Presenter, company name and addressTitle and subtitleVersion number and dateDecorative sidebar

Table of Contents

[Project Description 2](#_heading=h.gjdgxs)

[Overview (CO-5) 2](#_heading=h.30j0zll)

Global Trends (EM@FSE-E)3

Market Analysis (EM@FSE-K)3

Security [Considerations (SER-2)](#_heading=h.30j0zll) 4

Key Requirements [(SER-2)](#_heading=h.3znysh7) 5

Deliverables (CER-1)6

Acronyms and Abbreviations [(ABET-3):](#_heading=h.30j0zll) 8

Design and Architecture9

Design Description [(ABET-1, ABET-2):](#_heading=h.1t3h5sf) 9

Alternate Design Possibilities [(EM@FSE-B):](#_heading=h.4d34og8) 10

Implementation Strategy11

High-Level Work Breakdown Structure [(SER-1):](#_heading=h.17dp8vu) 11

Schedule/Timeline [(SER-1):](#_heading=h.3rdcrjn) 13

Required Hardware [(SER-1, EM@FSE-O):](#_heading=h.lnxbz9) 13

Third Party Content  [(SER-1, EM@FSE-O):](#_heading=h.lnxbz9) 14

Quality [(SER-2)](#_heading=h.1ksv4uv) 14

References/Source of Information [(EM@FSE-Q):](#_heading=h.44sinio) 14

Scalability [(EM@FSE-J):](#_heading=h.2jxsxqh) 15

Other Special Considerations (ABET-2[):](#_heading=h.z337ya) 15

Process15

Process Description and Justification [(SER-1):](#_heading=h.4i7ojhp) 15

Tools [(SER-1, EM@FSE-O):](#_heading=h.2xcytpi) 15

Roles and Responsibilities [(SER-1):](#_heading=h.2xcytpi) 16

Location of Project Artifacts [(SER-1):](#_heading=h.2xcytpi) 16

Sponsor Communications [(ABET-3):](#_heading=h.2xcytpi) 16

Risk Management17

Identify Potential Risks [(SER-2):](#_heading=h.2xcytpi) 17

Mitigation Strategies [(SER-2):](#_heading=h.2xcytpi) 18

# Project Description

## Overview (ABET-2):

The aim of this project is to create a web based tool that allows users to learn and experiment with simulations for sampling a metal asteroid in the NASA Psyche mission. This product aims to help users understand the possible solutions for sampling a metal-based (M-type) asteroid by allowing them to interact with a virtual environment. The system will utilize real-world examples of asteroid sampling to aid in creating a realistic simulation. NASA’s Psyche mission aims to create an engaging and accessible platform for the scientific community and for the public to simulate M-type asteroid sampling and analysis without the need to download software. NASA Psyche wants to address the lack of easily accessible tools that allow for public and more professional communities to interact with and simulate sampling scenarios. The key challenge here is to develop a browser based tool that creates an intuitive application for public consumption, allowing users to visualize, learn and interact with asteroid sampling in a meaningful way. The project will be built using web based frameworks, such as VUE and Unity, and hosted on Firebase for easy access.

## GLOBAL TRENDS (EM@FSE-e)

Education:

The core focus of this project is to provide an educational platform for the public, scientists, and those interested to engage with the NASA Psyche mission in an interactive way. By creating a web based simulator this project promotes science based education by offering a hands-on learning experience into the possibilities of space exploration.

Accessibility:

Now, more than ever before, bringing information to the public in an exciting, motivating, and informative format is a global trend and the Psyche mission is no different. Open source intelligence and knowledge sharing can bring in more of the general public to help solve problems of scale, complexity, and of public concern. This project will be public facing and freely available to experiment with, as are other aspects of the overall Psyche mission.

Energy:

By studying a planetesimal core, more information will potentially be available to answer questions about magnetism. Residual magnetic signatures can be measured and hypotheses could be formed. This could seed research that leads to breakthroughs in renewable energy production.

Data:

This project utilizes data from the Psyche mission to provide users with an accurate simulation. Making data from space exploration missions such as the NASA Psyche mission more accessible can foster a richer learning experience for the general public.

## MARKET ANALYSIS (EM@FSE-k)

There currently exists no product to facilitate scientists and engineers to visualize a possible mission to Psyche. Once Psyche has been analyzed from orbit, data collected will be used to plan future exploration and sampling missions. Rover missions historically cost over $1 billion USD to launch. This application will give prospective scientists and engineers a tool by which they can freely simulate rover and robotic missions without the financial risks associated with a live mission. Fiscal savings aside, as a publically available web application, this simulation will also serve to foster curiosity and knowledge of the general public in M-type asteroids. While videos and articles exist, this application will be a hands-on interactive option for those interested in learning more about the Psyche mission and M-type asteroids.

SECURITY CONSIDERATIONS (SER-2)

Data:

The web application will be deployed and visible to a public audience. However, no PII or any sensitive data from users of the web application or simulation widget will be stored or transmitted.

Firebase:

Deployment on Firebase will use industry standard SSL/TLS encryption for data marshaling between client and server. Content Security Policies (CSP) via Firebase can be updated as appropriate to mitigate cross-site scripting and injection attacks.

Source Code:

Source code will be produced with safety protocols in mind. This will include proper exception handling, graceful exits, and sanitizing input data on both the client and server sides as needed.

Backups:

The repository will have a backup on each of the developers local machine as well as the remote github repository. If anything fails or is altered, there will be fresh copies available.

Technology Stack:

The tools and engines to create and run the environment will utilize the most updated versions where appropriate while maintaining compatibility and interoperability.

## 

## Key Requirements (SER-2):

Functional:

* Web Based Access: The system must be fully accessible via a web browser, without the need for additional software installations.
* Simulations must operate in real time: The tool allows the user to interact with a visual simulation that can perform operations. Ensure that users can initiate and complete these simulations can provide accurate data within a realistic timeframe.
* Interactive 3D Environment: The simulation will allow users to manipulate a 3D object that are visual representations of the asteroid surface to conduct virtual samplings.
* Zoom and Pan: Users should be able to zoom in and out across the 3D model to allow more detailed exploration.

Technical:

* Imagery and Data Integration: The tool will utilize sample imagery and data to simulate this experience.
* Optimized image loading: Images shall be loaded efficiently using caching to ensure faster performance.
* Platform Compatibility: The system will be hosted on the FIrebase website.
* Resolution Scaling: System shall support higher resolution images and adjust the display according to a user’s device.

## 

## Deliverables (SER-1):

Preliminary Stage:

* Project Planning and Documentation:
  + Document the overall project scope, including identification of main use cases, actors, roles and interactions
  + Develop UML class diagrams to define core system components.
  + Develop UML Activity diagrams to represent the workflow of user interactions within the application.
* User Interface Design and Planning:
  + Devise a plan to implement a UI and specific elements that cover use cases for major use cases.
  + Develop UI prototypes for the 3D interactive simulation.
  + Refine and validate the interaction design, focusing on key user actions.
* Unity 3D Simulation Planning:
  + Create a development plan for the Unity 3D simulation, including design of the asteroid model, the logic for user interactions, and the sampling mechanisms.

Component Implementation Stage:

* Prototyping and Testing:
  + Develop a prototype of the 3D simulation using Unity and deploy it via WebGL to test basic interactions.
  + Conduct functional testing on key elements ensuring that 80% of requirements are met.
  + Integrate Unity component into the Firebase webpage.
* Unity UI Validation:
  + Validate UI/UX by ensuring that the design aligns with the functional requirements of the project.
  + Conduct internal reviews to assess the intuitiveness and responsiveness of the user interface controls.
  + Produce a demo for the sponsor.
* Document bugs and issues
  + Review system to identify bugs and functional issues
  + Document identified issues to resolve in the future
* Conduct functional tests
  + Create unit tests to resolve issues
  + Ensure functionality of code base meets expectations

Final Stage:

* Finalize project requirements
  + Review all project specifications to ensure all requirements are addressed
  + Verify implementation of features
* Finalize bug fixes and issues
  + Identify any remaining bugs found during final testing phase
  + Implement fixes to address known issues
  + Document bugs and update documentation as necessary
* Draft documentation
  + Create manuals and how to guides to explain key features and functions for general users
  + Write documentation for developers, covering the system’s architecture and code structure

## 

## Acronyms and abbreviations (ABET-3):

| Abbreviation | Description |
| --- | --- |
| M-Type | A type of asteroid mainly consisting of metals as opposed to rock material |
| VUE | A web application framework |
| Unity | A gaming platform engine |
| CSP | Firebase’s *Content Security Policy* allows developers to make specify various security concerns such as forced HTTPS |
| Psyche | This could describe both the name of the asteroid or the name of the orbiter |
| Repo | The main GitHub repository where the web application and its source code will be stored |
| US | User story from the taiga board |
| main | The public facing git branch where final source code will be integrated into for deployment |
| dev | The developers git branch where all reviewed code will be merged into |
| IDE | Integrated Development Environment |
| RTOS | Real Time Operating System |

# 

# Design and Architecture

### Design Description (ABET-1, ABET-2):

User Interface (UI) Component:

The UI component is responsible for managing the presentation layer of the simulation. It will serve as the interface through which users interact with the system as well as the “View” section of the MVC design pattern.

Areas of Functionality:

* Dashboard: Displays the simulation status and available tools, such as sample collection devices.
* Control Panel: Provides buttons and inputs for starting, pausing, and interacting with the simulation.
* Feedback Display: Shows results of sampling actions (e.g. material type, quantity collected)
* Responsive Design: Ensures the simulation works well on a variety of screen sizes and devices.

Simulation Engine Component (Unity WebGL):

The simulation engine is the core module responsible for handling the logic, physics, and mechanics of the surface sampling process on the asteroid. Unity, with a WebGL build, will be the engine running the simulation on the web. It will act as the “Model” in the MVC pattern.

Areas of Functionality:

* Asteroid Model: Represents the physical characteristics of the M-Type asteroid, including its metal/rock composition and surface conditions.
* Samping Mechanics: Manages the interaction between the user-controlled sampling tool and the asteroid surface, determining the outcomes such as material type and quantity collected.
* Physics Simulation: Handles gravity, collisions, and other forces involved in sampling on a low-gravity, metal-rich asteroid.
* Randomization: Ensures the distribution of materials on the asteroid surface varies per simulation, adding unpredictability to the sampling process.

Data Management Component:

The data management component handles the saving, retrieval, and processing of both user interaction data and simulation results. This component is responsible for maintaining a history of user actions and results, as well as ensuring data consistency. It will also aid the “Model” in the MVC pattern.

Areas of Functionality:

* User Session Data: Tracks individual user interactions, such as number of samples taken, material composition, and timestamps.
* Simulation State Storage: Saves simulation progress and the state of the environment, allowing users to resume sessions or analyze past results.
* Result Logging: Stores results of sampling activities for potential analysis or reporting to scientific communities.

External Systems Integration Component:

This component handles the interaction with external systems, specifically Firebase for web hosting and deployment. It will serve as the “Controller” portion of the MVC model.

Areas of Functionality:

* Firebase Hosting: Manages the deployment of the web-based simulation, ensuring it’s available for public use.
* Real-time Database: Stores and retrieves session data, ensuring that user interactions and results are preserved across multiple sessions.

### Alternate Design Possibilities (EM@FSE-b):

A design alternative the team considered was component-based design with a framework (e.g. Phaser) instead of a game engine. The proposed design was decided on instead due to the plethora of Unity built games on Nasa’s Psyche website, as well as the sponsor providing guides on how to work with Unity. In addition, each component in the proposed design can be developed, tested, and maintained independently. The system can also grow and evolve as requirements change fairly easily.

| Criteria | Proposed Design | Game Framework |
| --- | --- | --- |
| Modularity | High | Medium |
| Scalability | High | Medium |
| Development Complexity | Medium | Medium |
| Performance | High | High |
| Visual Quality | High | Medium |
| Learning Curve | Medium | Low |
| Deployment Complexity | Medium | Medium |
| Community Support | High | High |
| Cost of Changes | Low | Medium |

# Implementation Strategy

## High-level Work Breakdown Structure (SER-1):

Preliminary Stage:

* Document the Project Scope
  + Work Scope: Identify and document main use cases, actors, use cases, roles, and interactions for the 3D simulation.
  + Skillset: Technical writing, Communication, Business Analysis
  + Time: 1 day
* Develop UML Diagrams
  + Work Scope: Create UML Diagrams to define the system’s core components.
  + Skillset: Proficiency in UML creation, object oriented design, workflow modeling
  + Time: 5 days
* Create Functional UI Mockups
  + Work Scope: Design and implement functional UI mockups within the Unity environment.
  + Skillset: UI design, prototyping tools
  + Time: 6 days

Component Implementation Stage:

* Develop Unity Simulation Prototype
  + Work Scope: Build and develop Unity prototype to be deployed using Unity WebGL
  + Skillset: Familiarity with Unity, understanding of C#
  + Time: 2 week
* Conduct testing
  + Work scope: Perform testing of the 3D simulation, ensuring it meets 80% of functional requirements.
  + Skillset: C#, debugging skills, reading stack traces
  + Time: 1 week
* Integrate Unity Component into Firebase
  + Work scope: Integrate the Unity 3D simulation into the firebase webpage
  + Skillset: WebGL integration, web page development
  + Time: 5 days
* Validate UI and Functional Design
  + Work scope: Ensure that the design aligns with functional requirements of the project
  + Skillset: C#, understanding of unit testing, unity experience
  + Time: 4 days
* Produce demo for sponsors
  + Work scope: Create a demo to showcase the functionality of the simulation. Present to sponsor for feedback.
  + Skillset: Public speaking skills, presentation skills
  + Time: 3 days
* Identify and Document known issues
  + Work scope: Review the system to identify and document known issues

Final Stage

* Finalize project requirements
  + Work scope: Ensure that all the project requirements are met
  + Skillset: Unity development, functional testing
  + Time: 1 week
* Finalize bug fixes and issues
  + Work scope: Identify and resolve any remaining bugs and issues
  + Skillset: Debugging skills, unity development (if necessary)
* Create documentation
  + Work scope: Write detailed documentation, including how to guides and technical documentation
  + Skillset: Technical writing skills, Understanding of documenting code bases
  + Time: 3 days

## 

## Schedule / Timeline (SER-1):

Semester 1:

* Sponsor Meeting 10/08/2024 at 4:30pm
* Artifact Review: YouTube Video Week of 10/21/2024
* Sponsor Meeting 10/22/2024 at 4:30pm
* Artifact Review: YouTube Video Week of 11/04/2024
* Sponsor Meeting 11/05/2024 at 4:30pm
* Artifact Review: YouTube Video Week of 11/18/2024
* Sponsor Meeting 11/19/2024 at 4:30pm
* Artifact Review: YouTube Video Week of 11/18/2024
* Unity Game Widget Milestone: Week of 11/25/2024
* Sponsor Meeting 12/02/2024 at 4:30pm
* Final Session 1 Deliverables 12/06/2024
* Winter Break 12/06/2024
  + Independent work on winter break sprint

Semester 2:

* Spring Session Begins 01/13/2025
* Sponsor Meeting 01/14/2025 at 4:30pm
* Artifact Review: YouTube Video Week of 01/20/2025
* Sponsor Meeting 01/28/2025 at 4:30pm
* Artifact Review: YouTube Video Week of 02/03/2025
* Web Application Milestone: Week of 02/10/2025
* Sponsor Meeting 02/11/2025 at 4:30pm
* Artifact Review: YouTube Video Week of 02/17/2025
* Sponsor Meeting 02/25/2025 at 4:30pm
* Artifact Review: YouTube Video Week of 03/03/2025
* Sponsor Meeting 03/11/2025 at 4:30pm
* Web App Integration Milestone: Week of 03/17/2025
* Artifact Review: YouTube Video Week of 03/17/2025
* Sponsor Meeting 03/25/2025 at 4:30pm
* Artifact Review: YouTube Video Week of 03/31/2025
* Sponsor Meeting 04/08/2025 at 4:30pm
* Artifact Review: YouTube Video Week of 04/14/2025
* Sponsor Meeting 04/22/2025 at 4:30pm
* Final Capstone Deliverables: 05/02/2025
* End Capstone 05/02/2025

## Required Hardware (SER-1, EM@FSE-o):

* Laptop or Desktop computer for each developer
* Server space allocated for deployment on Firebase
* Minimum of 8 gigabytes of ram for gaming engine development

## Third party content (SER-1, EM@FSE-o):

Images:

Images for the web application may need to be pulled from several sources to fully inform the users about potentially sampling an M-Type asteroid.

Audio:

Custom audio may need to be created for the game. Royalty free music for non-commercial use could also be used.

## Quality (SER-2):

Keeping code neat will be a necessary component of this project. This can be measured by using comments for each section of code to indicate the purpose of the segment. Additionally, using consistent and appropriate indentation will be helpful. Of course at a minimum, all basic requirements the sponsor has given us must be met and additional things are a plus one those base level requirements are met. Errors that affect the requirements must be addressed primarily and then warnings and less significant errors can be addressed if time permits.

## References/Sources of Information (EM@FSE-q):

Physics Resources:

[Gravity Applications](http://www2.phy.ilstu.edu/~bkc/astronomy/gravappl/gravapplb.htm#:~:text=At%20the%20same%20distance%2C%20a,therefore%2C%20orbit%20at%20faster%20speeds)

[Impact of Mass on Orbital Speed - Khan Academy](https://www.khanacademy.org/science/mechanics-essentials/xafb2c8d81b6e70e3:how-to-make-the-earth-a-black-hole/xafb2c8d81b6e70e3:are-all-of-saturn-s-rings-moving-at-the-same-speed/v/impact-of-mass-on-orbital-speed)

Geological Resources:

[Magnetic Reversals](https://www.youtube.com/watch?v=BCzCmldiaWQ)

Previous Mission Examples:

[Osiris Rex](https://www.youtube.com/watch?v=42EwbQ3afPA&ab_channel=NASAGoddard)

[More Osiris Rex](https://www.youtube.com/watch?v=xj0O-fLSV7c&ab_channel=NASAGoddard)

[NASA Osiris Rex Source](https://science.nasa.gov/mission/osiris-rex/)

Unity:

[User Manual](https://docs.unity3d.com/Manual/index.html)

[Unity Tutorial for Beginners](https://www.youtube.com/watch?v=XtQMytORBmM&t=2109s)

VUE:

[VUE Documentation](https://vuejs.org/)

## Scalability (EM@FSE-J):

The simulation the project aims to create can garner support from the masses by showcasing the mission before it actually happens. The additional support could bring more investors which would allow us to scale up if necessary or desired. Something to be considered is that we do not want the visuals to be too demanding for an average computer since it will likely be 3d rendering. The way this can be reduced/addressed is by avoiding using super demanding technology (unreal engine 5 for example).

## 

## Other Special considerations (ABET-2):

Interoperability with NASA’s Psyche website was a major consideration when deciding on the design of the project. It was important that the project operate well with the current format Psyche had in place. The project design was informed by the games already hosted on their site.

# process

## Process Description and justification (SER-1):

* The application will be developed with the Agile Scrum methodology with weekly stand-ups on Monday mornings.
  + Monday stand-ups have been chosen because of team member availability, with asynchronous communication all other times during the week.
* Taiga will be used to manage the Scrum board and track burn downs.
* Application code will be uploaded and maintained in a GitHub repository with each member maintaining their own branch to be merged into Dev.
  + Team members shall have their code reviewed before it is merged into Dev for quality control and testing.
* Every two weeks the team will meet with the sponsor to share progress.

## Tools (SER-1, EM@FSE-O):

* IDE: each developer can use a preferred IDE such as Eclipse, IntelliJ, VSCode, etc.
* Gradle: a build tool for projects
* VUE: a framework for designing web applications
* GitHub: [click here](https://github.com/MissionToPsyche-Iridium/iridium_22d_m-type_sim-se) : location of the remote repository will be kept
* Git: version control system
* Docker: a container for the project
* Taiga: [click here](https://tree.taiga.io/project/madd-ei-group-31-nasa-psyche-mission-m-type-asteroid-sampling-simulator/timeline) : a scrum board to stay on task
* Google Drive [click here](https://drive.google.com/drive/folders/1C13MSVplNpeHRB1AB5qPD9AZS_jwbd8D?usp=drive_link): a place where resources, notes, documents, and assets will be stored

## Roles and Responsibilities (SER-1):

* Product Owner: reviews, maintains and plays role in the creation of the backlog and prioritizes its tasks for sprints and implementations
* Scrum Master: maintains scrum process and acts as coach throughout a sprint cycle
* Developer: implements user stories with source code, artifacts, or other necessary mediums
* Communications Lead: acts as the lead communicator between the sponsor and the team

## Location of Project Artifacts (SER-1):

| [Sponsor Google Drive Folder](https://drive.google.com/drive/folders/1sHy2kvcDx4lFAaYkeBb4fHFhTXwib6ft?usp=drive_link) | [Team Google Drive Folder](https://drive.google.com/drive/folders/1C13MSVplNpeHRB1AB5qPD9AZS_jwbd8D?usp=drive_link) | [GitHub](https://github.com/MissionToPsyche-Iridium/iridium_22d_m-type_sim-se) |
| --- | --- | --- |

The Sponsor Google Drive will hold all artifacts provided and managed by the sponsor. This includes research papers, guidelines, and other various resources. The Team Google Drive will include any documents created by the team. It is structured with folders for process, reports, communication, and design. Folders will be added as needed for further clarity on any new type of artifacts created. Google drive is useful due to the ease of access it provides to all users with access to the folders. It is also notoriously reliable and provides its own, native version control. GitHub will be used to store the codebase. Project iterations will be easy to find and run due to version control using Git. The team has experience with both GitHub and Git, so no research will need to be done to aid its use amongst members. GitHub and Git are both considered to be reliable services. Their popularity serves as evidence to this fact.

## 

## Sponsor communications (ABEt-3):

Synchronous Communication:

Sponsor meetings are held once every two weeks via Zoom.

Asynchronous Communication:

Asynchronous follow-up questions are communicated through a designated slack channel on an as-needed basis. Standup meetings are conducted every Monday and will recap the progress individuals have made.

# 

# Risk management

## 

## identified Potential risks (SER-2):

Risk 1: Limited Storage Space

Storage space may be limited if the gaming engine requires storing large amounts player’s data for scoreboards and statefulness

Incident Rate: Per interaction after deployment

Risk 2: Asset Storage

Assets need to be hosted in proper buckets for linkable public access. Host assets on Google Drive could lead to CORS issues

Incident Rate: Several times throughout development

Risk 3: Game Widget Integration

Creating a gaming widget that works as designed and integrates as a widget into a web application could fail

Incident Rate: Once, at time of component integration

Risk 4: Firebase Deployment

A completed project may run into issues making itself visible to the public if deployment process has complications

Incident Rate: Once, at the end of project

Risk 5: Scheduling

Times that work well for the entire team for asynchronous meetings are limited.

Incident Rate: Weekly

## 

## mitigation strategies (SER-2):

Risk 1: Limited Storage Space

Approach: Storing user data using JSON files and Unity’s PlayerPrefs. Also avoiding keeping large amounts of data in memory by saving and loading it as needed.

Cost: A minor increase in development time to implement serialization techniques.

Risk 2: Asset Storage

Approach: Utilizing Firebase hosting to store and serve assets as well as configuration of CORS rules.

Cost: No cost. Firebase is free and scalable. Setup time is minimal.

Risk 3: Game Widget Integration

Approach: Export the project using Unity WebGL and embed it in the website using iframe.

Cost: Minimal additional development time. Slight effort required to test iframe responsiveness and integration with the web app’s design.

Risk 4: Firebase Deployment

Approach: Use Continuous Deployment (CD) through Firebase Hosting’s preview channels. Integrate with a CI/CD pipeline using GitHub Actions.

Cost: Low setup cost. Regular maintenance on tests required.

Risk 5: Scheduling

Approach: Schedule regular meetings and check-ins in advance. Stand Up every Monday, Sponsor meeting every 2 weeks, etc. Communicate asynchronously on a frequent basis.

Cost: Minimal impact if managed well, but possible delays in the event of a communication breakdown.