TBA Case

Technical Design Document

version 1.1

Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Author | Reviewer | Date | Comment |
| 1 | Hamid Sahlolbey |  | 8/06/2019 | Initial Document |
| 1.1 | Hamid Sahlolbey |  | 12/06/2019 | Applying Changes |

0. Preface

This document is a general technical design document to guide the developers about the building blocks of the system.

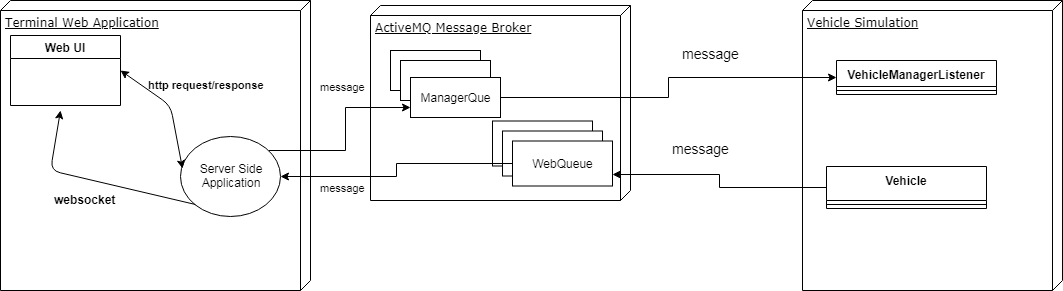
1. Introduction

As we can understand from system definition document it is a simulation, monitoring and controller of robotic vehicles.

2. System Architecture

In this section we will define the system component their responsibilities and how they interact.

2.1 System Block Diagram



**2.2 System Components**

The system is divided to 2 major components. As described in the next two sub-sections.

**2.2.1 Terminal Web Application:**

This is a web application responsible for interacting with user and sending user commands to "Vehicle Simulation" component. The communication between "Terminal Web Application" and "Vehicle Simulation" is via asynchronous messaging through ActiveMQ message broker.

This component receives the locations of moving vehicles through asynchronous messaging and sends them to web browser through websocket technology in real time.

**2.2.2 Vehicle Simulation**

This component is responsible for managing simulated vehicles. Its interaction with Terminal Web Application is through messaging and it simulates vehicle movement.

When the component receives create command it creates a vehicle and store it in memory. No persistent is implemented in this version as not requested in project definition.

When the component receives move command it finds the requested vehicle and ask it to move. The vehicle create a new thread and simulate the movement by calculating new locations. The calculated locations is sent to Terminal Web Application through asynchronous messaging each two seconds.

**3. Implementation Technologies**

**3.1 Spring Boot**

Spring boot will be used as integration and dependency injection framework.

**3.2 ActiveMq Classic**

ActiveMQ will be used as message broker. The AMQP method of messaging will be used as this platform independent.

**3.3 Websockets**

Websockets will be used for sending location information from web application server to web browser in real time.

**3.4 WEB UI Technology**

The WEB UI is simple HTML + Javascript+JQUERY+AJAX.