advantage of other skill sets</li> <li>○ Out of budget (on the vast majority of solutions)</li> </ul> </li> </ul> <p><b>Touch Screen Device/Computer Monitor</b></p> <ul style="list-style-type: none"> <li>● Pros <ul style="list-style-type: none"> <li>○ Easier than most solutions</li> <li>○ Within given budget</li> </ul> </li> <li>● Cons <ul style="list-style-type: none"> <li>○ Most display are not convex</li> <li>○ Similar to Google Maps or MeteoEarth</li> <li>○ Software heavy/does not take advantage of other skill sets</li> </ul> </li> </ul>
--	--

*Table 3 - Decision Making*

### 4.3 PROPOSED DESIGN

Currently, the team is trying to implement a Unity project and use the Microsoft HoloLens emulator. We have been researching how to make calls to our chosen API's. Previously, we got

access to a Hololens device and verified that it meets the expectations for the display. So far, our research and trials have indicated that our design decisions are viable and a good fit for the project.

### 4.3.1 Design Visual and Description

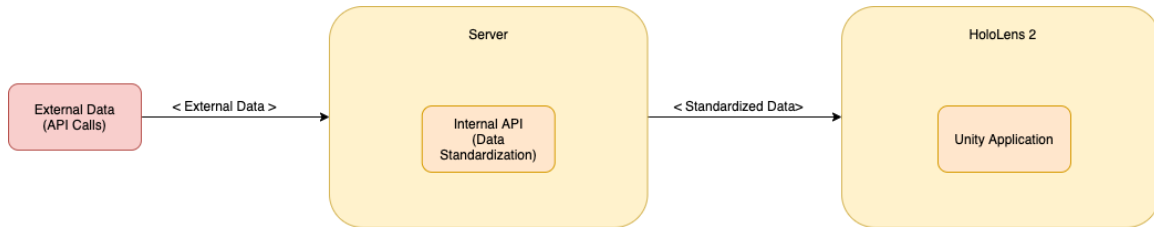


Figure 3 - Flow of Data Diagram

1. External data comes into the server through API calls
2. The server organizes and standardizes the data to our internal API specifications
3. The unity application calls our internal API and receives the standardized data
4. The unity applications translates the data into a 3D visualization
5. The HoloLens 2 displays the visualization for the user

### TIMELINE OF EVENTS DIAGRAM (Swim lane)

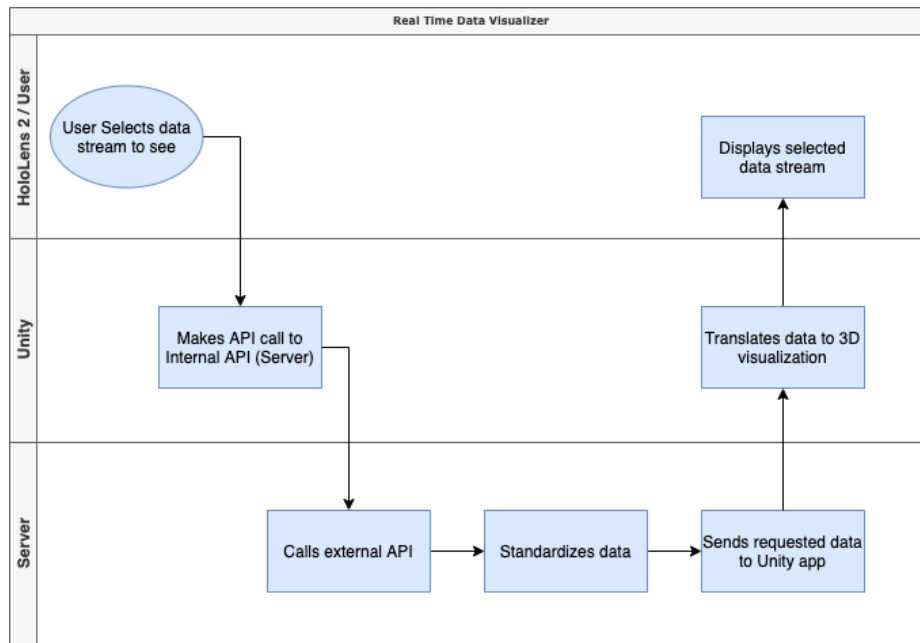


Figure 4 - Swimlane Diagram

### 4.3.2 Functionality

Expected usage sequence:

1. Users are instructed on the project's background, purpose, and what they are about to see and interact with by the person supervising the demo.
2. Users are walked through putting on the Hololens 2 device and securing it to their head with any necessary adjustments.
3. The program is started by the users and they are left to experiment with it with the supervising person available for any questions or difficulties.

The current design would satisfy the requirements by meeting the client's and potential user's needs.

### 4.3.3 Areas of Concern and Development

The primary concerns are making sure the product adequately manages to show the intended data in a way that is easy for users to understand and navigate without frustration. This is meant as a demonstration piece for the visualization hardware and techniques, so the quality of the user experience is fundamental to its purpose.

The immediate plans for addressing those concerns are to rely on well-researched UX and UI techniques, do early functional testing with people in the user group, and prioritize performance for a smooth viewing experience.

Questions for the client and faculty advisor include:

1. Approximately how long should it take a user to try out every viewing mode in the application?
2. How much time is a user intended to spend interacting with the application?
3. What is the main takeaway a user should have from the experience?

## 4.4 TECHNOLOGY CONSIDERATIONS

After deciding to use AR technology, we had to choose between the Hololens and the Hololens 2

Hololens:

1. Already owned by the university (no purchase required, stays within budget)

Hololens 2:

1. Greater field of view compared to the Hololens (52° vs 34°)
2. Higher display resolution compared to the Hololens (2048 x 1080 px vs 1280 x 720 px)
3. Had to be purchased for the purpose of this project (price point beyond budget)

## 4.5 DESIGN ANALYSIS

Currently, we are developing a prototype consisting of a static 3D visualization of the globe which can be manipulated within the Hololens 2. We have been successful in creating the globe's sphere and altering its position and size. However, we are still working on grounding the sphere in our

reality in some way so it will have a fixed position until the user moves it. The basic premise of our design seems to be highly feasible and leaves room for innovation and creativity.

#### 4.6 DESIGN PLAN

Our design process will follow the task schedule outlined above, with the project requirements acting as a guide for design choices (both large and small). Certain design elements will be refined through iterative methods, allowing us to actively gauge their effectiveness.



## 5 Testing

### 5.1 UNIT TESTING

Unity Application

Tool: Unity Test Framework

- UI Tests
  - Globe Scaling
    - Pass: Gesture enables and disables zooming mode
    - Fail: Zooming mode can't be enabled or disabled via the UI
  - Globe Rotation
    - Pass: Gesture enables and disables rotation mode
    - Fail: Rotation mode can't be enabled or disabled via the UI
  - Data Overlays
    - Pass: Selecting and deselecting data streams toggles their visibility on the globe
    - Fail: Data streams can't be toggled on and off
  - Application Exit
    - Pass: Pressing on the exit button closes the application
    - Fail: Application can't be exited gracefully from the UI
- Visualization Logic Tests
  - Data Refreshing
    - Pass: Given new data, old overlays are discarded and replaced with newly generated ones
    - Fail: Old overlays are not properly deleted or are not replaced with a correct version
  - Overlay Generation
    - Pass: For each data set, an overlay texture is generated for the globe
    - Fail: Overlays are malformed or missing for any data set
  - Multiple Overlays
    - Pass: Multiple overlays (up to 3) can be overlapped without interfering with each other
    - Fail: Additional overlays obstruct or cover existing ones
- Server Interaction
  - Data Retrieval
    - Pass: Application can make appropriate requests to the server for the data needed by the overlays enabled through the UI
    - Fail: Requests are malformed, unnecessary, or retrieving the wrong data

Data Server

Tool: Postman / Pytest

- External Data Retrieval
  - Data fetching
    - Pass: Application can make appropriate requests to external APIs for the data requested by the Unity client
    - Fail: Requests are malformed, unnecessary, or retrieving the wrong data
  - Query Building

- Pass: Queries are built with the correct parameters to fetch the necessary data
    - Fail: Queries are missing filters or other necessary information to make the correct request
  - Data processing
    - API layer
      - Pass: External data is mutated to fit the visualization client's standard and format
      - Fail: Data is lost or malformed in the mutation

## 5.2 INTERFACE TESTING

Server:

- External API Interface
  - This will be validated and verified using Unit Tests
  - This will be functionally tested using a tool like Postman to make sure requests are compatible with the used external APIs
- Internal API Interface
  - Unit tests will cover the individual API operations
  - Functional tests will be made using a tool like Postman to ensure that the server responds as expected to API calls

Unity Application:

- Server API interface
  - Unit tests will verify that the requests meet the specifications and requirements
  - Functional tests will be made using a tool like Postman to verify that requests are compatible with the server API specification

## 5.3 INTEGRATION TESTING

We will test the external API -> HoloLens -> Unity engine path to ensure that our implementation doesn't overwhelm the hardware resources of the HoloLens. The HoloLens framerate, processor/memory/network utilization, and other performance metrics will be viewed and inspected through the Windows Device Portal and the Windows Performance Analyzer. These tools will allow us to ensure that our performance requirements (particularly, those pertaining to framerate and smoothness) are met.

An integration test for the Unity Application will involve verifying that given a mock data stream, the unity application will be able to generate the visualization and respond to various types of UI interactions. This test can be written using the Unity Test Framework in Play Mode.

An integration test for the server will involve verifying that for a request made to its API, it is able to generate the appropriate queries to return a mock stream of external data, mutate it to the appropriate standard, and return the new formatted data to the client.

## 5.4 SYSTEM TESTING

Functional Requirements :

- The application will be able to scale the three dimensional globe to the user's liking
  - The Unity UI unit tests and the Unity application integration test will verify this requirement is met.
- The application will be able to rotate the globe to give users a full 360 degree view of all geographical based data
  - The Unity UI unit tests and the Unity application integration test will verify this requirement is met.
- This application will have its own API layer that standardizes incoming data streams (Compatibility layer)
  - The Server unit tests and integration test will verify this requirement is met.
- This application will be able to take 3 different types of geographical data streams as input
  - The Unity UI unit tests and the Unity application integration test will verify this requirement is met.
- This application will be able to layer different types of data on the globe visualization for a multi type data visualization
  - The Unity UI unit tests and the Unity application integration test will verify this requirement is met.

## 5.5 REGRESSION TESTING

Unit tests and integration tests will automatically run on a test stage of our CI/CD pipeline, so that new additions are proven to work along existing functionality. This will be done with the Gitlab CI/CD pipelines tool and its merged result functionality which will run as if the changes from the source branch have already been merged into the target branch. Merge requests that fail these tests will not be merged into the project. The unit tests and integration tests will be made in accordance with the requirements, so this approach will verify that overall project requirements are still being met with every addition. Additions that don't include any necessary tests for added functionality will also not be merged.

## 5.6 ACCEPTANCE TESTING

Extensive manual testing with the Hololens device will be done by team members and volunteer beta testers to verify that non-functional requirements are being met. The requirements that will be focused on during this testing stage are:

- This application will have a "pick up and go" ease of use where no technical developer is needed for a user to understand how to run the application
- This application will maintain a stable 60 fps through the Hololens visualization (Constraint)
- This application will need to run smoothly utilising the limited Hololens hardware performance (Constraint)

The client will also be asked to interact with the program and give feedback about each feature at different stages of early development and prototyping.

### 5.7 SECURITY TESTING (IF APPLICABLE)

Not applicable.

### 5.8 RESULTS

These tests will ensure that our project implementation will comply with the project requirements and meet the client's needs. A thorough testing methodology will allow us to avoid any hidden or edge-case problems that, while not impacting the majority of the user experience, could compromise the reliability and accuracy that our project intends to achieve. Our testing will also help us evaluate the overall usability of our project for the end user.

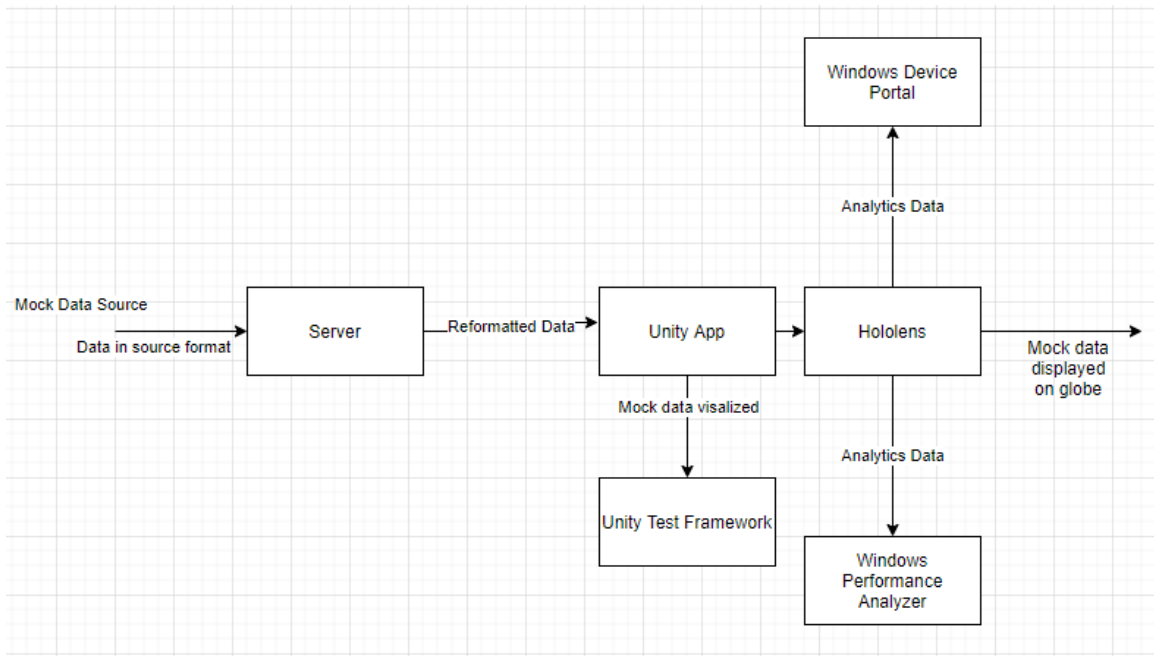


Figure 5 - Test Flow Diagram

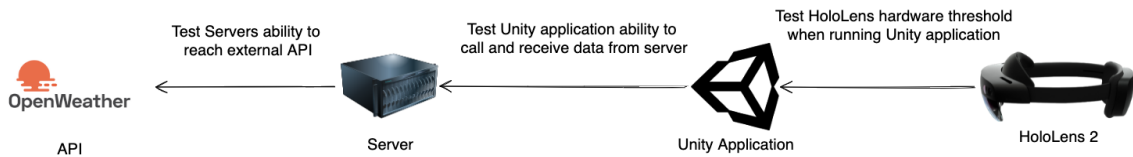


Figure 6 - High-level Test Flow Diagram

## 6 Implementation

Our preliminary implementation includes finishing our globe prototype using Unity for the Hololens. The data for this prototype will be static and will simply test if the visualizer works. Along with this prototype, we will create a prototype of the server we will use to parse and organize the data we receive. Again, the data received will not be from our intended data sets until we are able to test its functionality.

## 7 Professionalism

### 7.1 AREAS OF RESPONSIBILITY

1. Work Competence
  - a. “2.2 Acquire and maintain professional competence”
    - i. “A professional must participate in setting standards for appropriate levels of competence, and strive to achieve those standards. Upgrading technical knowledge and competence can be achieved in several ways: doing independent study; attending seminars, conferences, or courses; and being involved in professional organizations.”
  - b. Keep yourself up to date with modern standards by actively learning something new to maintain professional competence. This is done on an individual level as a responsibility to stay professional.
  - c. “II. 2 Engineers shall perform services only in the areas of their competence.” “II. 5 Engineers shall avoid deceptive acts.”
    - i. According to the NSPE, an engineer needs to be qualified to perform a certain task before they choose to undertake it. They should not sign up for any tasks they have no control over. They must choose to lead a project only when they have sufficient experience in the field. An engineer must not falsify their qualifications to secure work, nor can they bribe someone to tip scales in their favor
2. Financial Responsibility
  - a. “1.3 Be honest and trustworthy”
    - i. “Computing professionals should be honest about their qualifications, and about any limitations in their competence to complete a task.”
  - b. Clearly communicate technical and personal limitations that might conflict with the ability to complete a task, not misrepresenting skillset. If a new feature might not be feasible to implement, that is clearly communicated to stakeholders
  - c. “II.4 Engineers shall act for each employer or client as faithful agents or trustees”
    - i. The NSPE describes that engineers shall be upfront about potential conflicts of interest when providing a service, engineers will not accept compensation from more than one party for the exact same work or accept compensation for work not done by them. ACM more so says that the engineer shall be honest about the work that they have completed, its value to the customer and what it is worth. The engineer shall not overstate their work in an attempt to be compensated more.
3. Communication Honesty
  - a. “1.3 Be honest and trustworthy”
    - i. “The honest computing professional will not make deliberately false or deceptive claims about a system or system design, but will instead provide full disclosure of all pertinent system limitations and problems.”

- b. Communicate truthfully in regards to qualifications, work completed, complications encountered, etc. and do not deceive or intentionally mislead/misguide users, investors, society, employees, or employers.
  - c. "II.3 Engineers shall issue public statements only in an objective and truthful manner."
    - i. The NSPE has multiple sections which address communicational honesty. This section, in particular, addresses how an engineer should present information about successes, challenges, and employment to the public. It notes, "They shall include all relevant and pertinent information in such reports, statements, or testimony, which should bear the date indicating when it was current."
- 4. Health, Safety, Well-Being
  - a. "1.2 Avoid harm to others"
    - i. "This principle prohibits use of computing technology in ways that result in harm to any of the following: users, the general, public, employees, employers"
  - b. Do everything possible as a company, developer, designer ect. to mitigate the risk of harm resulting from your work. This entails thought, analysis, testing and following set standards when creating solutions to problems. When signs of risk or future harm are noticed in the system it is your obligation to report it no matter the direct consequence that may come to you as an individual.
  - c. "II.1. Engineers shall hold paramount the safety, health, and welfare of the public."
    - i. NSPE gives principles of how engineers should operate in their work to not partake in unlawful practices. These unlawful practices can in turn cause harm to public welfare. ACM focuses directly on the actions and product causing harm to the public
- 5. Property Ownership
  - a. "1.6 Give proper credit for intellectual property"
    - i. "Computing professionals are obligated to protect the integrity of intellectual property. Specifically, one must not take credit for other's ideas or work, even in cases where the work has not been explicitly protected by copyright, patent, etc."
  - b. This means that we shouldn't take credit for work that isn't ours. Situations where intellectual property is presented (directly or indirectly) as being owned by anyone other than the rightful owner should be actively avoided. Even if the intellectual property isn't protected by law, careful attention should be given to ensure that proper ownership is respected.
  - c. "III.9 Engineers shall give credit for engineering work to those to whom credit is due, and will recognize the proprietary interests of others."
    - i. The NSPE entry focuses a bit more on the proper usage of others' resources (only utilizing resources in ways that have been authorized, not duplicating designs without the owner's permission, etc.), while the AMC code focuses on properly crediting the creators of an intellectual property.
- 6. Sustainability
  - a. "1.1 Contribute to society and human well-being"
    - i. "In addition to a safe social environment, human well-being includes a safe natural environment. Therefore, computing professionals who design and

- develop systems must be alert to, and make others aware of, any potential damage to the local or global environment.”
    - b. Be aware of the impacts the product you’re developing can have on local and global communities in regards to environmental issues. Examples of this can include waste reduction, energy efficiency, and the durability or recyclability of materials.
    - c. “III.2 Engineers shall at all times strive to serve the public interest.”
      - i. The NSPE code mentions that “Engineers are encouraged to adhere to the principles of sustainable development in order to protect the environment for future generations.” It does not explicitly mention the need to alert others of the potential damage like the ACM does.
- 7. Social Responsibility
  - a. “1.1 When designing or implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare”
  - b. Computing professionals must avoid developing products that will be harmful to people and society at large. They must be aware and up to date on social concerns and potential consequences of their projects.
  - c. “III. 2 Engineers shall at all times strive to serve the public interest.” “Engineers shall avoid all conduct or practice that deceives the public.”
    - i. The NSPE says,engineers should participate in civic affairs to build and develop their community. They should expand awareness on the achievements of engineering, and stick to sustainable development to protect the environment. Engineers should not sign contracts that break these codes, and report any client that forces them to do so to the appropriate authorities. Engineers must keep their statements factual and must avoid manipulating or hiding data to fit a narrative.

## 7.2 PROJECT SPECIFIC PROFESSIONAL RESPONSIBILITY AREAS

1. Work Competence
  - a. This applies in our project because we are working on new and unfamiliar technology that requires our members to do independent study about the tools and development practices involved. Technical knowledge is being shared within the team and from outside sources in order to help team members achieve competence in the necessary areas of study.
  - b. The team is performing medium in this because we are still on the early setup stages in regards to the development of the software and more focused on design work. This has meant some technical learning needs to be delayed, but will be focused on closer to the implementation phase of the project.
2. Financial Responsibility
  - a. Financial responsibility plays a role in our project, as it can be tempting to select high-end data sources that are, while easier to work with, very expensive. The continuous costs of the data APIs could easily be passed on to the client and the university, degrading the current and long-term value of our project.
  - b. We believe that our performance for this responsibility is high. To minimize our project costs, we have located a free data source that will fulfill the needs of the client.



3. Communication Honesty
  - a. This is applicable to the project because of the relationship between team members and the client and advisor. The team must be upfront with the client about current progress and make an effort to keep them updated at reasonable intervals.
  - b. The team is performing high in this area because of regular meetings with our client where they have been kept up to date on current progress and have been able to give timely feedback and advice to the team.
4. Health, Safety, Well-Being
  - a. This is applicable to the project primarily because the software will be loaded onto a wearable device. The team will make sure to abide by the safety and health recommendations of the Hololens device in order to ensure that the software encourages proper and safe use of it.
  - b. The team is performing high in this area because we are abiding by Microsoft's official documentation when it comes to interactions on the Hololens and their recommended usage.
5. Property Ownership
  - a. This is applicable to our team as we will be using API sources to receive data. We will also be using a Microsoft Hololens 2, a device we did not create. It is important to share this information with our users and our client so we give credit where it is due.
  - b. The team is performing high in this area as we have communicated this well with our client as well as in our documentation.
6. Sustainability
  - a. This criteria does not apply to our project as we will be only using one device and the energy the device requires is minimal.
7. Social Responsibility
  - a. This is applicable to our team since the project is a display piece that showcases the ability of Iowa State University engineers. This shows prospective students what they can look forward to building during their time here.
  - b. This is not currently applicable to the project since it has not been deployed yet and we can not gauge the success of the project when it comes to attracting students.

### 7.3 MOST APPLICABLE PROFESSIONAL RESPONSIBILITY AREA

Communication Honesty is important to our project because of the relationship between our team and our client and advisor. It has meant keeping our client updated through regular meetings and reports as well as being upfront about any existing tools and software that will be utilized in our final product. We have been able to communicate effectively with the client and ensured that they understand the limitations of the project, difficulties, and planned features. This has allowed the client to respond and give feedback that impacts design decisions in a timely manner.

## 8 Closing Material

### 8.1 DISCUSSION

Not applicable to our current progress.

### 8.2 CONCLUSION

Not applicable to our current progress.

### 8.3 REFERENCES

Not applicable to our current progress.

### 8.4 APPENDICES

Not applicable to our current progress.

#### 8.4.1 Team Contract

Team Members:

- |                      |                   |
|----------------------|-------------------|
| 1) Ami Ikanovic      | 2) Benjamin Kelly |
| 3) Elizabeth Nelson  | 4) Isaac Littler  |
| 5) Parth Padmanabhan | 6) Scott Fank     |
| 7) Zahydee Machado   |                   |

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
  - a. Mondays at 4:30 (face-to-face with a virtual option)
  - b. Student Innovation Center
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
  - a. Discord between group members
  - b. Emails between client and group as well as weekly meeting via Zoom
  - c. Documentation on shared Google Drive
3. Decision-making policy (e.g., consensus, majority vote):
  - a. Voting based on motions
  - b. Unanimously agreed compromises
  - c. If changes to Team Contract need to be made, they must be unanimous
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
  - a. Meeting minutes will be kept in a Google Document
  - b. Minute keeper assigned on Sunday
  - c. Minutes should include project ideas, questions, problems, improvements, etc.

- d. Take note of roll call for each meeting

#### Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
  - a. Notify team members if meeting will be missed by Sunday unless there is an unexpected emergency
  - b. Expected to attend all meetings if able
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
  - a. Stay accountable for the work assigned
  - b. Continuous work throughout the time approaching the deadline
  - c. If a team member cannot meet the deadline, inform the team prior to the deadline
3. Expected level of communication with other team members:
  - a. Check the Discord daily
  - b. Communicate regularly with team about progress and problems
  - c. Respect other team members with differing opinions and ideas
4. Expected level of commitment to team decisions and tasks:
  - a. Contribute to decision making and assigned tasks

#### Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
  - a. Ami Ikanovic: Facilitator / Connector / Representative
  - b. Elizabeth Nelson: Facilitator / Arbitrator / Explorer
  - c. Benjamin Kelly: Arbitrator / Connector / Representative
  - d. Zahydee Machado: Gatekeeper / Arbitrator / Explorer
  - e. Parth Padmanabhan: Information Provider / Explorer / Facilitator
  - f. Scott Fank: Passive Collector / Initiator / Representative
  - g. Isaac Littler: Connector
2. Strategies for supporting and guiding the work of all team members:
  - a. Open and honest communication
  - b. Assigning roles with entire group present
  - c. Work as a large group, sub-groups, and individuals
3. Strategies for recognizing the contributions of all team members:
  - a. Begin meetings with a discussion of what was done, what is being worked on, and what will be done (Agile style)

#### Collaboration and Inclusion

1. Skills, expertise, and unique perspectives each team member brings to the team
  - a. Ami Ikanovic - Software focused, mainly back end work, data transformation, databases and API's. Client communication and management skills.
  - b. Elizabeth Nelson: Software Focused, Visual Design, Website Design, Experience with 3D programs such as Cinema4D, Blender, etc., Frontend Development
  - c. Benjamin Kelly - Software Focused, Visual Design, database management, 3D art & animation (Blender), frontend dev, OOP

- d. Zahydee Machado: Focus on Software, Interactive Media and Design, Game programming (Unity and Unreal), 3D art and animation (Blender), Frontend dev (Angular TS)
  - e. Parth Padmanabhan: Software focused, some circuit design experience, experience with Unity for game development (could be used in visualization). Taken classes for embedded systems and computer architecture.
  - f. Scott Fank - Hardware focused, circuits, 3D printing/modeling, general embedded systems
  - g. Isaac Littler - Software & embedded systems internships, CprE major focusing on software, Godot
2. Strategies for encouraging and support contributions and ideas from all team members:
    - a. Encouraging questions, ideas, and criticism by taking note of all ideas for future reference or use
    - b. Continuous brainstorming to find alternate solutions
  3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
    - a. First address team member
    - b. If not resolved, address among group
    - c. If not resolved, address with TA
    - d. If not resolved, address with professor

#### Goal-Setting, Planning, and Execution

1. Team goals for this semester:
  - a. Create a collaborative and productive group
  - b. Complete design of product and full report
2. Strategies for planning and assigning individual and team work:
  - a. Communication of tasks within a taskboard (Gitlab) and discuss which team members can complete them
3. Strategies for keeping on task:
  - a. Standup style meetings
  - b. Plan an agenda for following meetings based on meeting minutes
  - c. Motioning to vote

#### Consequences for Not Adhering to Team Contract

1. Identify the issue and address it as a group
2. If the issue persists, discuss with TA or professor

\*\*\*\*\*

- a) I participated in formulating the standards, roles, and procedures as stated in this contract.
- b) I understand that I am obligated to abide by these terms and conditions.
- c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

- |                      |                |
|----------------------|----------------|
| 1) Benjamin Kelly    | DATE 9/13/2021 |
| 2) Elizabeth Nelson  | DATE 9/13/2021 |
| 3) Zahydee Machado   | DATE 9/13/2021 |
| 4) Ami Ikanovic      | DATE 9/13/2021 |
| 5) Parth Padmanabhan | DATE 9/13/2021 |
| 6) Scott Fank        | DATE 9/13/2021 |
| 7) Isaac Littler     | DATE 9/13/2021 |