ÇANKAYA UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

CENG 408

SOFTWARE DESIGN DOCUMENT

Members

Turgut Utku ALTINKAYA

Burak ATEŞ

Yunus Emre DİNÇEL

Bayram Alper KILIÇ

İlteriş SAMUR

Table of Contents

1. Introduction 4
   1.1. Purpose of this Document 4
   1.2. Scope of this Document 4
   1.3. Glossary 5
2. Overview of Document 5
3. System Design 6
   3.1. Architectural Design 6
   3.2. Data Flow Diagrams 7
      3.2.1. Context Diagram 7
      3.2.2. Level 1 DFDs 8
   3.3. Class Diagrams 11
      3.3.1 Class Diagram for Remote Computer 11
      3.3.2 Class Diagram for Vehicle (Raspberry Pi) 12
   3.4. Activity Diagrams 13
      3.4.1. Movement 13
      3.4.2 Data Stream 14
      3.4.3. Manual Mapping 15
      3.4.4. Autonomous Mapping 16
      3.4.5. Object Detection 17
   3.5. Sequence Diagrams 18
      3.5.1. Movement 18
      3.5.2. Data Stream 19
      3.5.3. Manual Mapping 19
      3.5.4. Autonomous Mapping 20
      3.5.5. Object Detection 21
4. User Interfaces 22
   4.1. Home Page 22
   4.2. Admin Dashboard with 2D Map 22
   4.3. Admin Dashboard with 3D Map 23

4.4. Interactieve UI Link 23

1. Introduction

1.1. Purpose of this Document

The purpose of this document is to provide a detailed presentation of the structure and architecture of the software included in Sentinel. The document aims to be reference material for developers, stakeholders, and team members by making the design decisions and implementation techniques in the operation of the system clearer. The document mainly includes sequence diagrams, activity diagrams, data flow diagrams (DFD), and user interface designs that help users and team members to get a better understanding of the system's functions, workflows, and interactions. This approach not only aims to facilitate effective project management through clear communication among team members, but also plays an important role in being a guide for future changes or iterations.

1.2. Scope of this Document

The Sentinel idea stemmed from the development of unmanned vehicle technologies in modern warfare. In today's world, many countries use technology in the defense industry for both self-defense and operational purposes. When these vehicles are not used, they cause many soldiers to lose their lives. For this reason, there is a great need for unmanned vehicles in war zones. For this reason, many countries are investing in these vehicles. This situation caught our interest and we decided to make an autonomous environment mapping vehicle. However, our vehicle is a proof of concept and isn't suitable for the war zone. The main purpose of Sentinel is to create 2D and 3D maps of its surroundings by moving autonomously. The vehicle can be sent to dangerous places to autonomously map the environment, thus preventing personnel loss. It can be used to create a map of the environment in many areas such as buildings that are not safe to enter, war zones and places contaminated with harmful gases. The importance of this is that it gives an idea about the place without human interaction and provides detailed 3D environment mapping that shows dangers in advance. It can also be used to detect life in rescue operations when a thermal camera is used. The vehicle is mainly equipped with Raspberry Pi5, which runs on Ubuntu, YDLidar X2, and Picamera 3. The data collected from the lidar sensor and camera are combined and used to generate the 3D map utilizing the rtabmap_ros, a package used to generate a 3D point of clouds of the environment and/or to create a 2D occupancy grid map for navigation. It will mainly use the Robot Operating System (ROS) to control both autonomous and manual movement, and environment mapping. The collected data is planned to be processed at a remote computer that is also equipped with ROS, and only the movement command from the remote computer will be returned to Sentinel. In this way, we also plan to provide a manual control mechanism which allows our users to interact with the vehicle any given time. Moreover, the generated map and the camera footage will be available to the user at the remote computer in real-time. After the map has been generated, we plan to assign jobs to the Sentinel, such as finding the red chair from the generated map, and it will find that red chair. To achieve this, we also plan to detect the objects using YOLO, a library for object detection, and then store these information using rosbag. In summary, users can control the movement of the vehicle, view images from the vehicle, and view 2D and 3D maps in real time. All of this

is done on a remote computer. On the other hand, Sentinel is responsible for sending sensor and camera data to the remote computer and obeying the command received from the remote computer. In the following sections of this SDD document, sequence diagrams, activity diagrams, and user interfaces of DFD (Data Flow Diagrams) will be shown to provide the development team with a better understanding and impression of the system functions and interfaces, and also to facilitate future improvements.

1.3. Glossary

Term	Description
Pi Camera Module 3	Compact camera from Raspberry Pi
Raspberry Pi 5	Small single-board computer
ROS Publisher/Subscriber	Publish/subscribe is a messaging pattern where publishers categorize messages into classes that are received by subscribers.
Rosbag	A set of tools for recording and playing back to ROS topics.
RTAB-Map	ROS framework for 3D mapping.
The Sentinel	The sentinel is a Discovery Vehicle. The name of our project, The Sentinel, is an autonomous vehicle with features such as 3D mapping and object detection.
Ubuntu	Linux based operating system
YDLidar X2	360-degree two-dimensional rangefinder sensor.
YOLO	You Look Only Once. Open source library that helps you object detection.

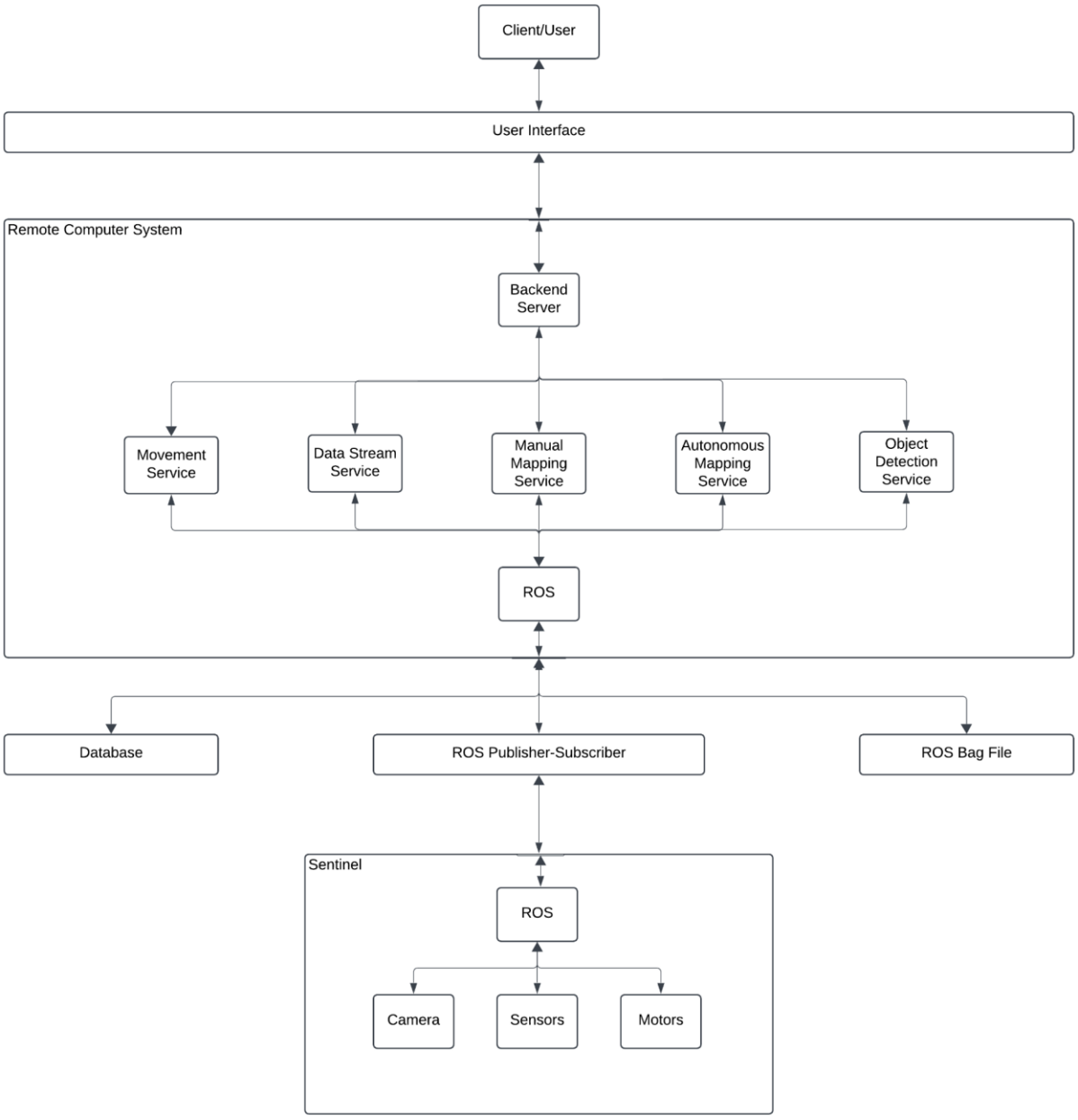
2. Overview of Document

This report provides an explanation of how the system's designed and functions, and user interfaces. At the beginning, the Architectural Design is discussed to illustrate how the different parts of the system work together. Then, The Data Flow Diagram section presents how information moves within the system starting from the Context Diagram and progressing to more detailed level 1 DFD illustrations, like information movement, data exchange procedures manual and autonomous mapping and object recognition. Class Diagram shows how the system will be implemented and which inheritions will be used. The Activity Diagrams illustrate workflows for tasks to be followed step by step. Sequence Diagram

representations give examples for the order of interactions between various system components; ultimately the document encompasses User Interfaces prototypes that help users to understand how layout works. Also, some interaction examples are given in this section.

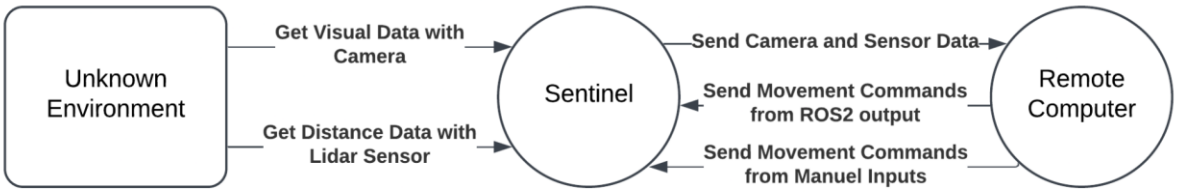
3. System Design

3.1. Architectural Design



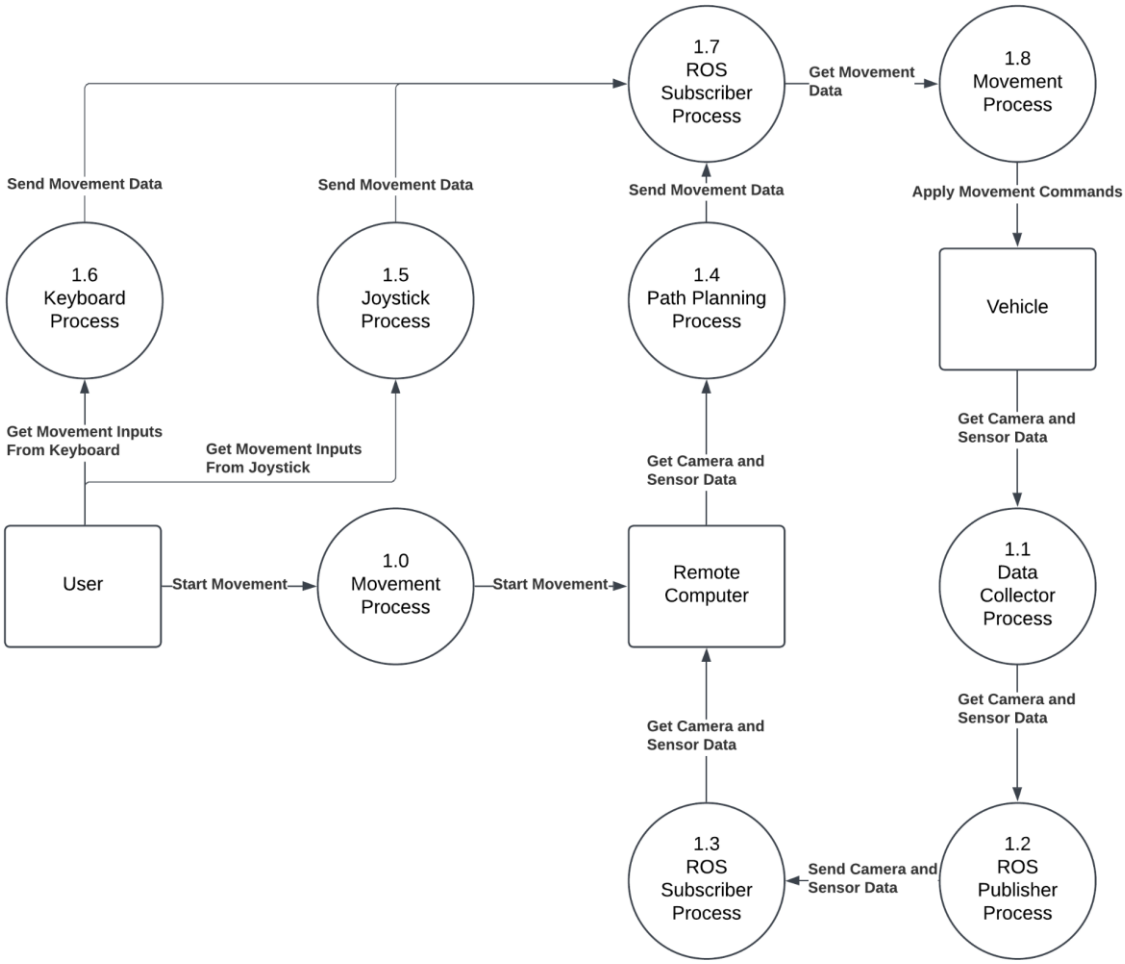
3.2. Data Flow Diagrams

3.2.1. Context Diagram



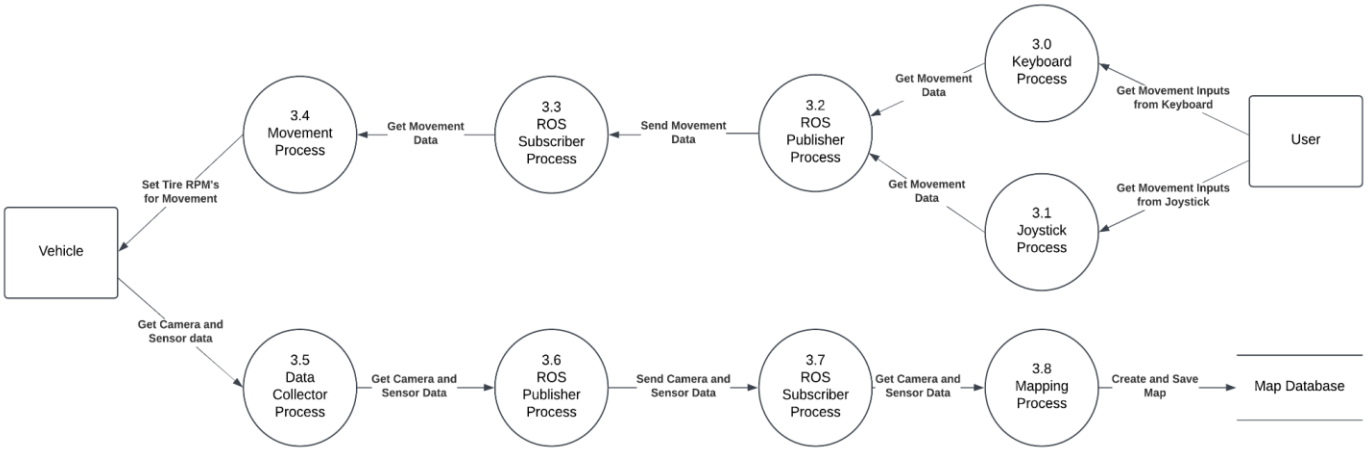
3.2.2. Level 1 DFDs

3.2.2.1. Movement

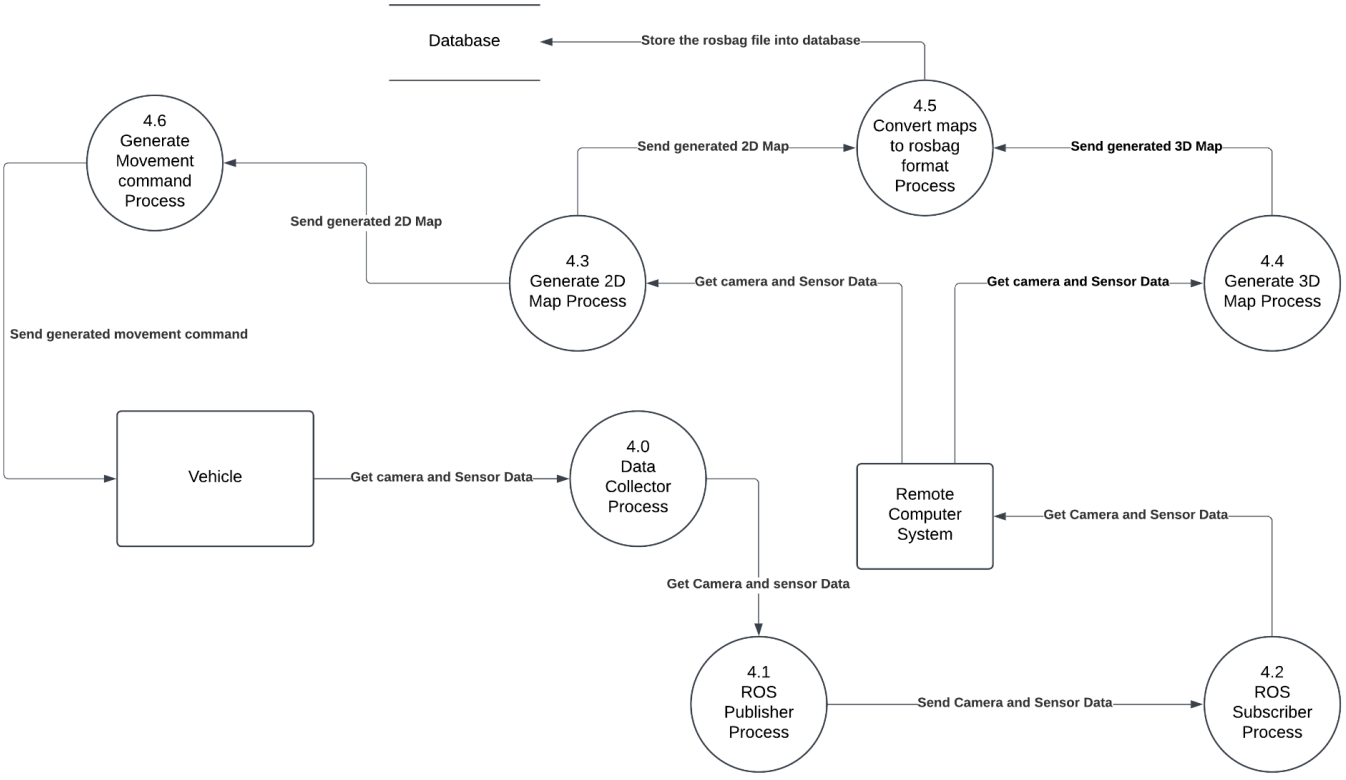


3.2.2.2. Data Stream

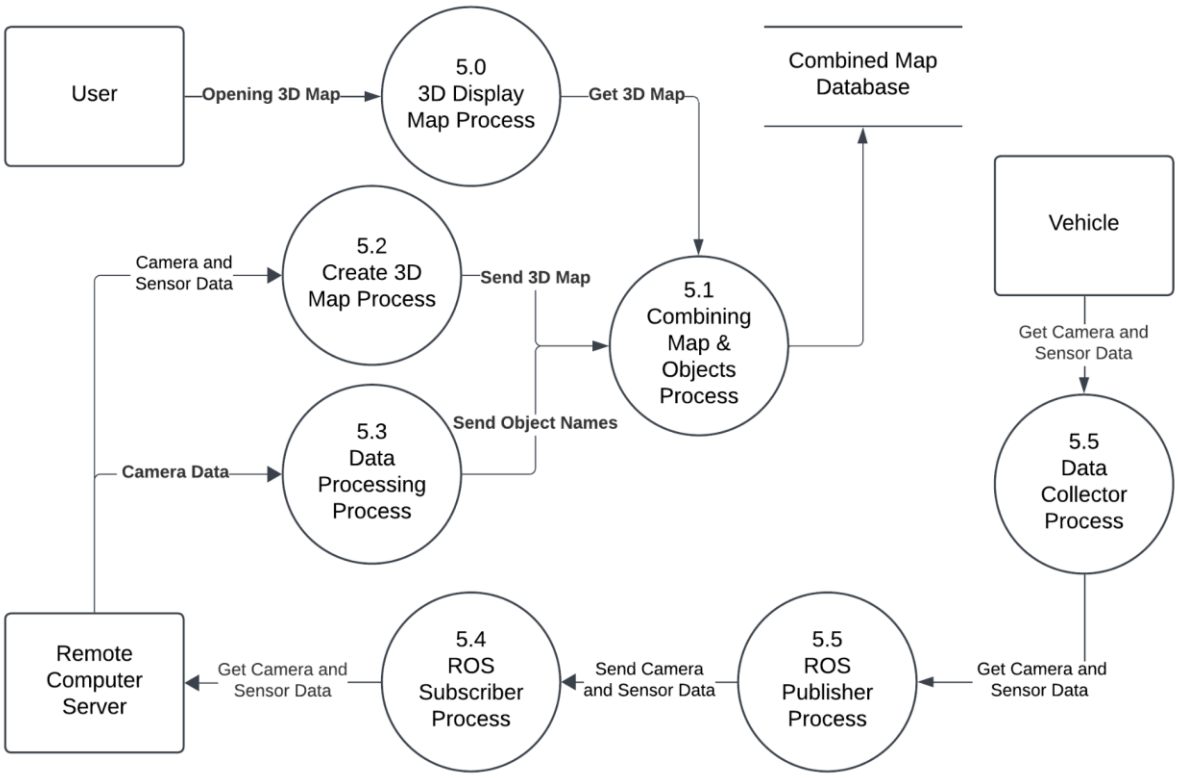
3.2.2.3. Manual Mapping



3.2.3.4. Autonomous Mapping

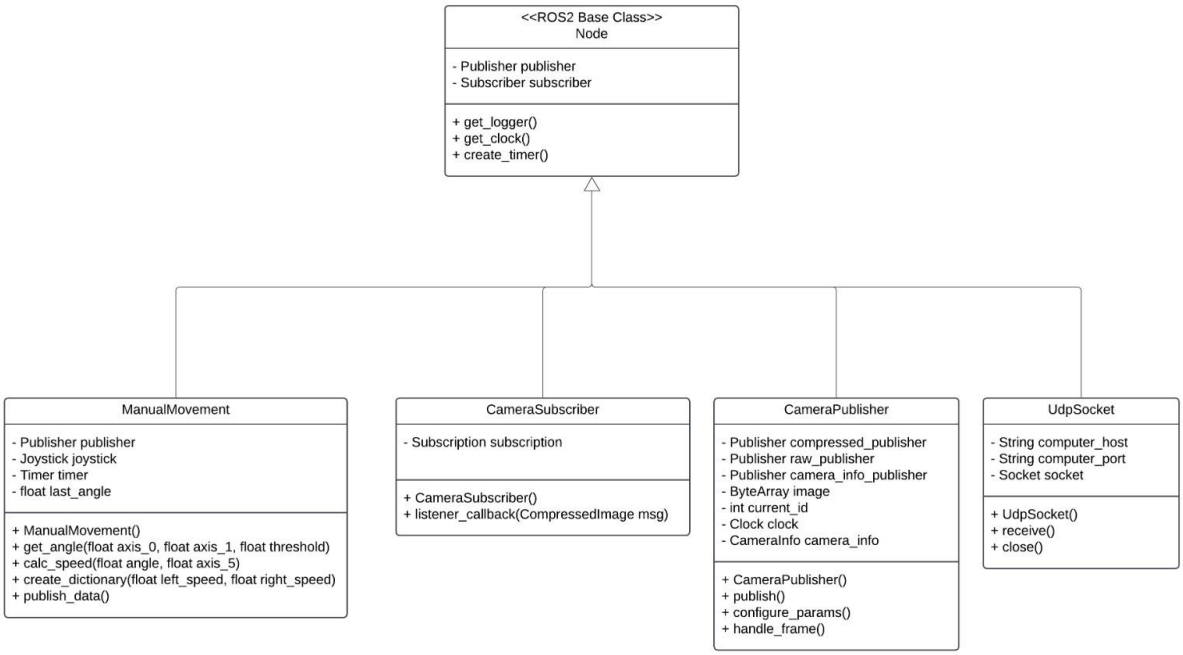


3.2.3.5. Object Detection

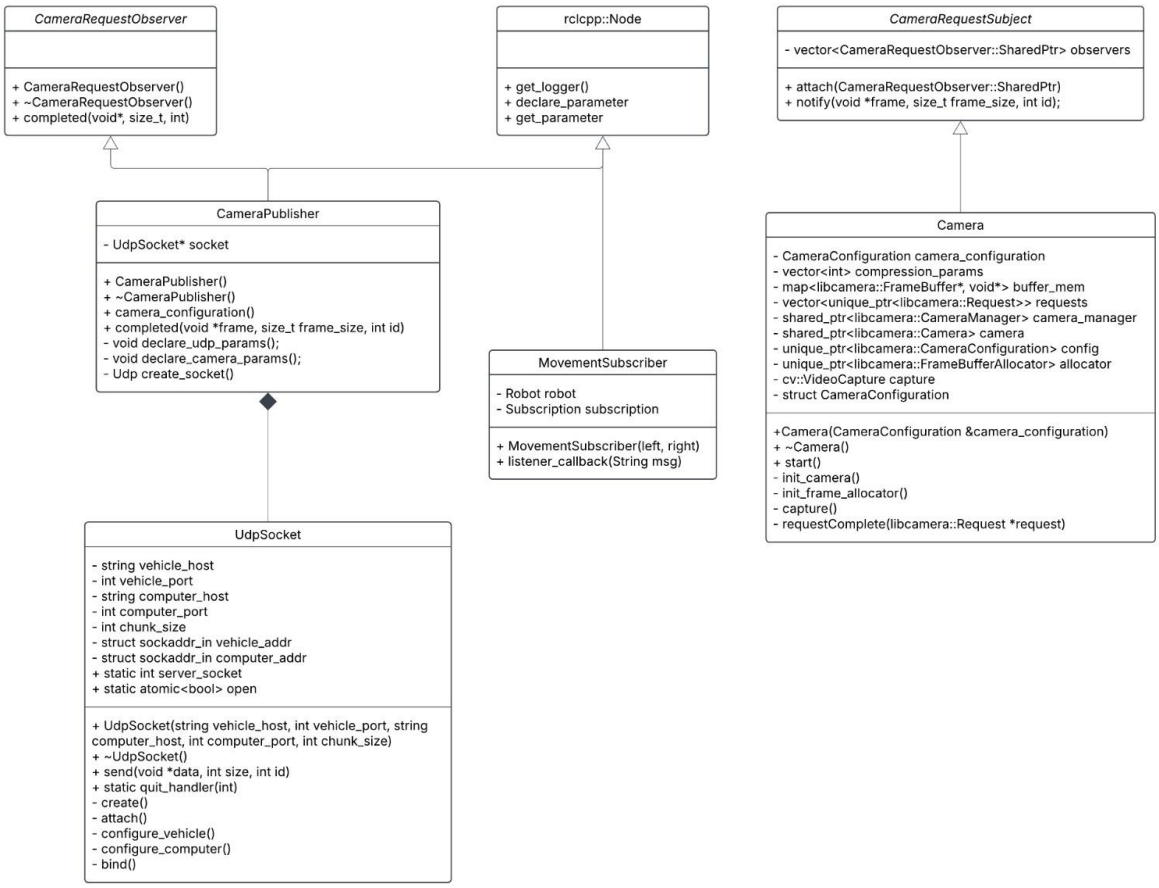


3.3. Class Diagrams

3.3.1 Class Diagram for Remote Computer

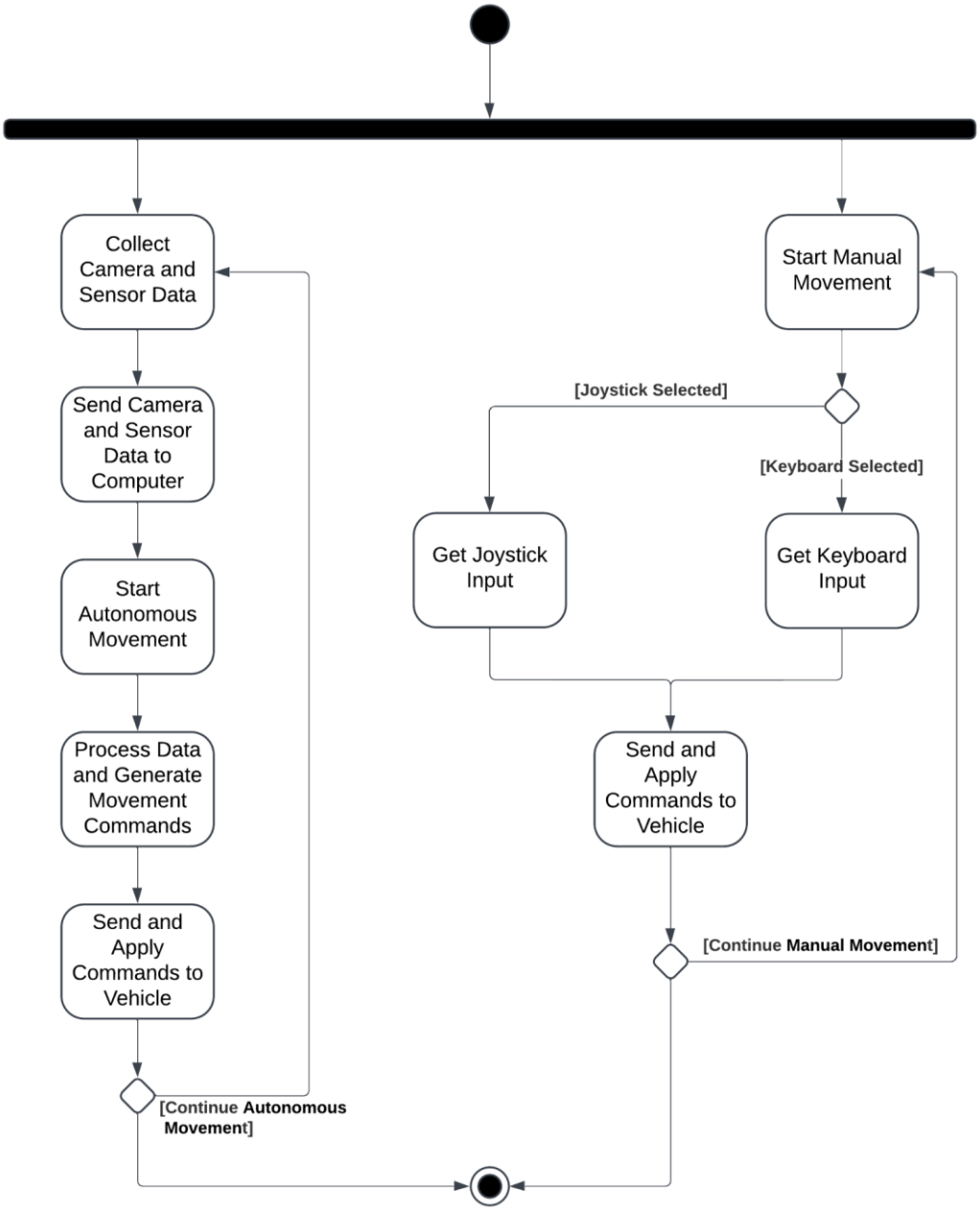


3.3.2 Class Diagram for Vehicle (Raspberry Pi)

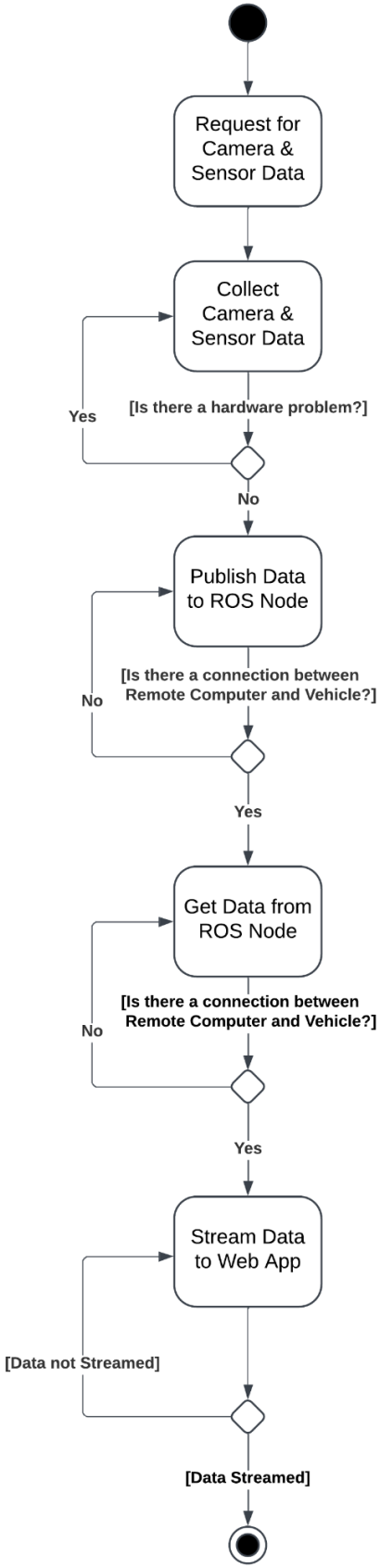


3.4. Activity Diagrams

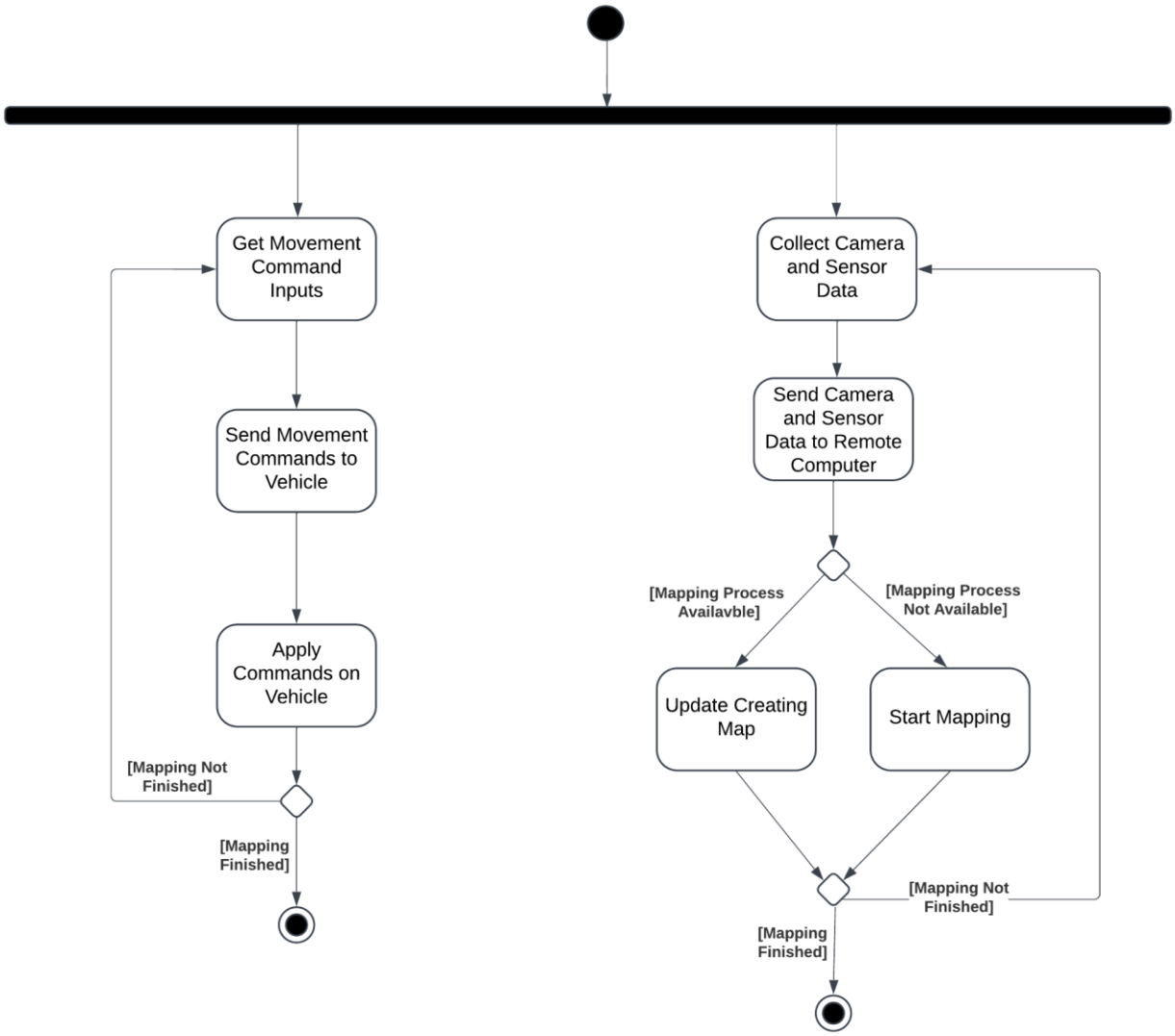
3.4.1. Movement



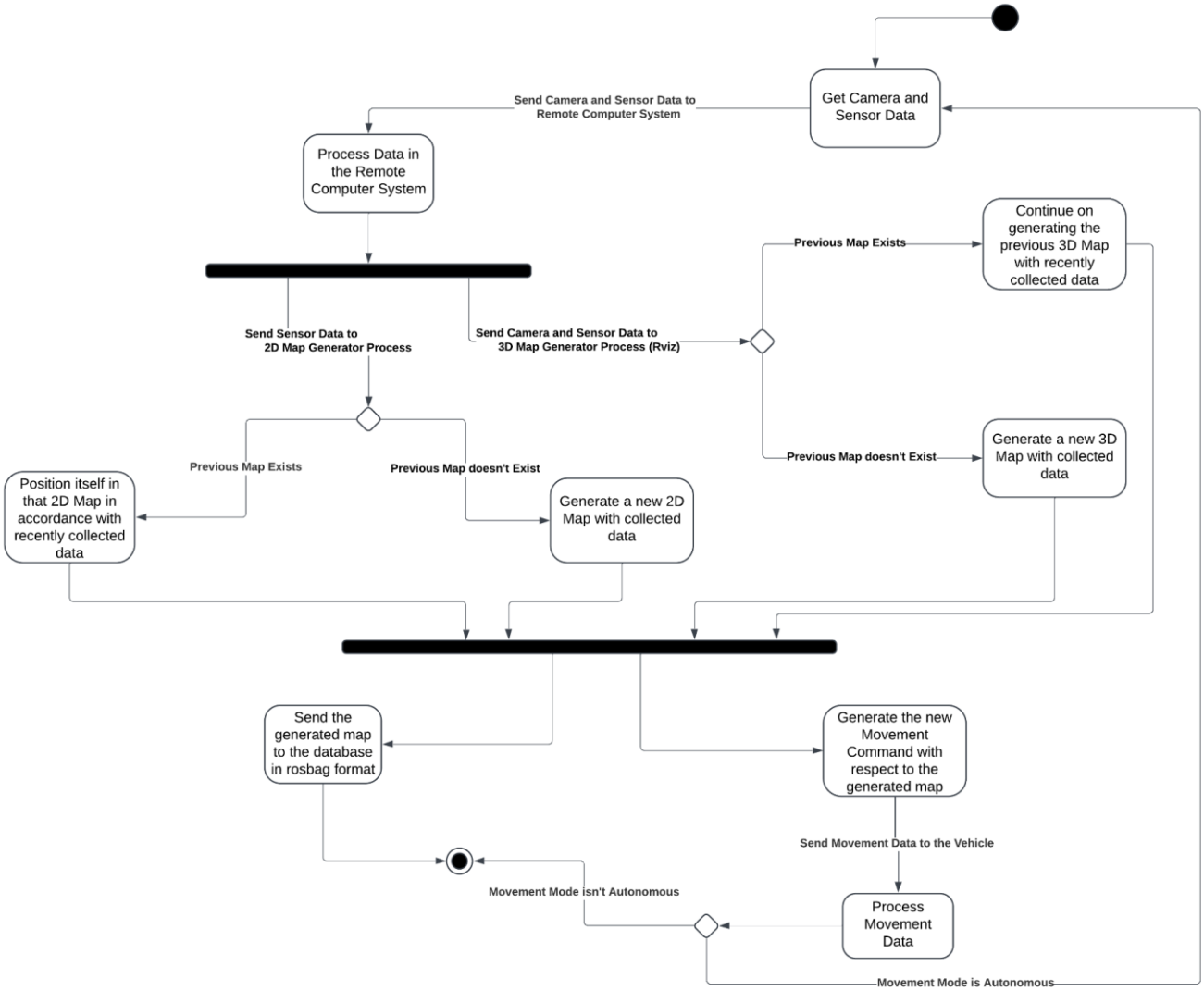
3.4.2 Data Stream



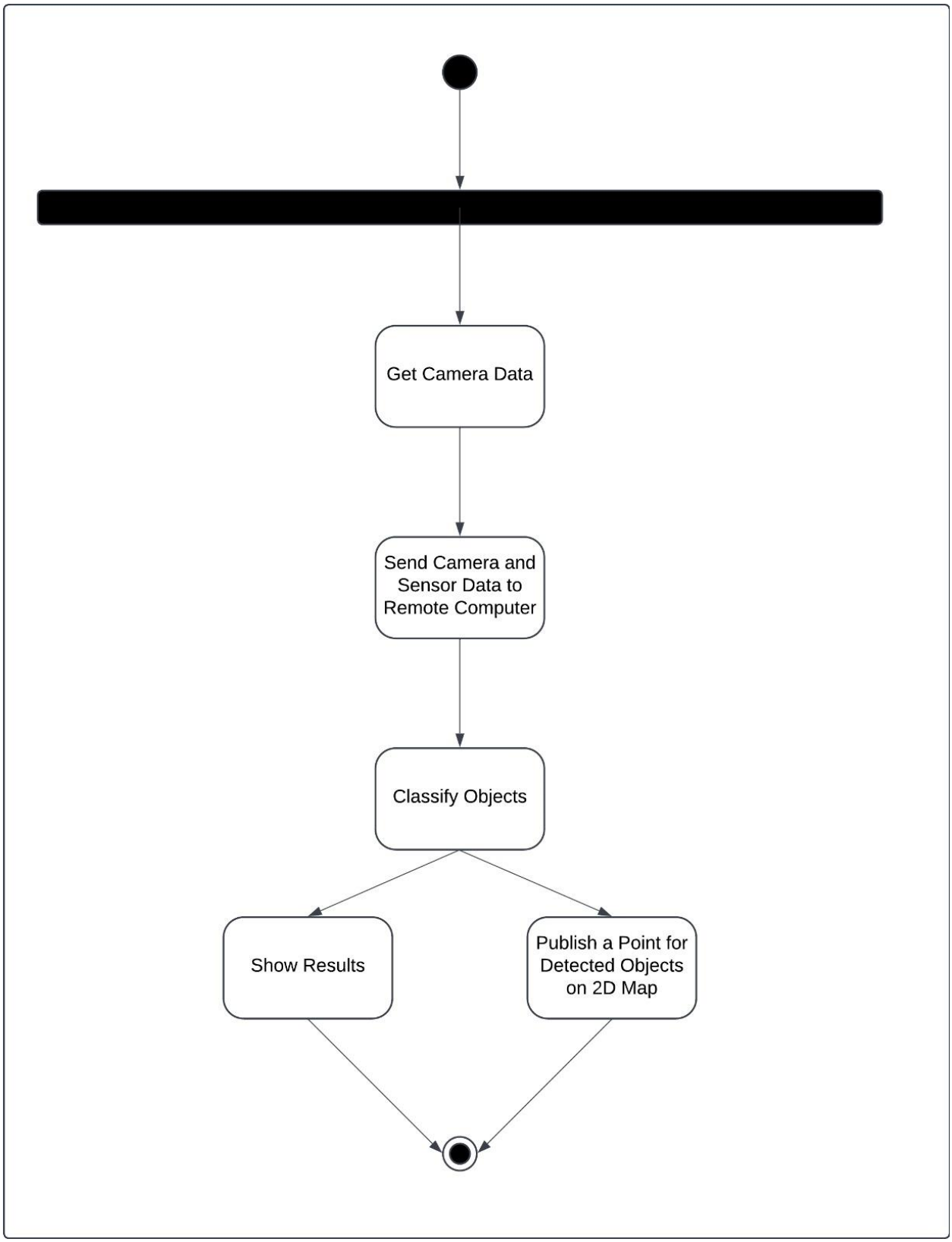
3.4.3. Manual Mapping



3.4.4. Autonomous Mapping

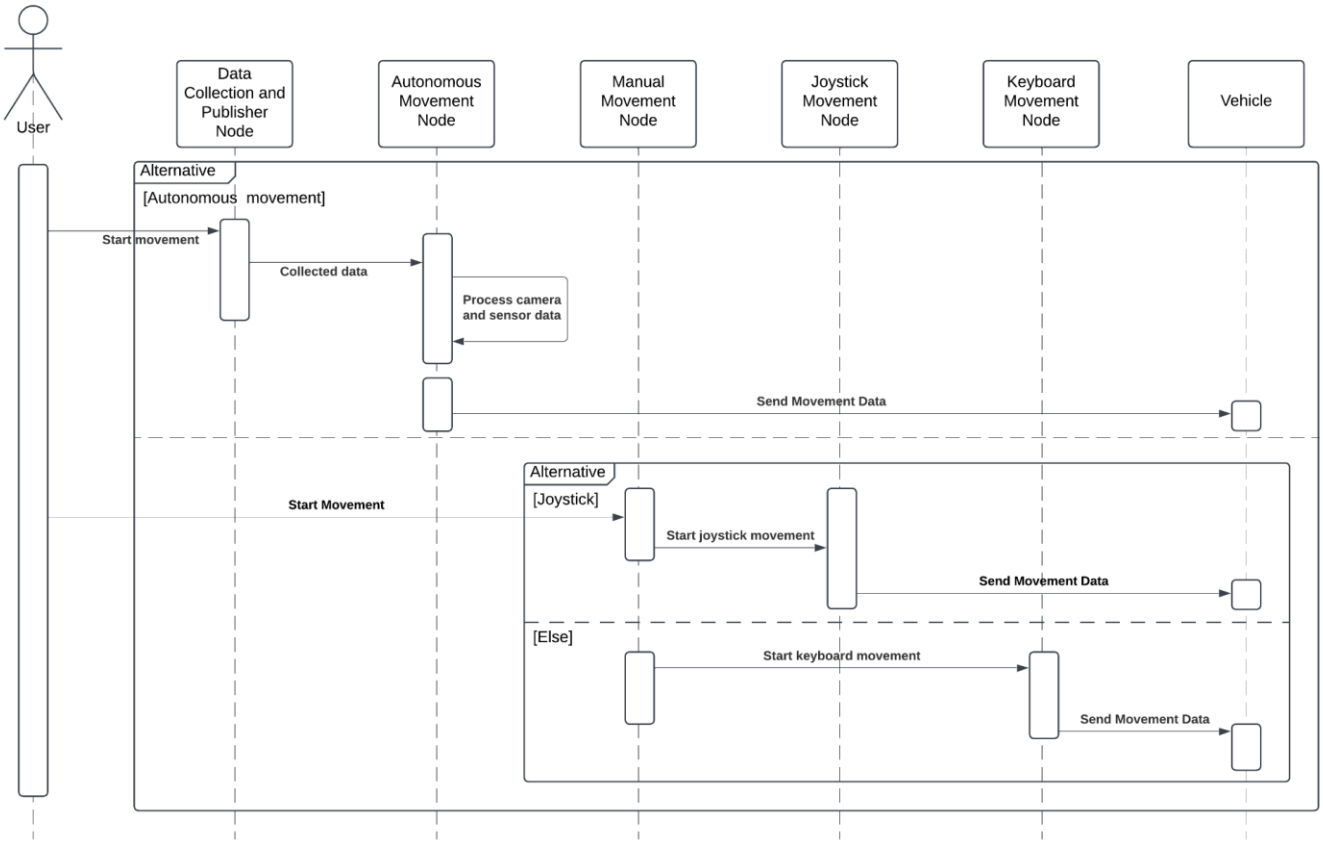


3.4.5. Object Detection

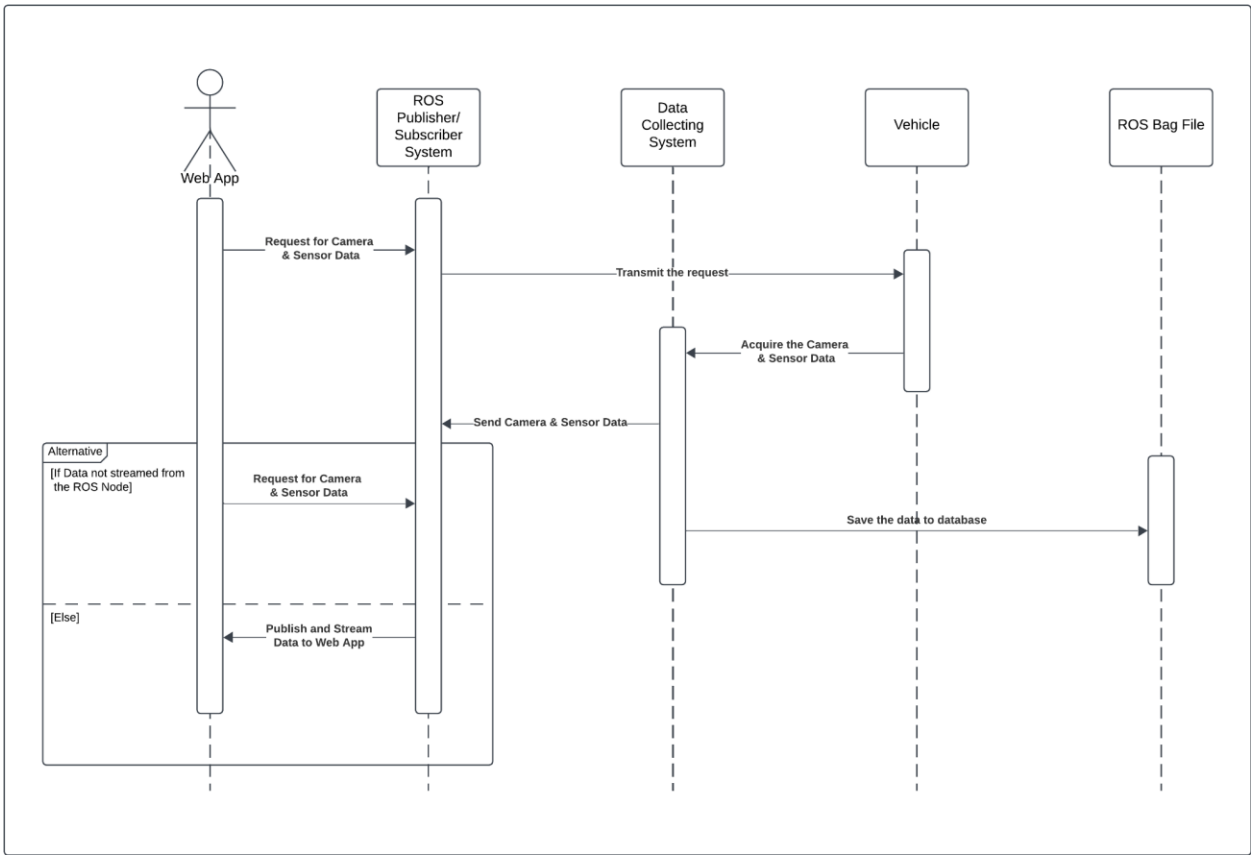


3.5. Sequence Diagrams

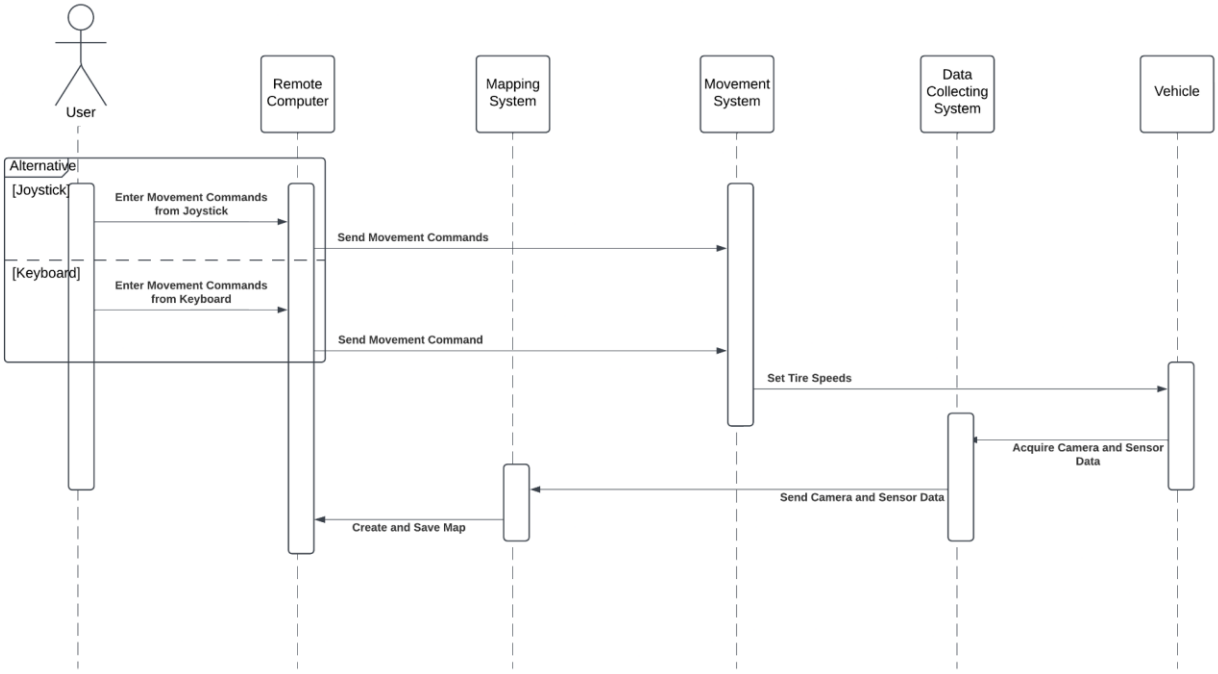
3.5.1. Movement



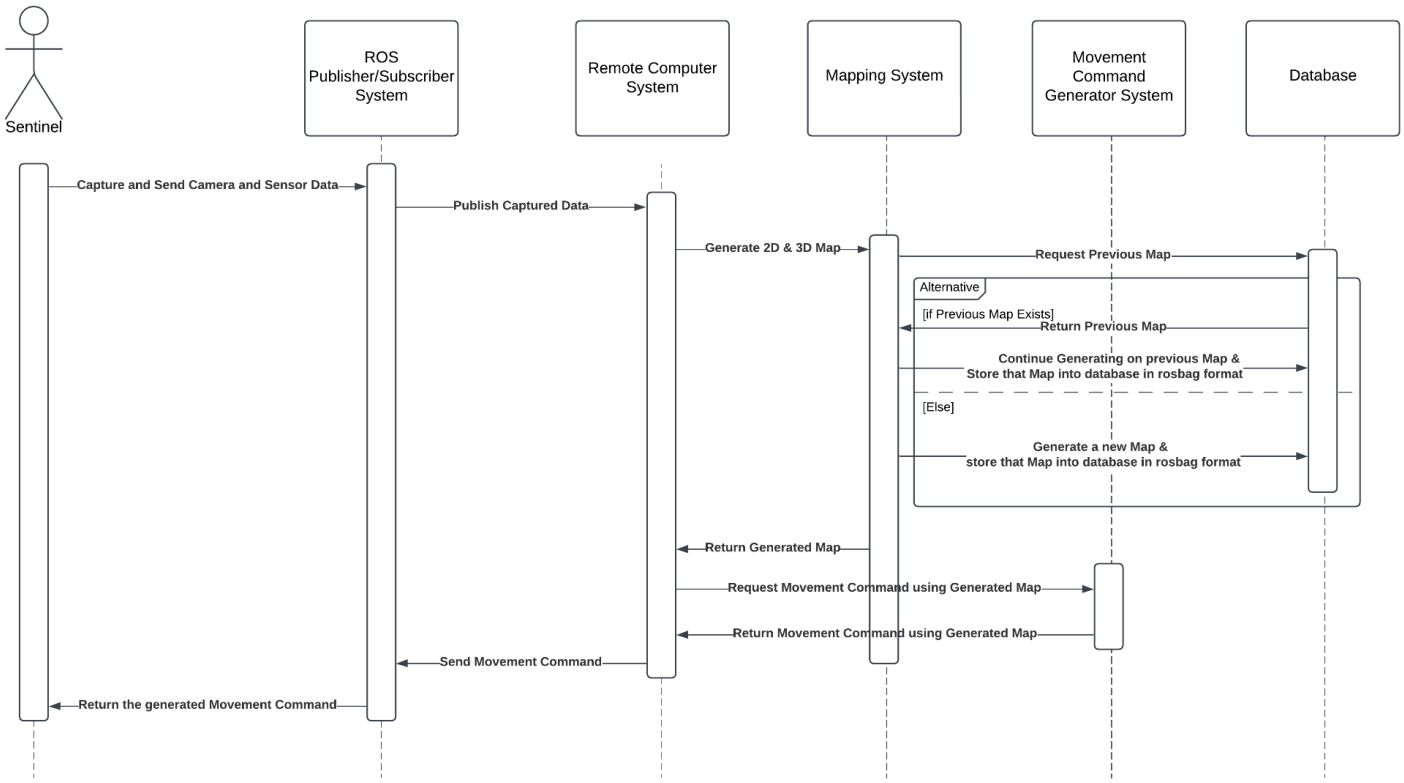
3.5.2. Data Stream



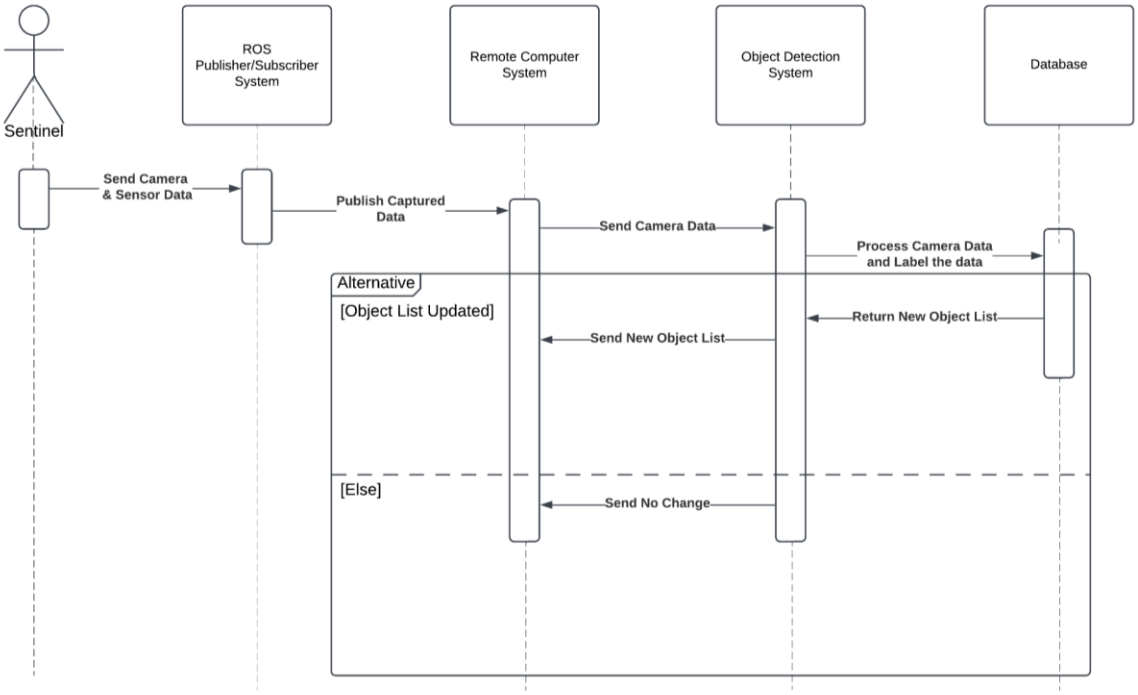
3.5.3. Manual Mapping



3.5.4. Autonomous Mapping

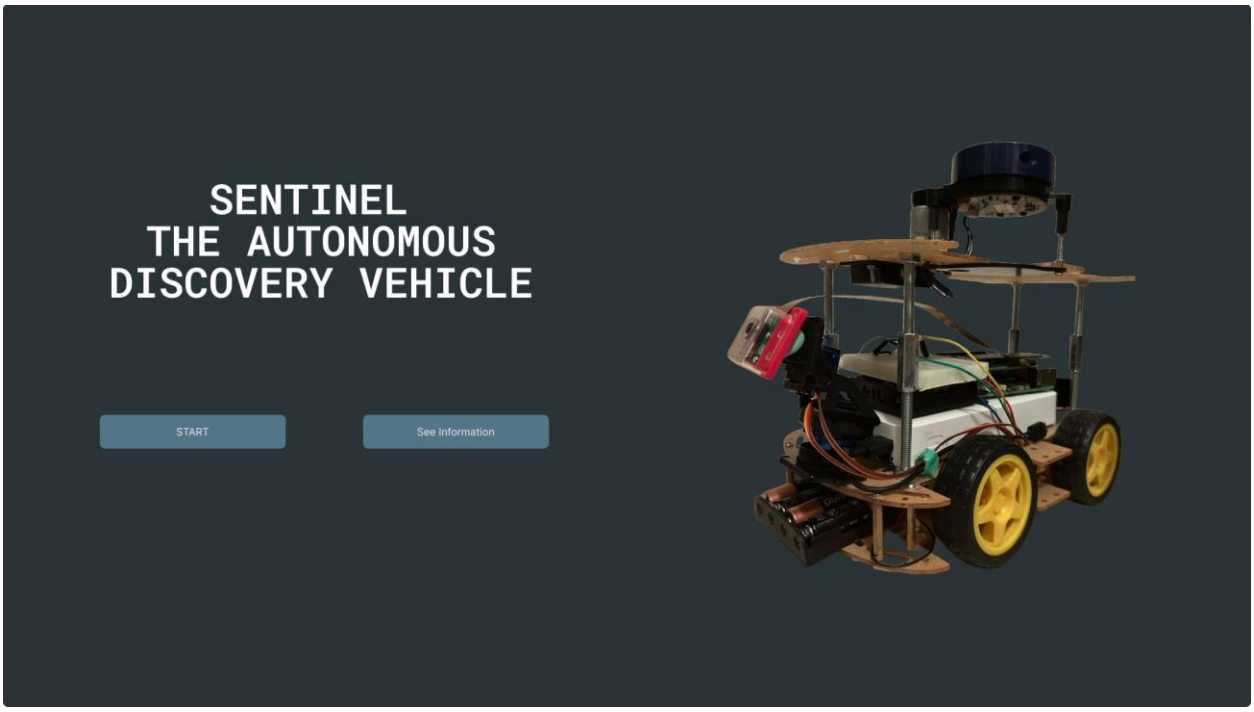


3.5.5. Object Detection

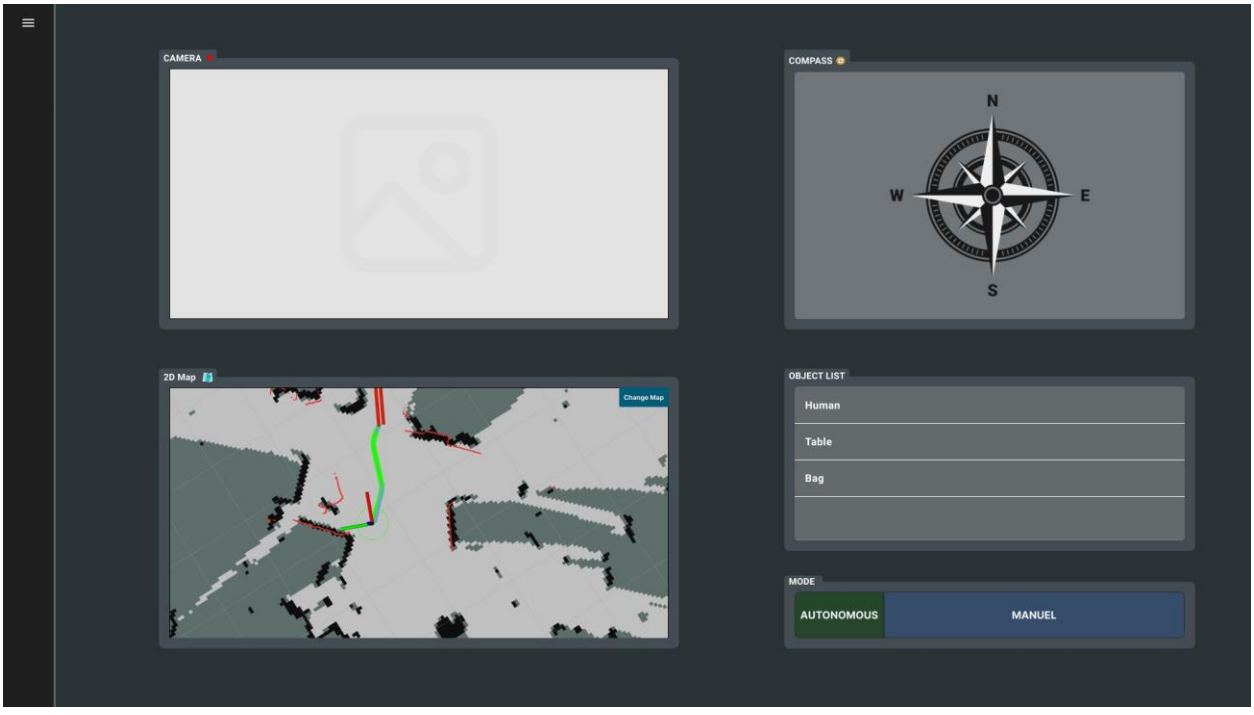


4. User Interfaces

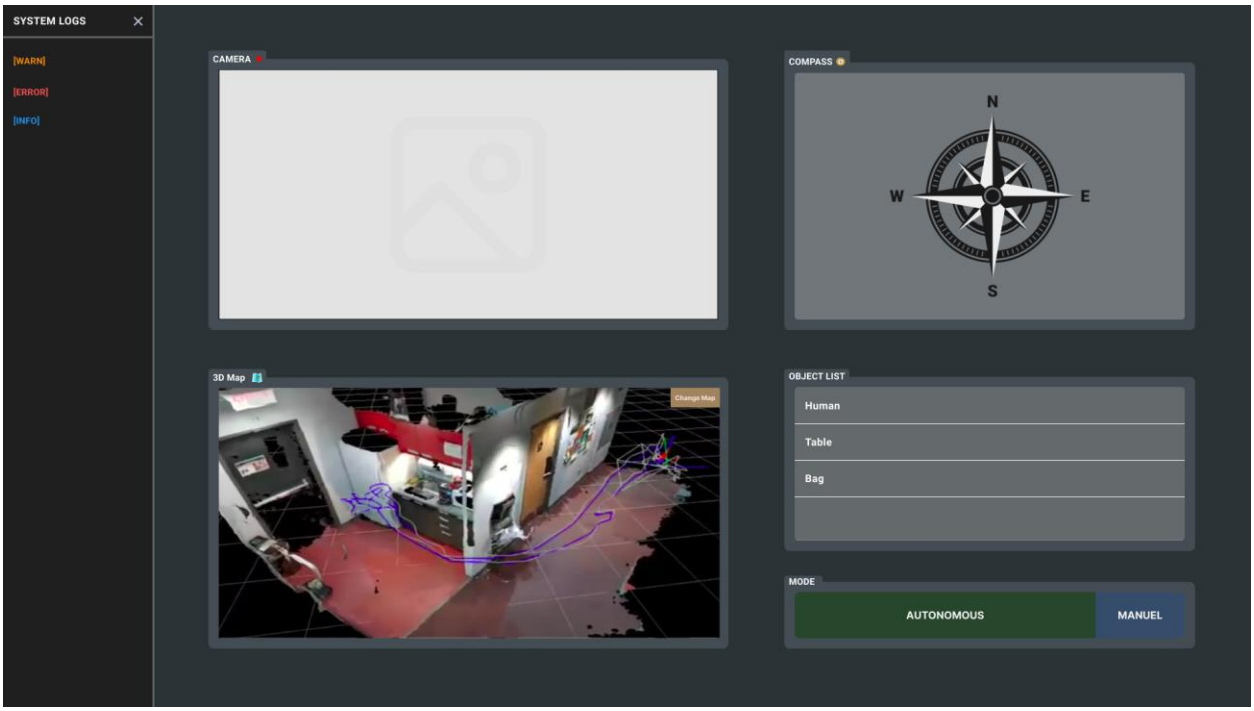
4.1. Home Page



4.2. Admin Dashboard with 2D Map



4.3. Admin Dashboard with 3D Map



4.4. Interactieve UI Link

[Interactieve User Interface on Figma]