Development Plan
LiDart

Team 10
Jonathan Casella
Kareem Elmokattaf
Michaela Schnull
Neeraj Ahluwalia

Table 1: Revision History

Date	Developer(s)	Change
26/Sep/2022	Michaela Schnull	Initial Release
14/Nov/2022	Michaela Schnull	Updated team communication plan and team member roles and responsibilities Revised the scope of the POC Plan

Table 2: Acronyms

Acronym	Description
API	**A**pplication **P**rogramming **I**nterface
CAD	**C**omputer **A**ided **D**esign
CI	**C**ontinuous **I**ntegration
LiDAR	**Li**ght **D**etection **A**nd **R**anging
POC	**P**roof **o**f **C**oncept
PR	**P**ull **R**equest
UI	**U**ser **I**nterface
UX	**U**ser **E**xperience

1 Introduction

3D scanning is a versatile technology that is used across many industries, but its uses are often limited by high cost and complexity. LiDart aims to build a low-cost, simple to use 3D scanning robot. A software suite will process data obtained from the robot and provide a user interface. LiDart's end product will be a wheel-based mobile robot with all required sensors on-board that can be connected to over WiFi.

2 Team Meeting Plan

Weekly meetings will take place in H.G. Thode Library of Science and Engineering. The frequency of meetings is subject to change depending the needs of the project. Meetings will be used to review the project schedule, discuss design decisions, identify and create GitHub issues, and assign actions to be taken. This includes creating plans for upcoming deliverables, reviewing/creating issues tracked on GitHub, and holding code walk-throughs/design reviews.

3 Team Communication Plan

The team will use instant messaging for items that require and urgent response. Communication will also occur through GitHub using issues. Users will be tagged in issues that require their attention. All team members members may schedule meetings to address specific issues.

4 Team Member Roles and Responsibilities

The following roles and responsibilities have been assigned to team members. Team members may take on additional responsibilities, for example by reviewing other team members' work.

4.1 Jonathan Casella

- Development and implementation of computer vision algorithms
- Development and implementation of localization algorithms
- Creation of a user application that displays scanning results

4.2 Kareem Elmokattaf

- Development of the controls software for the robot
- Development and implementation of localization algorithms
- Interfacing of hardware and software systems
- UI/UX design of a user application that displays scanning results

4.3 Michaela Schnull

- Electrical design of the robot, including the creation of electrical schematics to document electrical design
- Interfacing of hardware and software systems
- Project management activities, including maintenance of the project board on GitHub, budgeting, scheduling, and acting as the team liaison

4.4 Neeraj Ahluwalia

- Mechanical design of the robot, including the creation of CAD models
- Interfacing of electrical components in the mechanical design
- Marketing activities, including logo design and video presentations

5 Workflow Plan

5.1 GitHub Development Workflow

The workflow depicted in Figure 1 will be followed throughout the development process. This workflow supports CI and issue tracking trough GitHub. Commits should be frequent and have descriptive messages.

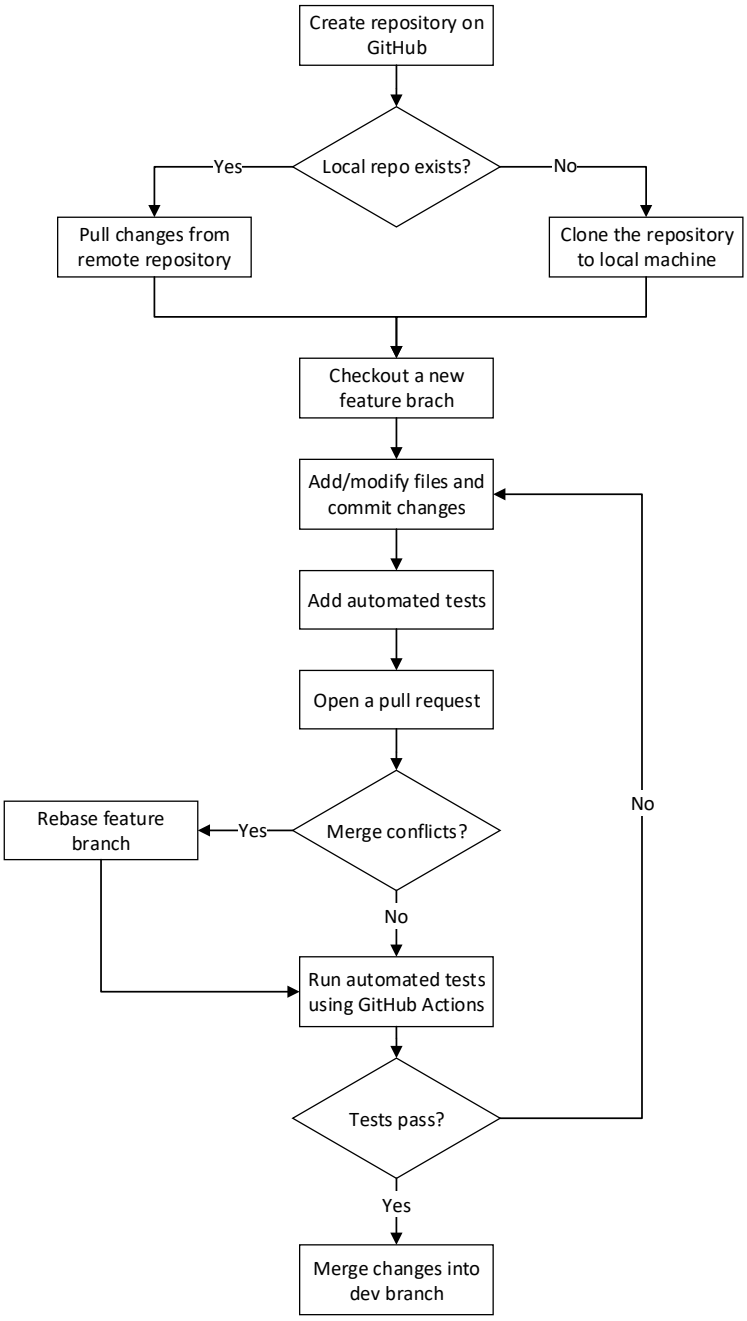

Figure 1: GitHub Development Workflow

5.2 Branches

Development will take place on a development (dev) branch. Feature branches will be used to fix bugs and add new features. They will be deleted after merging into the dev branch. Changes from the dev branch will be merged to the main branch after they have been reviewed and integration testing has been performed. Figure 2 depicts a sample branching structure.

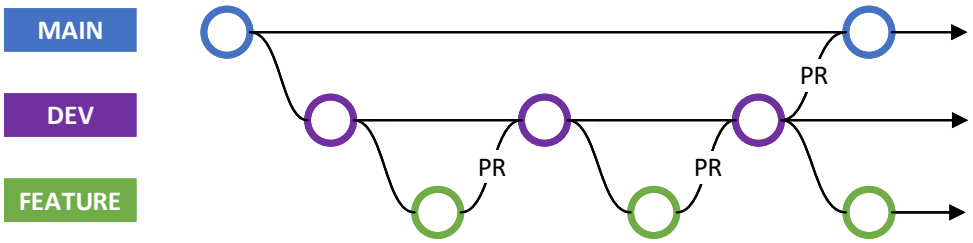

Figure 2: Sample Branching Structure

5.3 Issue Tracking

GitHub issues will be used to plan and track tasks. All team members can create new issues and assign them to other members. The following guidelines will be followed when working with issues:

- Issue templates will be used for bug reports and new feature requests
- Issues that require multiple steps should include task lists
- Default labels provided by GitHub will be used to classify the type of issue
- Issues should be linked to associated branches

A GitHub project board will be used to track and organize issues. Issues will be sorted into *Todo*, *In Progress*, and *Done* columns in a tabular view. Furthermore, issues will be assigned fields to categorize them based on priority and discipline. The priority field will classify issues as *High*, *Medium*, or *Low* priority. The discipline field will classify issues as *Mechanical*, *Electrical*, or *Software*.

6 Proof of Concept Demonstration Plan

6.1 Inaccurate Localization

Indoor robot localization is a complex and challenging problem. The robot must be able to determine its position using image-based camera localization. Challenges associated with this include probabilistic sensor data, sensor aliasing, noise in images, and unfavorable geometric characteristics of the surrounding environment. To mitigate this risk, a simulation of the robot and markers will be developed for the proof of concept demonstration. State estimation will be used to perform localization in the presence of simulated process and measurement noise. If adequate results are not obtained during POC testing, sensor fusion will be used in the final design to reduce uncertainty.

6.2 Robot Mechanical and Electrical Design

The robot must be able to move and position itself. A prototype of the robot will be created to test the electrical and mechanical design concepts. It should be able to execute simple movement commands during POC testing.

7 Technology

- Autodesk Inventor: CAD tool used to develop and model the mechanical design of the robot
- AutoCAD Electrical: CAD tool used to create electrical schematics
- Autodesk EAGLE: CAD tool used to design printed circuit boards
- Rust: High performance, low-level programming language ideal for embedded systems with built-in unit-testing
- Rustfmt: Lint tool designed for the Rust programming language
- OpenGL: API used to render scanning data
- OpenCV: Real-time computer vision library
- AprilTags: Visual marker system designed for use in robotics and camera calibration
- GitHub: Version control software with tools for CI and project management

8 Coding Standard

The *Rust Style Guide* will be used as a coding standard.

9 Project Scheduling

A Gantt chart will be used for scheduling. Additional tasks will be added as the project progresses.

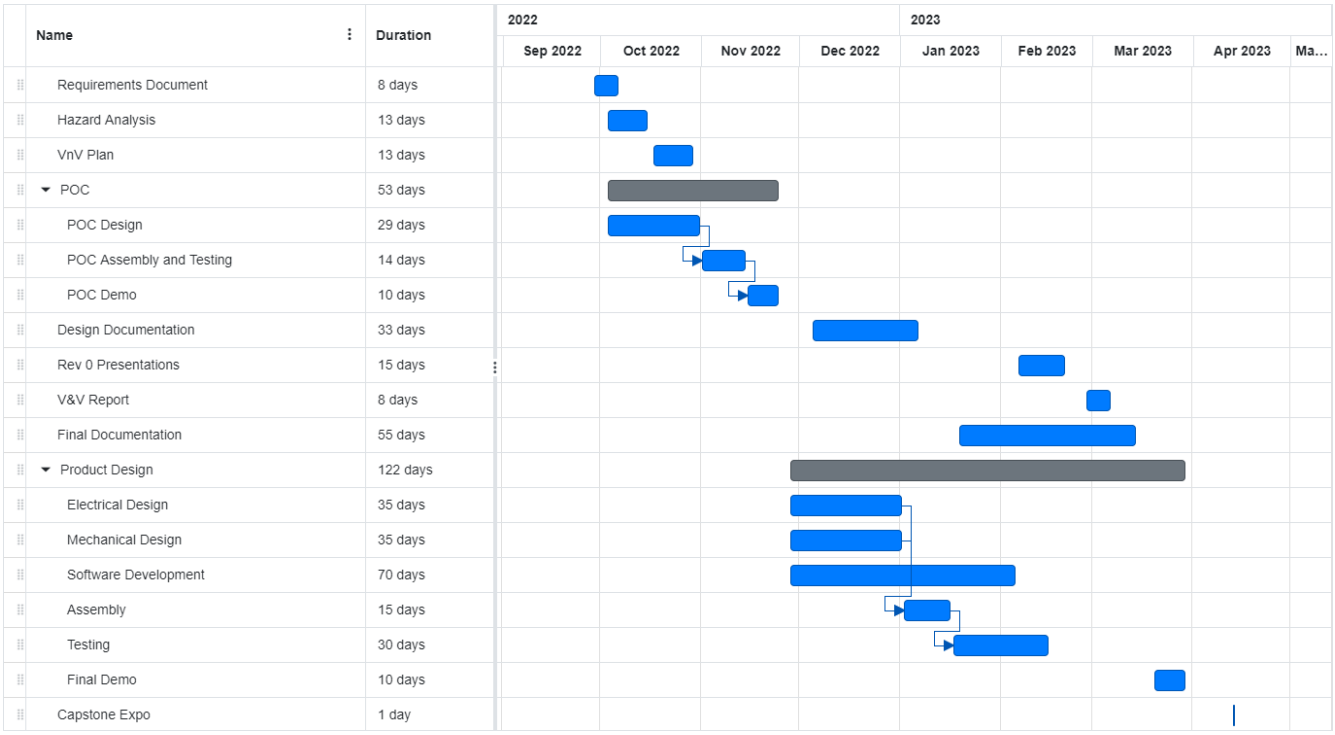

Figure 3: Project Schedule