Development Plan
Computer Science Capstone Project
AI for Chest X-Ray read

Group Number 15

Supervisor:
Name: Dr. Mehdi Moradi
Email: moradm4@mcmaster.ca
Organization: McMaster University, CAS

Team Members:
Name: Suhaas Parcha
Email: parchas@mcmaster.ca

Name: Suvansh Dutt
Email: dutts1@mcmaster.ca

Name: Yuvraj Singh Sandhu
Email: sandhy1@mcmaster.ca

Name: Ujjwal Raj
Email: raju1@mcmaster.ca

October 25, 2024

Contents

Revision History

Date	Developer(s)	Change
23rd October 2024	Suhaas, Yuvraj	Started Initial draft and created the template.
24th October 2024	Yuvraj, Ujjwal, Suvansh	Finalized V0.
15th February 2025	Suhaas, Yuvraj Ujjwal, Suvansh	Updated requirements
3rd April 2025	Suhaas, Yuvraj Ujjwal, Suvansh	Final changes

1 Team Meetings Plan

Virtual Meetings:

- Platform: Discord Voice Channels.
- Frequency: Bi-weekly scheduled meetings with additional meetings as needed.
- Agenda: Shared in advance via Teams, Discord & WhatsApp to ensure all members are prepared.
- Document key points and action items from each meeting in a text file on github.

2 Team Communication Plan

Comunication platforms:

1. **Microsoft Teams**
   - Purpose: Communication with our Supervisor, Teaching Assistant
   - Usage: Seek assistance from the supervisor and teaching assistants. By inviting them to the dedicated project team, members can easily pose questions, request feedback, and receive guidance on project-related matters.
2. **Whatsapp Group**
   - Purpose: Quick updates, informal communication, and urgent notifications.
   - Usage: Share brief messages, reminders about meetings, and immediate questions or updates.
3. **Discord Channel**
   - Purpose: Central hub for collaboration, virtual meetings, and detailed discussions.
   - Usage: Hold virtual meetings, share documents and resources that might be useful for this project

Project Management Tools:

- Github Issues:
  - Purpose: Track tasks (Completed/ On-going), bugs, and feature requests.
  - Usage: Create issues for each task or bug with detailed descriptions; assign issues to team members and set labels for prioritization; use milestones to group related issues and track progress towards major project goals.

3 Team Member Roles

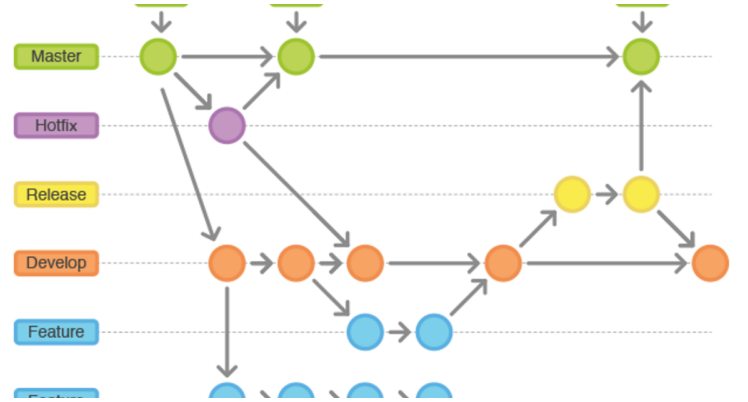
Team Member	Mac Id	Role
Suhaas Parcha	parchas	Frontend and Backend
Suvansh Dutt	dutts1	Frontend and Backend
Yuvraj Singh Sandhu	sandhy1	Frontend and Backend
Ujjwal Raj	raju1	Frontend and Backend

To maximize collaboration and mitigate the risk of errors, every team member will be encouraged to contribute to all aspects of the project. This approach fosters a shared understanding of the project and enhances collective problem-solving.

4 Workflow Plan

4.1 Workflow using software

- The team will use "Git Flow", requiring separate branches for each feature or bug fix, peer reviews via pull requests, and prohibiting direct commits to the main branch.



- The management of the model will be accomplished by utilizing the structure of github/git
- Each member works on a separate branch name after their feature or task.
- Once a task is completed, the members creates a pull request to merge the branch into origin.
- In case of merge conflicts, a reviewer will fix the code and eventually accept the pull request to merge the branch into main. For any merge request to be full-filled, it requires at least 1 reviewer.
- Members will not have access to directly merge into main. The reviewing process is very important.
- Quality of life changes or designs are communicated and changed upon approval of the group

- Git comments and issues are utilized for when the main directly has a bug
- The bug vary in different levels of urgency where higher the level, the more important it is. The levels are determined based on the complexity of the issue caused.
- In case of an unfixable high level bug, the owner of the repository can rollback the project to a stable version.
- We use a standardized issue template to ensure consistency. Each issue/pull request includes important details such as a description, expected behavior, actual behavior, and steps to reproduce (for bugs), or a clear explanation of the feature/enhancement request.

4.2 Issues solved with group dynamics

- **Communication Gaps**: In order to avoid occasional communication delays, especially during decision-making. We implemented regular stand-up meetings and adopted Discord and Whatsapp as our main communication tool to ensure quick and clear communication across the team.
- **Majority Voting System**: To handle issues and new ideas, we implemented a majority voting system. Each team member has one vote, and decisions are made based on the majority outcome. This ensures that all voices are heard and considered. In case of conflicts or unresolved issues, we escalate the matter to our supervisor. However, conflicts are rare as the team maintains a level-headed and understanding approach to problem-solving.
- **Idea Proposal and Voting**: When a team member comes up with a new idea or enhancement, it is proposed to the group and put through a voting process. Each team member casts one vote, and if the idea is accepted by majority vote, it is integrated into the project plan. This democratic process ensures that new ideas are evaluated fairly and collectively.
- **Quality Assurance**: Some errors initially slipped through due to a lack of formal review. To improve this, we introduced a policy that requires at least one team member to review and approve all pull requests before merging, which has significantly improved code quality.

5 Proof of Concept Demonstration Plan

5.1 Risks

The primary risks to the success of our project are:

- **AI Model Accuracy**: There is a risk that the AI model may not achieve the necessary accuracy in detecting diseases from chest X-rays. Poor accuracy could lead to false positives or false negatives, undermining the system's reliability.

- **Data Security Compliance**: Given the sensitive nature of medical data, there is a risk of not handling the encryption of users and uploads in a secure manner. Ensuring secure handling, encryption, and access control for patient data is crucial.
- **User Interface Usability**: The user interface might not be intuitive enough for general users who may have limited technical skills. Poor usability could hinder the adoption of the system by target users.

5.2 What Will Be Demonstrated

To mitigate the above risks, the proof of concept will demonstrate the following:

- **AI Model Accuracy**: We will demonstrate the AI model's ability to detect diseases with an acceptable level of accuracy post testing. The accuracy would depict the low number of false positives, false negatives, true negatives to be at an acceptable level of an early stage product.
- **Frontend Development:** We will showcase a functional front-end interface that simulates the user experience of the complete system. This demonstration will include mock-ups of key features. While not connected to the back-end during this proof of concept, the front-end will illustrate the intended user flow and interface design, it will provide with a clear vision of the final product's look and feel.
- **User Interface Usability**: We will include a segment which goes over our design choices, asset placement decisions and our reasoning as to how it can help make the interface more intuitive and usable for an average user with no technical experience. Including key design characteristics which are crucial in creating a good user interface.

6 Technology

- Specific programming language - Python will be used to train and maintain the back-end, while the front-end would be maintained using Svelte.
- Specific unit testing framework -unittest Python, Git Automated Unit test.

  We will in future be using the testing framework to reduce the load on ourselves, It allows us to check if there any compilation errors.
- Libraries we will likely be using - Pytorch would be used to train the model, Numpy, Pandas for data manipulation, seaborn/matplotlib for visualisation, Svelte framework for our website, Tailwind CSS for designing/animations.
- Tools we will likely be using -VsCode, Webstorm, Git for version control and project features addition, Discord and Whatsapp for team communication, Figma for designing.
- Computational Power: Access to Nvidia RTX 4090. we are considering renting a server equipped with more GPU power in the future if our project's demands does not meet, ensuring scalability and access to high-performance hardware as needed.

7 Coding Standard

- **Version Control and Branching:** Every new feature or bug fix must be developed in a separate Git branch. Branch names should reflect the task, e.g., 'feature-login', 'bugfix-upload'.
  All code must be committed with meaningful messages that clearly explain the changes, such as 'Added validation for login form' or 'Fixed memory leak in file upload module'.

- **Code Review:** All code must undergo a peer review process before being merged into the main branch. At least one team member must approve the pull request.
  Reviewers are expected to check the code for readability, logic, potential bugs, and adherence to coding standards.

- **Formatting/linting**: -We are all using the same formatter/linter for our python/javascript code.

- **Code Modularization:** We are going to split our work into modules, each one responsible for doing a core/essential task.

- **Error Handling:** We will implement error handling to manage exceptions gracefully. User will be provided with a detailed error message if encountered.

- **Dynamic Code and theme**: Code changes are minimized, as updates to values can be made in a single location. All configurations, constants, and settings should be stored in configuration files or environment variables. Use dynamic variables to ensure flexibility and adaptability across different environments. It Supports easier adaptation of the software to new requirements or environments.

8 Project Scheduling

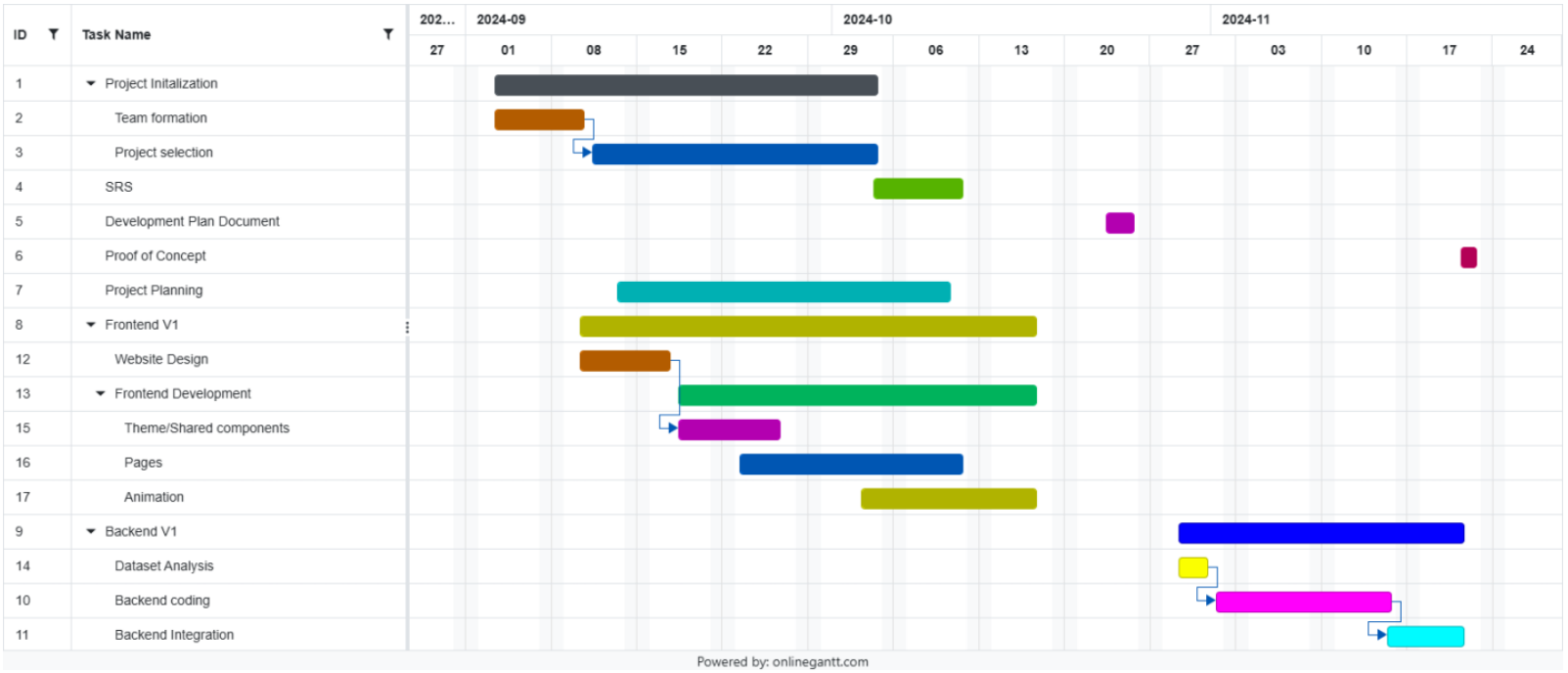


Figure 1: September-December

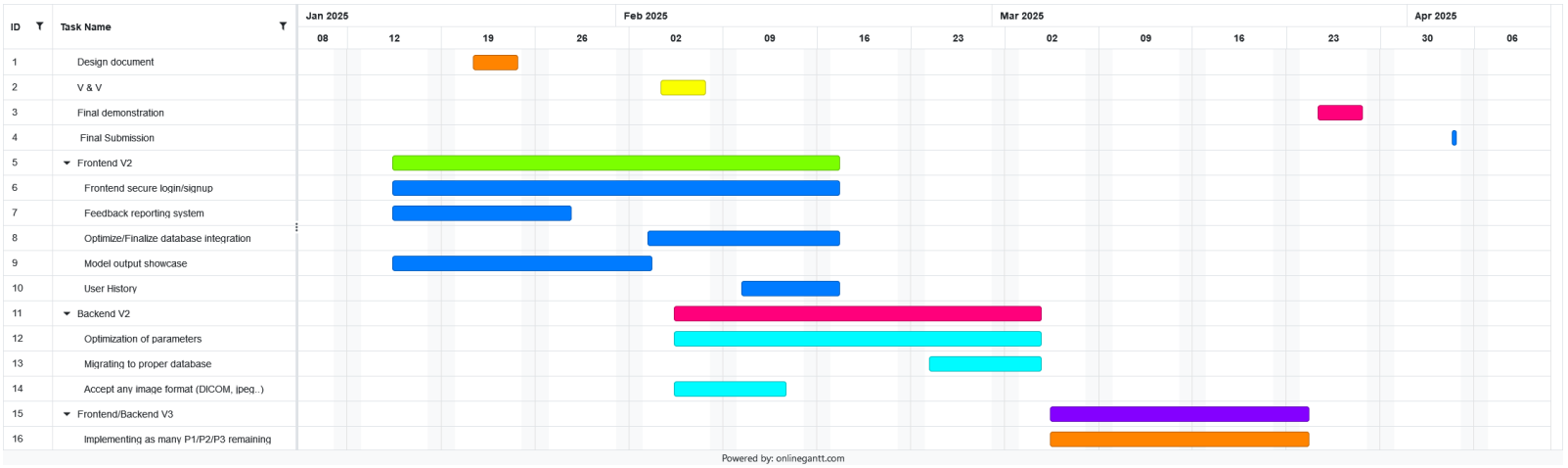


Figure 2: January-April