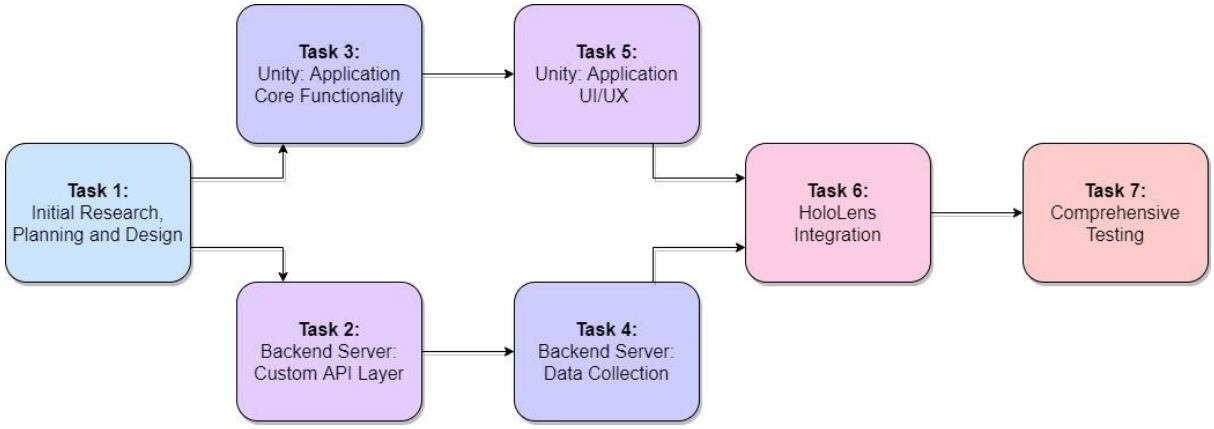
2 Project Plan

2.1 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 flushed 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.

2.2 Task Decomposition



2.3 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

2.4 Project Timeline/Schedule

Gantt Chart

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 flushed 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.

2.5 Risks And Risk Management/Mitigation

Consider for each task what risks exist (certain performance targets may not be met; certain tools may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

Agile projects can associate risks and risk mitigation with each sprint.

- 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)

2.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort total number of person-hours required to perform the task.

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 memb I'm ers)

2.7 Other Resource Requirements

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

- Microsoft HoloLens 1 (Possible upgrade to HoloLens 2)
- Unity Hub
- Unity Asset Store
- Weather APIs

Gantt Chart

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

WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Initial Research, Planning and Design**				
1.1	Research on Competition	9/13/21	9/16/21	3	100%
1.2	Finding APIs for Data	9/16/21	10/11/21	25	90%
1.3	Research on Hardware Platforms	9/17/21	9/26/21	9	100%
1.4	Research on Software Platforms	9/20/21	9/28/21	8	100%
1.5	Presenting solutions to client	10/1/21	10/1/21	1	100%
2	**Backend Server: Custom API Layer**				
2.1	Send+Recieve data from Unity	1/24/22	1/30/22	6	0%
2.2	Organize Data in new API for app	2/7/22	2/13/22	6	0%
3	**Unity: Application Core Functionality**				
3.1	App can call custom API	1/24/22	2/6/22	12	0%
3.2	Implementation of 3D globe	2/7/22	2/13/22	6	0%
3.3	Plotting a single data stream	2/14/22	2/20/22	6	0%
3.4	Plotting multiple data streams	2/21/22	2/27/22	6	0%
3.5	Implement zooming into a location	2/28/22	3/6/22	6	0%
3.7	Implement layering data streams	3/7/22	3/13/22	6	0%
4	**Backend Server: Data Collection**				
4.1	Call and Recive Historical data	2/14/22	2/20/22	6	0%
4.2	Call and Recive Real-time data	2/21/22	2/27/22	6	0%
4.3	Call and Recieve data from multiple streams	2/28/22	3/6/22	6	0%
4.4	Project Performance	3/7/22	3/13/22	6	0%
5	**Unity: Application UX/UI**				
5.1	Implementation of Custom pattern visualization for individual data	3/14/22	3/20/22	6	0%
5.2	Implementation of Custom Legends for individual data streams	3/21/22	3/27/22	6	0%
5.3	Implementation of menu selection to display data streams	3/28/22	4/3/22	5	0%
6	**Hololens Integration**				
6.1	Hololens default hand gestures integrated with menu selection	4/4/22	4/10/22	6	0%
6.2	Hololens default hand gestures integrated with scaling and zooming	4/11/22	4/17/22	6	0%
7	**Comprehensive Testing**				
7.1	User can navigate all primary features within 10 minutes	4/18/22	4/24/22	6	0%
7.2	Data retrieval and command results need to be processed within 1 second of user input	4/18/22	4/24/22	6	0%
7.3	App maintains 720p image resolution	4/18/22	4/24/22	6	0%
7.4	App runs on a stable 60 fps	4/18/22	4/24/22	6	0%