Development Plan
Software Engineering

Team 17, Team RAdiAIdance
Allison Cook
Ibrahim Issa
Mohaansh Pranjal
Nathaniel Hu
Tushar Aggarwal

Table 1: Revision History

Date	**Developer(s)**	**Change**
09/23/2023	Nathaniel Hu	Updated project header information, began initial drafting of various sections and subsections
09/25/2023	Tushar Aggarwal	Updated project header information, added to/updated Team Meeting Plan, Team Member Roles, Workflow Plan, Technology and Coding Standard sections
09/25/2023	Tushar Aggarwal	Final review of entire document, corrected spelling errors, formatting
09/25/2023	Mohaansh Pranjal	Modified/Added to Team Communication Plan and Proof of Concept Demonstration Plan subsections
09/25/2023	Allison Cook	Second review of entire document
09/25/2023	Nathaniel Hu	Transferred Development Plan document text from collaborative document into this LaTeX document; Did some finishing touches
...	...	...

In this development plan document, the team meeting plan will be outlined in detail. The team communication plan will also be discussed, considering the methods of communication, and defining expected communication etiquette (e.g., responding within 24 hours). The team member roles will also be defined here with specific responsibilities outlined for each role. The workflow plan will

describe how git will be used to manage project workflows, and how issues will be managed.

The proof-of-concept demonstration plan will also be outlined to discuss the main risk(s) for the project's success and the plan to demonstrate that we will be able to overcome these risk(s). The technology that is planned to be used for this project will also be described in more detail. The coding standard(s) for this project will also be included in detail. The project schedule will outline the expected project milestone completion dates and include details describing what is expected to be completed by when.

1 Team Meeting Plan

The team's meeting plan is outlined in detail in each of the relevant subsections below.

1.1 Meeting Schedule

The team's meeting schedule is outlined in detail below:

- Monday at 2:30 pm – 4:20 pm (weekly, during tutorial time slot)
  - Work on project deliverables, and discuss concerns.
  - Meet in person at a prearranged meeting place (TBD), or online.
- Tuesday, Wednesday at 12:30 pm – 1:20 pm (weekly, during lecture time slots)
  - Attend lectures, and get instruction/feedback from professors and TAs on project work.
  - Meet in person at ITB AB102 (during live lectures), or in person elsewhere or online.
- Friday at 12:30 pm – 1:20 pm (weekly, during lecture time slot)
  - Attend lectures, and get instruction/feedback from professors, and TAs on project work.
  - Meet in person at ITB 137 (during live lectures), or in person elsewhere or online.
- Friday at 2:30 pm – 3:30 pm (weekly, outside of scheduled classes)
  - Meet in person with supervisor, Dr. Mehdi Moradi, to discuss project progress, ask clarifying questions, seek feedback on existing work completed and plan next steps.

1.2 Meeting Rules

The team's meeting rules are outlined in detail below:

- The above meeting plan is subject to change based on the availability of all team members.
- Unless otherwise notified ahead of time or in case of emergency, all team members are expected to attend all planned meetings in full.
- Before each meeting, an agenda will be created, to which team members may add discussion topics to cover during the meeting.
- During each meeting, the meeting chair will:
  - Manage the meetings to ensure that all discussions stay on topic.
  - Refer to the agenda to guide the meeting and may include or exclude discussion topics at their discretion.
  - Be responsible for creating, updating, and managing the GitHub issues created to document the details of each meeting.

1.3 Meeting Roles

The following meeting roles have been defined by the team with the associated meeting role responsibilities, as outlined in detail below:

- **Chair**: This individual will change every meeting.
  - They ensure that the meeting stays on track and that all agenda items are addressed.
  - They are also responsible for ensuring that the meeting progresses at a decent pace and starts and finishes on time.
- **Scribe**: This individual will change every meeting.
  - This role is to record all decisions made by the team, any problems that may occur in the meeting, and track the outcome of the meeting.
  - These notes are to be posted to the meeting issue that is created for all team members to see.
- **Scrum Master**: Tushar Aggarwal
  - While this role will be elaborated upon in the "Team Member Roles" section, the scrum master is responsible for updating the GitHub per (applicable) new action items as well as any new information about current GitHub issues.
  - They will also be responsible for ensuring that all action items are assigned to a member and have a due date.

2 Team Communication Plan

For the team communication plan, the following rules and guidelines outline the general expectations of this plan for all team members:

- The main channels of communication will be MS Teams, with other communication channels to potentially be added later (e.g., Discord, etc.)
  - Meetings should be scheduled ahead of time on MS Teams to ensure they appear on all members' MS Teams calendars.
  - Syncs are scheduled as recurring meetings on MS Teams, but can be cancelled or rescheduled as needed.
  - Meetings should only be used for longer updates and discussions and should not be used for quick updates or questions.
  - The team will use the MS Teams chat for quicker updates and minor roadblocks or questions.
  - All team members are expected to respond to all messages (or at least indicate they have read all messages, ideally through message reactions) within 12 hours.

  For further details on meetings, please see the "Team Meeting Plan" section.

- Communication with any supervisors, stakeholders or instructors will be carried out by email and all team members should be added to the CC list.
  - Tushar will oversee communication with the profs/supervisor.
  - In addition, we will be communicating with Lyuyang Wang (Capstone TA) via MS Teams for assistance, and during his office hours whenever necessary.

- GitHub Issues will be the centralized for note-taking, brainstorming, and formalizing of project deliverables.
  - No other word processors or project management tools will be used.
  - Communication regarding project issues is encouraged to be done using GitHub issues so that comments are directly attached to the code they concern.

- If any team members are unable to attend any scheduled meetings, they must give notice at least 1 hour before the scheduled meeting start time, except for in emergencies.

The above rules are general guidelines subject to change and will be enforced at team discretion.

Name	Email Address
Allison Cook	cooka27@mcmaster.ca
Ibrahim Issa	issai2@mcmaster.ca
Mohaansh Pranjal	pranjalm@mcmaster.ca
Nathaniel Hu	hun4@mcmaster.ca
Tushar Aggarwal	aggarwt@mcmaster.ca

2.1 Team Member Contact Information

The following table contains the names and email addresses of all team members:

3 Team Member Roles

The roles and responsibilities of each team member are outlined in detail as shown below:

Role Descriptions

- **Developer**: All
  The developer role is the default role for all team members. While members may be experts in a particular domain, all members are expected to be doing development work throughout the project.
- **GitHub Administrator**: Tushar
  The lead QA for Git will be responsible for enforcing the Git-related sections of the workflow plan. They will be held accountable for ensuring that the code base is as organized as possible and that team members adhere to the workflow plan.
- **Project Manager**: Tushar
  The project manager will be responsible for all communication between the team and all project stakeholders. The project manager sets and enforces the schedule for deadlines and due dates (with the caveat that members must reach a consensus on the schedule).
- **Scrum Master**: Ibrahim, Tushar
  The scrum master will be responsible for updating the GitHub issues per (applicable) new action items and as well as any new information about the current GitHub issues. They will ensure that there is linkage between GitHub issues and GitHub commits.
- **Scribe**: Any, By Assignment
  For more information on the scribe role, please see the "Team Meeting Plan" section.

Responsibilities

- Allison Cook – Backend, Frontend
- Ibrahim Issa – Computer Vision/AI, Backend, Frontend
- Mohaansh Pranjal – Backend, Frontend
- Nathaniel Hu – Backend, Frontend
- Tushar Aggarwal – AI, Backend, Frontend

It should be noted that the roles and responsibilities specified above are subject to change at the discretion of the team.

4 Workflow Plan

The outline and workflow of the team's workflow plan are detailed in the subsections below.

4.1 Outline

The detailed outline of our workflow plan is as shown below:

- GitHub Issues will be used to manage and delegate tasks, track features and bugs, document meetings, research, and other miscellaneous project items.
- All issues must be categorized by one or more labels and assigned to one or more team members (aside from meeting issues). Issue templates will be created and used on an as-needed basis.
- A unique branch will be created for all features, bugs, testing, and environment setup issues.
- A pull request is subject to review by another team member and will be required before merging an unprotected branch to the main branch.
- A branch must also pass all automated test cases before merging. Code changes should be made in frequent, atomic commits while referencing the commit number for traceability.
- All deliverable documents may be directly edited on the main branch without the need for a pull request.

4.2 Workflow

For each sprint, tickets will be visible on an issue-tracking (Kanban) board with four (4) columns, in one of the following four (4) states:

- **To Do**: Tickets that have not yet been started in any capacity but are assigned to a sprint.

- **In Progress**: Tickets that the assignee/assigned developer has begun working on.
- **In Verification**: Tickets that the assignee/assigned developer has completed and is ready to be merged into the master branch.
- **Done**: Tickets that have been merged into the master branch.

The following process describes the lifecycle of an issue on GitHub:

1. Create a new issue detailing the new feature(s) to be developed (if one does not already exist). Be sure to include:
   (a) A general description of the feature(s) and any changes that need to be made.
   (b) A rationale for this change (as is necessary)
   (c) Any dependencies (as applicable)
   (d) Acceptance criteria for verifying the feature(s) have been implemented according to expectations.
2. Pull any new changes from the development branch.
3. Create a new branch to develop the new feature(s).
4. When developing the new feature(s), be sure to:
   (a) Write new unit tests/perform unit testing on the newly created modules and functions.
   (b) Push any changes to the created branch.
   (c) Have the changes reviewed and approved by 1-2 other team members (ideally the relevant project area expert/main driver).
   (d) Merge the changes into the development branch after the approval(s) have been made.
   (e) Periodically perform regression testing on the changes merged into the development branch. Create issues for all bugs found in a similar process to that outlined in step 1.
   (f) Repeat steps 2 – 8 for all bugs discovered during regression testing from step 9.
   (g) Repeat regression testing to ensure all discovered bugs from step 9 have been resolved, and that no new bugs have been introduced.
   (h) Changes merged into the development branch will be merged into the main branch periodically, ideally after the regression testing has been completed and all discovered bugs have been addressed and retested.

The general steps outlined above are subject to change and adaptation at the team's discretion.

5 Proof of Concept Demonstration Plan

The following points are a general outline of the team's plan for the proof-of-concept demonstration:

1. Identify the major risks that may prevent the successful completion of this project's proposed solution.
2. Implement a proof of concept of the major risks to include in the proof-of-concept demonstration.

For the proof-of-concept demonstration, the following risk(s) will be demonstrated to prove that they can be overcome to ensure the project's successful completion:

1. Prove that the AI/computer vision component needed to analyze the chest X-ray images and perform diagnoses is feasible. This would be shown in the demo by implementing a simple machine learning model that is able to identify certain diseases in a chest X-ray image.
2. Prove that the mobile/web application (with server backend) for this solution is feasible. The main risk here would be ensuring secure access of data.
3. *Extra*: Prove that the natural language component needed to produce a free-form or structured radiology report is feasible.

6 Technology

The following technologies will be used by the team during the software development process to implement the proposed solution:

- **Frontend Languages**: JavaScript, CSS
- **Backend Languages**: Python
- **Frontend Libraries**: React, Tailwind CSS
- **Backend Libraries**: PyTorch, OpenCV, pandas
- **Unit Testing**: Jest, Pytest
- **Code Coverage**: Istanbul, Coverage.py
- **Linters**: Prettier, ESLint, Flake8, Black
- **Continuous Integration and Branch Protection**: GitHub Actions
- **Other Software Tools**: git, Overleaf, VSCode

7 Coding Standard

The following coding standard(s) will be adhered to by the team during the software development process:

Branch Naming Convention
Ensure that all your branches are named according to the following convention:

- `feature/<feature-name>` for new features
- `bugfix/<bug-name>` for bug fixes
- `improvement/<improvement-name>` for improvements
- `chore/<chore-name>` for chores

Ensure that the name of your branch is descriptive of the feature/bug/improvement/chore you are working on and is written in `kebab-case`.
Examples:

- `feature/add-login-page`
- `bugfix/fix-login-page`
- `improvement/update-login-page`
- `chore/update-readme`

Commit Message Convention
This project uses Conventional Commits for commit messages. Ensure that all your commit messages are written according to the following convention:
`<type>: <description>`

Where `type` is one of the following:

- `feat` for new features
- `fix` for bug fixes

Examples:

- `feat: add confirm password field`
- `fix: submit button not working`

Coding/Programming Convention
Coding standards will be language specific, with all Python code being upheld to the PEP 8 style guide and all JavaScript code being upheld to Airbnb's style guide.

8 Project Scheduling

The following schedule provides a general idea of the project timeline that the team will follow for the various deliverables:

- **Problem Statement and Goals** – Due: September 25th, 2023
- **Development Plan** – Due: September 25th, 2023
- **Requirements Document, Revision 0** – Due: October 4th, 2023
  - Start By: September 27th, 2023
  - Complete By: October 2nd – 3rd, 2023
- **Hazard Analysis, Revision 0** – Due: October 20th, 2023
  - Start By: October 6th, 2023
  - Complete By: October 17th – 18th, 2023
- **V&V Plan, Revision 0** – Due: November 3rd, 2023
  - Start By: October 22nd, 2023
  - Complete By: October 31st – November 1st, 2023
- **Proof of Concept Demonstration** – Due: November 13th – 24th, 2023
  - Start By: September 27th, 2023
  - Complete By: November 10th – 11th, 2023
- **Design Document, Revision 0** – Due: January 17th, 2024
  - Start By: September 27th, 2023
  - Complete By: January 14th, 2024
- **Revision 0 Demonstration** – Due: February 5th – 16th, 2024
  - Start By: September 27th, 2023
  - Complete By: February 1st, 2024
- **V&V Report, Revision 0** – Due: March 6th, 2024
  - Start By: November 13th, 2023
  - Complete By: March 2nd, 2024
- **Final Demonstration, Revision 1** – Due: March 18th – 29th, 2024
  - Start By: February 18th, 2024
  - Complete By: March 14th, 2024
- **EXPO Demonstration** – Due: April TBD

  - Start By: February 18th, 2023
  - Complete By: March 31st, 2023

- **Final Documentation, Revision 1** – Due: April 4th, 2023
  - Start By: February 3rd, 2023
  - Complete By: March 31st, 2023

This schedule is intended to ensure that all milestones are met and that all project deliverables are completed ahead of schedule. This is to allow for extra time for the team members to review, edit and revise these documents with the supervisor and TA before they are signed off on and submitted.