

Project Advisor：[Dr Nadeem Majeed]

Name：[Usman Ghias, Usman Zahid, Qurban Hanif, Kainat, Alisha Sultan]

Roll Number：[bsef20m540, bsef20m534, BSIt]

Submission Date：[September 27, 2023]

Final Year Project

[Auto Pilot]

**Project Proposal Auto Pilot**

Table of Contents

Catalog

[1.1 Project Title 2](#_Toc146665023)

[1.2 Project Overview Statement 2](#_Toc146665024)

[1.3 High-level system components 5](#_Toc146665025)

[1.4 High Level Use Case Diagram 5](#_Toc146665026)

[1.5 List of optional functional units 5](#_Toc146665027)

[1.6 Exclusions 5](#_Toc146665028)

[1.7 Application Architecture 5](#_Toc146665029)

[1.9 Gantt chart 6](#_Toc146665030)

[Timeline 6](#_Toc146665031)

[Project Kanban View 7](#_Toc146665032)

[Gant View 7](#_Toc146665033)

[1.10 Hardware and Software Specification 8](#_Toc146665034)

[1.11 Tools and technologies used with reasoning 8](#_Toc146665035)

[Tools 8](#_Toc146665036)

[**Technology** 9](#_Toc146665037)

# 1.1 Project Title

Auto Pilot

# 1.2 Project Overview Statement

In a world where climate change looms as an urgent global crisis, our project, Deep learning-based framework for monitoring of debris-covered glacier from remotely sense images emerges as a pivotal endeavor. Escalating pace of glacier melting brings forth dire consequences, including the rise in sea levels and heightened risk of Glacier Lake Outburst Floods (GLOF). In an era where environmental preservation stands as a top priority, our project seeks to revolutionize the monitoring and classification of debris-covered glaciers, offering both precision and accessibility through our website and mobile application. Focused on debris-covered glaciers, we employ advanced convolutional neural networks (CNNs) to enhance monitoring precision. This dual approach enhances classification accuracy and ensures robust feature extraction.

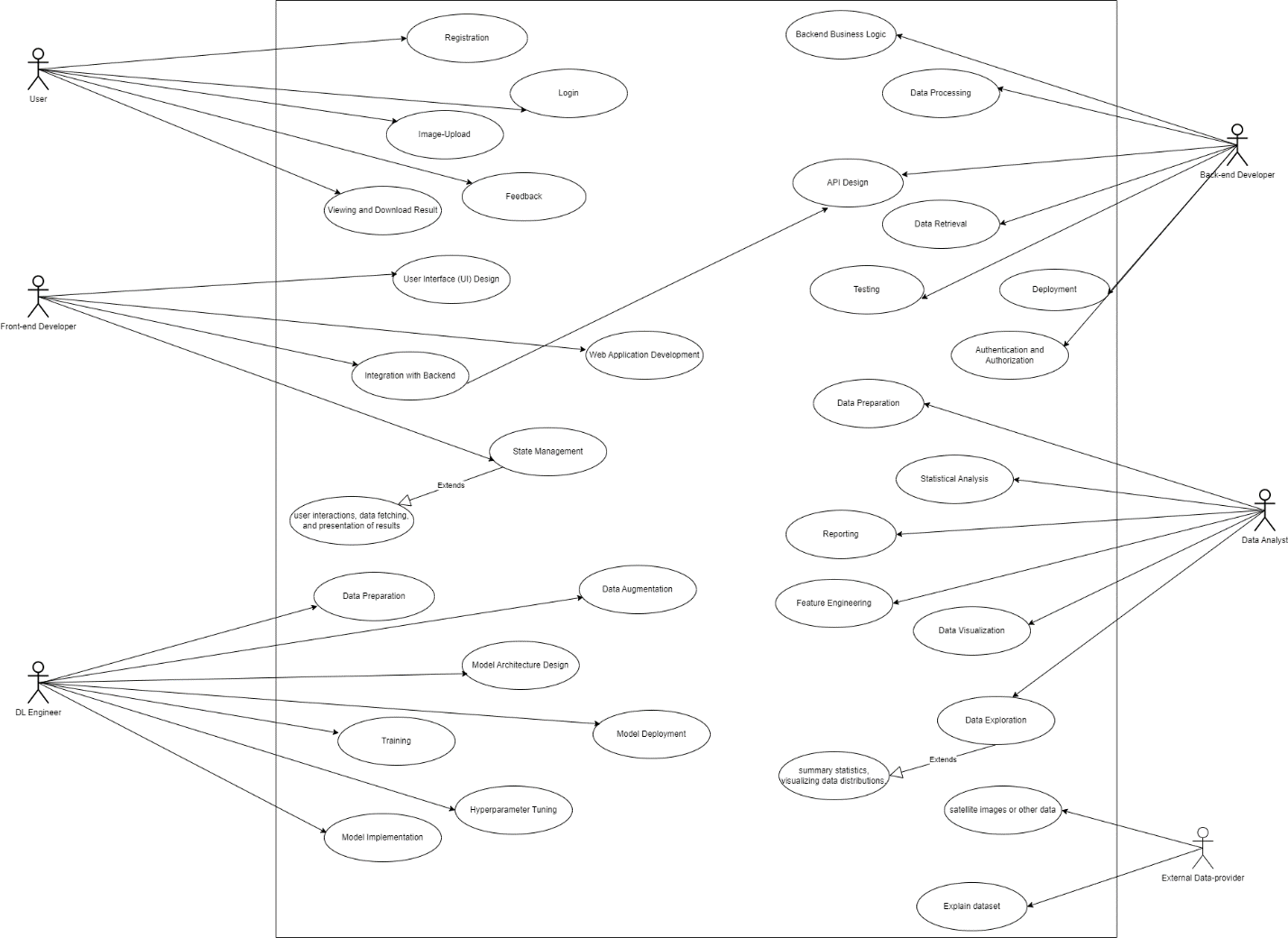
By generating missing data classes with generative adversarial networks (GANs) and optimizing our architecture, we pave the way for large-scale glacier mapping using freely available dataset. Additionally, our project extends to the development of a user-friendly web and mobile application, offering accessibility for on-the-go glacier information access. Furthermore, we commit to assessing our deep learning-based approach's performance by conducting a comprehensive comparative analysis with traditional machine learning methods, including Support Vector Machine (SVM), Random Forest (RF). In the ongoing battle against climate change, our project assumes a vital role, serving as a potent tool for comprehending glacier dynamics, promoting responsible environmental stewardship, and ensuring that critical information is readily available to researchers and stakeholders worldwide.

|  |  |  |
| --- | --- | --- |
| **Project Title:** Automated Deep Learning-based Framework for monitoring of Debris-Covered Glaciers from Remotely Sensed Images | | |
| **Group Leader:** Muhammad Usman Ghias | | |
| **Project Members:**   |  |  |  |  | | --- | --- | --- | --- | | Name | Roll No. | Email Address | Signature | | M. Usman Ghias | BSEF20M540 | bsef20m540@pucit.edu.pk | *UsmanGhias* | | Qurban Hanif | BSEF20M534 | bsef20m534@pucit.edu.pk | *Qurban* | | Aleesha Sultan | BSITF20A539 | Bsitf20a539@pucit.edu.pk |  | | Usman Zahid | BSITF20A | Bsitf20a |  | | Kainat | BSITF20A50 | Bsitf20a503@pucit.edu.pk |  | | | |
| **Project Goal:** Primary goal of Auto Pilot is to create web and mobile based application for show case our deep learning-based framework for monitoring of debris covered glaciers. This platform will empower researchers to classify their images in a professional, online environment, bridging the gap between academic exploration and real-world applications in glacier research. Our model will be able to identify glacier debris automatically by just uploading image which is done manually till now in space and science. Presenting  This project will focus on several key strategies:   * Develop a versatile and adaptable deep learning framework capable of seamlessly integrating new functionalities and adjustments. * Engineer a system with inherent ease of maintenance and scalability, achieved through modular code and reusability. * Prioritize codebase testability to streamline debugging efforts and elevate overall system quality. * Simplify database interactions and enhance codebase decoupling through the implementation of Object-Relational Mapping (ORM). * Create **consistent**, easy-to-use, and reliable interfaces for interacting with the system for internal and external components and users. * Properly working APIs for Mobile and Web Application | | |
| Objectives:   |  |  | | --- | --- | | Sr.# |  | | 1 | **Improve code maintainability and reusability**: Utilize object-oriented design patterns to create modular and reusable code that is easily maintainable, making it easier to evolve the system over time. | | 2 | **Develop a robust and extensible object-oriented architecture:** Apply the principles of object-oriented analysis and design to create a flexible and scalable system that can accommodate new features and changes in the future. | | 3 | **Enhance code testability:** Implement techniques such as dependency injection and mock objects to increase the ease of testing the codebase and reducing the effort required for bug fixing. | | 4 | **Responsive Design:** People prefer to visit from mobile application instead from web application, so our motive is to make website responsive for debris-covered glaciers. | | 5 | **Deep Learning Framework:** A lot of deep learning models are available online; we will prefer to use open source and freely available resource for our deep learning frameworks. | | 6 | **Implement Object-Relational Mapping (ORM):** Use an ORM framework like Hibernate or **Entity** Framework, to ease the interaction with the relational database and make the codebase more decoupled from the database | | 7 | **Improve scalability and performance:** Design the system to handle increased usage and take measures to improve the performance and response times from our AI model. | | | |
| **Project Success criteria**:  The success of our project will be determined by several critical factors. First, achieving a detection accuracy rate of over **90%** through deep learning models will be a primary indicator. And other measurable criteria such as a **flexible**, **extensible** architecture of web application; **maintainable**, understandable code-base integration; **efficient**, **effective** testing of deep learning model, mobile and web app; successful implementation of ORM and **DDD** (Domain Driven Design); Improved scalability and performance, and user-friendly interfaces for uploading images. Finally, optimizing efficiency, reducing research time, and demonstrating cost-effectiveness in project execution and environmental impact mitigation will round out the success criteria. |
| **Assumptions**   * Internet connectivity * Compatible device and browser or mobile operating system * Access to reliable upload satellite imagery and glacier raster data. * User adoption and engagement. * Basic understanding of how to use the internet and navigate a website/mobile app. * Updated version of the Application/Website * Have little bit background knowledge of GIS. * Must be able to differentiate colors.   **Risks and Obstacles**  Risks and obstacles when accessing the Amazon Decor mobile **application** or **website**:   * Copyright and licensing issues with data sources. * Technical difficulties * Compatibility issues * Network connectivity * Data privacy concerns. * Technical complexities in Algorithm understanding. * Confusing or difficult to understand raster and vector data. * Unreliable or slow customer service. |
| **Organization Address**: Punjab University College of Information Technology(**PUCIT**), University of The Punjab, Allama Iqbal Campus, Shahrah-e-Quaid-e-Azam (The Mall), 54000, **Lahore**, Pakistan |
| **Type of project:** *Research*  *Development* |
| **Target End users:**   1. Climate Researchers and Scientists 2. Environmental Organizations 3. Government Agencies 4. Glacier Monitoring Agencies 5. Educational Institutions |
| **Development Technology:** Object Oriented Approach |
| **Platform:**  Web Application (also mapped to mobile application) |
| **Suggested Project Supervisor:** Sir Nadeem Majeed |
| **Approved By:**  Sir Nadeem Majeed |
| **Date:** September 26, 2023 |

# 1.3 High-level system components

* Authentication Module
* Deep Learning Module
* Uploading Raster

1.4 High Level Use Case Diagram



# 1.5 List of optional functional units

* Chatbot Integration
* Feedback module
* Social Media Plugin

# 1**.6 Exclusions**

There will be no exclusions in our project.

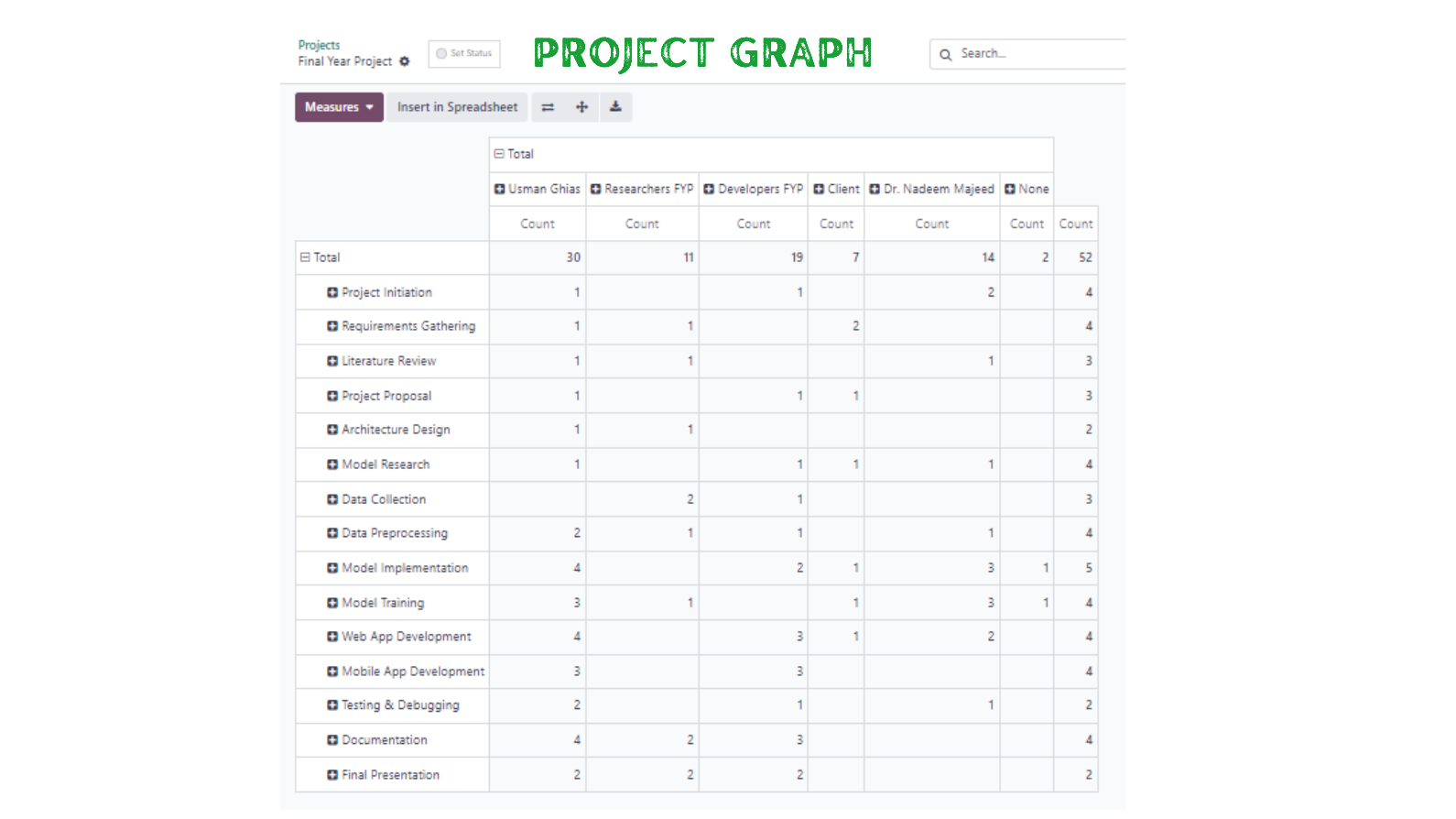
# 1.7 Application Architecture

We will follow 3-tier Architecture.

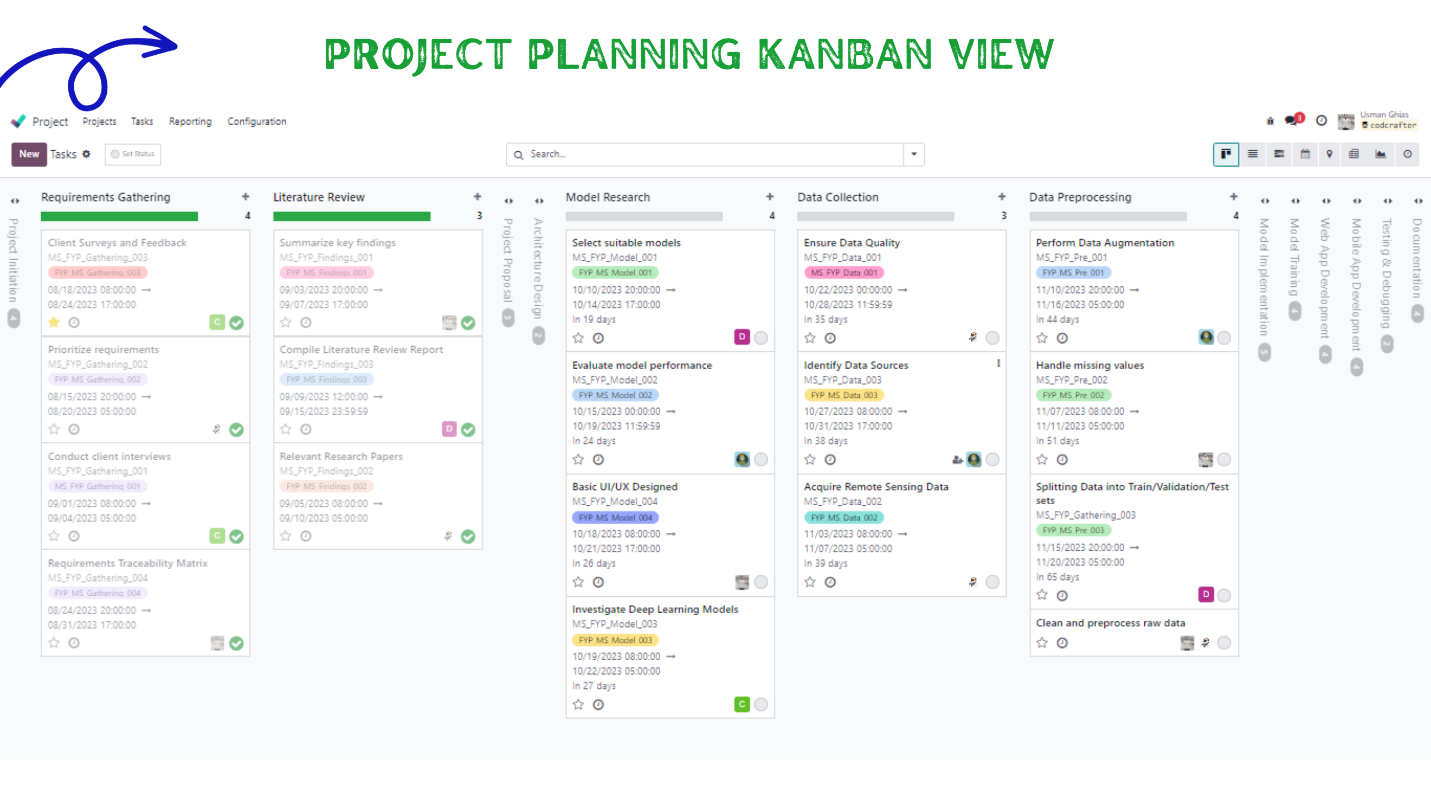
* React Native (Front End)
* Python (Back End)
* Postgres (Database)

# 1.9 Gantt chart

## Timeline



## Project Kanban View



## Gant View

A screenshot of a project planning chart

Description automatically generated

## 1.10 Hardware and Software Specification

#### Hardware: (Basic)

* Intel Core i7-5900 or with 2.5 GHz
* 16 GB of DDR3 RAM
* 256 GB SSD
* Graphical Processing Unit (GPU)

**Network Connectivity:**

Ensure you have a stable and high-speed internet connection, especially if your project involves downloading large geospatial datasets or working with cloud-based resources.

#### Hardware: (Recommended)

* Intel Core i7-9700 processor with 3.00 GHz or more
* 16 GB of DDR4 RAM, and an NVIDIA GeForce GTX
* 1080 graphical processing unit (GPU)

We will be using the recommended hardware for our training our model on deep learning frameworks that require significant GPU acceleration, you might consider adding one or more powerful GPUs, such as NVIDIA RTX series or NVIDIA A100 GPUs, to increase model training speed.

#### Software: (Recommended)

* Chrome
* Windows 10
* MacOS
* Linux 20.04 LTS
* Android 9+
* IOS 12+

## 1.11 Tools and technologies used with reasoning

### Tools

* **Google Earth Engine:** For data acquisition, processing and analysis of remote sensing data related to glaciers.
* **Erdas Imagine:** A powerful remote sensing and image processing software for working with geospatial data.
* **ArcGIS**: A comprehensive GIS platform for geospatial data management, analysis, and visualization.
* **Visual Studio Code:** A popular code editor for developing frontend and backend applications, including React Native, Python, and Django.
* **Android Studio:** The official IDE for Android app development, used for creating and testing the Android version of your mobile application.
* **Web GIS:** A generic term that encompasses various web-based Geographic Information System (GIS) tools and technologies, which you might use for web-based mapping and visualization.
* **Google Colab:** For collaborative coding, data analysis, and machine learning using Jupyter notebooks in the cloud.
* **Version Control:** Tools like Git and platforms like GitHub, GitLab or GitBucket can help manage code versions and collaborate with others.

### **Technology**

**Web Frontend**

* **Responsive Web Design:** Ensuring that frontend is responsive to various screen sizes and devices. We will be using CSS frameworks like bootstrap or CSS Grid for responsive design.
* **JavaScript:** JavaScript is a must-have for adding interactivity to your web application. You can use it to create dynamic maps, interactive charts, and user-driven features.
* **React**: A JavaScript library for building user interfaces. You will need Node.js and npm (Node Package Manager) for managing React applications. React allows you to create reusable UI components, which is helpful for building complex mapping interfaces. React’s virtual DOM minimizes page re-renders, optimizing performance when handling large datasets.
* **Tensor Flow:**
* **Pytorch**

**Web Backend**

* **Django Rest Framework:** For creating a RESTful API to serve data and interact with the frontend.
* **Node JS**

**Database Connectivity:**

* **MySQL**: A relational database management system (RDBMS) for storing and managing your project’s data. You will need to install MySQL Server and MySQL Workbench (a graphical tool for managing databases).
* **MongoDB:** MongoDB’s flexible document-based model allows you to store data in JSON-like documents, making it easy to represent geospatial information such as coordinates, timestamps, and associated metadata, making it suitable for storing and managing geospatial data associated with debris-covered glaciers.

**Python Libraries**

Matplotlib

2. Cartopy

3. Basemap

4. Mplleaflet

5. Follium

6. Geopandas

7. Shapely

8. Raster.io