

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

– Ho Chi Minh, October 2022 –

**Table of Contents**

[I. Record of Changes 3](#_heading=h.gjdgxs)

[II. Project Management Plan 4](#_heading=h.30j0zll)

[1. Overview 4](#_heading=h.1fob9te)

[1.1 Scope & Estimation 4](#_heading=h.3znysh7)

[1.2 Project Objectives 4](#_heading=h.2et92p0)

[1.3 Project Risks 4](#_heading=h.tyjcwt)

[2. Management Approach 5](#_heading=h.3dy6vkm)

[2.1 Project Process 5](#_heading=h.1t3h5sf)

[2.2 Quality Management 5](#_heading=h.4d34og8)

[2.3 Training Plan 5](#_heading=h.2s8eyo1)

[3. Project Deliverables 5](#_heading=h.17dp8vu)

[4. Responsibility Assignments 5](#_heading=h.3rdcrjn)

[5. Project Communications 6](#_heading=h.26in1rg)

[6. Configuration Management 6](#_heading=h.lnxbz9)

[6.1 Document Management 6](#_heading=h.35nkun2)

[6.2 Source Code Management 6](#_heading=h.1ksv4uv)

[6.3 Tools & Infrastructures 6](#_heading=h.44sinio)

# I. Record of Changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 10/12/2022 | A | Nguyễn Huỳnh Nhật Minh | Added Scope & estimation |
| 10/23/2022 | M | Nguyễn Lê Thăng Long | Fixing Gramma issues |
|  |  |  |  |

\*A - Added; M - Modified; D - Deleted

# II. Project Management Plan

## 1. Overview

### 1.1 Scope & Estimation

| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man - days)** |
| --- | --- | --- | --- |
| ***1*** | ***AMWR Mobile Application*** |  |  |
| 1.1 | Control dc motor |  |  |
| 1.1.1 | Rotate forward | Simple |  |
| 1.1.2 | Rotate backward | Simple |  |
| 1.2 | Control robot’s directions |  |  |
| 1.2.1 | Move forward | Simple |  |
| 1.2.2 | Move backward | Simple |  |
| 1.2.3 | Move left | Simple |  |
| 1.2.4 | Move right | Simple |  |
| 1.2.5 | Turn left | Simple |  |
| 1.2.6 | Turn right | Simple |  |
| 1.3 | Show robot status |  |  |
| 1.3.1 | Show battery capacity | Simple |  |
| 1.3.2 | Show motor speed | Simple |  |
| 1.3.3 | Show product weight when loaded | Simple |  |
| 1.5 | Set motor speed |  |  |
| 1.5.1 | Synchronise speed at 35 RPM | Simple |  |
| 1.5.2 | Adjust speed when unloaded/loaded | Complex |  |
| 1.5.3 | Set speed manually | Medium |  |
| 1.6 | Device connections |  |  |
| 1.6.1 | Scan for bluetooth device | Medium |  |
| 1.6.2 | Show connection status | Medium |  |
| 1.6.3 | Choose device to connect | Medium |  |
| 1.7 | Layout tracking |  |  |
| 1.7.1 | Show loading area layout | Complex |  |
| 1.7.2 | Show robot path layout | Complex |  |
| 1.7.3 | Show real-time video from  loading area camera | Complex |  |
| ***2*** | ***AMWR*** |  |  |
| 2.1 | Path finding |  |  |
| 2.1.1 | Find path from starting point to space occupied by products and vice versa | Medium |  |
| 2.1.2 | Find path from starting point to space not occupied by products and vice versa | Medium |  |
| 2.1.3 | Find path from  space occupied by products to  space not occupied by products | Medium |  |
| 2.2 | Shortest path learning |  |  |
| 2.2.1 | Learn shortest path from starting point to space occupied by products and vice versa | Complex |  |
| 2.2.2 | Learn shortest path from starting point to space not occupied by products and vice versa | Complex |  |
| 2.2.3 | Learn shortest path from  space occupied by products to  space not occupied by products | Complex |  |
| 2.3 | Obstacle detection |  |  |
| 2.3.1 | Detect obstacle by front sensors | Simple |  |
| 2.3.2 | Detect obstacle by left/right side sensors | Simple |  |
| 2.3.3 | Detect obstacle by back sensors | Simple |  |
| 2.3.4 | Stop when obstacle detected | Simple |  |
| 2.4 | Product loading and unloading |  |  |
| 2.4.1 | Load product by lifting mechanism | Simple |  |
| 2.4.2 | Unload product by lifting mechanism | Simple |  |
| 2.5 | Adjust motor speed |  |  |
| 2.5.1 | Adjust motor speed when loaded | Medium |  |
| 2.5.2 | Adjust motor speed when unloaded | Medium |  |
| 2.5.3 | Adjust motor’s speed at intersection | Medium |  |
| 2.6 | Bluetooth communication |  |  |
| 2.6.1 | Connect to mobile app | Medium |  |
| 2.6.2 | Receive operator instructions | Complex |  |
| 2.6.3 | Send status to mobile app | Medium |  |

### 

### 1.2 Project Objectives

The main objective of this project is to build a prototype AGV robot that can handle loading and unloading products from one location to another while following a designated path. A mobile application is used to keep track of empty/occupied locations and to communicate with the robot via bluetooth.

* *Milestone Timeliness (%):* So far, the completion of the project is: 40%
* *Allocated Effort (man-days): 4 - 340*

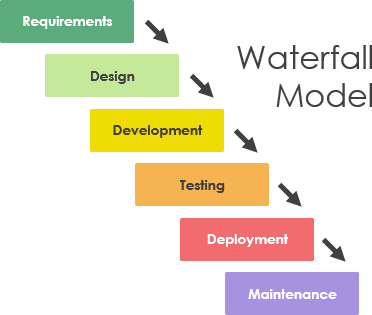
| **#** | **Testing Stage** | **Test Coverage** | **No. of Defects** | **% of Defect** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 1 | Reviewing | 0% | 0 | 0% | Project leader reviews code  before merging |
| 2 | Unit Test | 0% | 0 | 0% | Coder execute while coding |
| 3 | Integration Test | 0% | 0 | 0% | Tester execute when a  function is done |
| 4 | System Test | 0% | 0 | 0% | Tester execute when every require function is integrated |
| 5 | Acceptance Test | 0% | 0 | 0% | User verify the product |

### 1.3 Project Risks

| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| --- | --- | --- | --- | --- |
| 1 | Requirement changes | Medium | Medium | Accommodate new changes  while make sure it work with  other requirements |
| 2 | Scope creep | High | Medium | Have a meeting with the instructor  and discuss carefully the  best approach for the new scope |
| 3 | Code conflict when merge on Github | Medium | High | Project leader review team member code carefully before merge |
| 4 | Missing materials or wrong measurement when building robot | Low | Medium | Find and buy a new replacement and remeasure the robot carefully |
| 5 | Team member have an illness or absent and cannot complete tasks on time | Low | Low | Allocate other team members to fill in some of the tasks of absent member |
| 6 | Conflict between team member | High | Low | Project leader will step in as a middleman to resolve conflict and compromise if needed |
| 7 | Team member lack of knowledge on certain task | High | Medium | Have other team member support them in learning how to  do those tasks |

## 2. Management Approach

### 2.1 Project Process



We decided to use the Waterfall Model as an approach to manage and implement our project. The waterfall software process model is one of the simplest and most effective software process models. The waterfall model displays each step of the development process as a separate, sequential step so that it's easy to follow. You can view the development process from beginning to end with a glance and adjust steps as the development process continues.

The waterfall process model works well for simple software development projects that likely won't change significantly during the development process. The team understands each parameter thoroughly and can work through the development phases quickly.

### 2.2 Quality Management

* Defect Prevention:
  + If any defect is found, the related person must be notified immediately.
  + Defects must be carefully evaluated such as "How bad is the defect and can it damage the system?", "How long is the time to fix that defect?".
  + The deadline for fixing the defect must be specified clearly.
* Reviewing:
  + The project leader must be honest and not biassed towards any of the project members.
  + Defects should be documented with details such as priority, time to fix, and their effect on the project.
  + The person responsible for defects found must-have solutions to fix the defect as quickly as possible.
* Unit Testing:
  + The tester must prepare the test cases carefully and accurately and must not ignore any cases.
  + Test cases should cover both hardware modules and software modules.
  + Defects should be documented with details such as priority, time to fix, and their effect on the project.
  + The person responsible for defects found must-have solutions to fix the defect as quickly as possible.
* Integration Testing:
  + The tester must prepare the test cases carefully and accurately and must not ignore any cases.
  + Test cases for hardware modules should … \*
  + Test cases for software modules should … \*
  + Defects should be documented with details such as priority, time to fix, and its effect on the project.
  + The person responsible for defects found must-have solutions to fix the defect as quickly as possible.
  + Each hardware module responds smoothly and on time.
* System Testing:
  + The tester must prepare the test cases carefully and accurately and must not ignore any cases.
  + Test cases should cover the entire system functionality and the communication under development with external systems.
  + Defects should be documented with details such as priority, time to fix, and their effect on the project.
  + The person responsible for defects found must-have solutions to fix the defect as quickly as possible.
* Acceptance Testing:
  + The tester must prepare the test cases carefully and accurately and must not ignore any cases.
  + Test cases must match user requirements.
  + Defects should be documented with details such as priority, time to fix, and their effect on the project.
  + The person responsible for defects found must-have solutions to fix the defect as quickly as possible.

### 2.3 Training Plan

| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| --- | --- | --- | --- |
| Java-based mobile app coding convention | TuanCQ, LongNLT | 19/09/2022 - 24/09/2022 | Mandatory |
| Github | MinhNHN, TuanCQ, LongNLT, MinhNH | 19/09/2022 - 24/09/2022 | Mandatory |
| Arduino coding convention | MinhNHN | 26/09/2022 - 01/10/2022 | Mandatory |

## 3. Project Deliverables

*[Given the main project deliverables. Those can be internal and/or external deliverables. Students can prepare master schedule in the table format as below or in the more detailed structure as the sample in the attached sample file -* ***Report2\_Sample High Level Project Schedule.pdf****]*

| **#** | **Deliverable** | **Due Date** | **Notes** |
| --- | --- | --- | --- |
| 1 | … | dd/MM/yyyy | … |

## 4. Responsibility Assignments

*[Describe the main responsibilities in your project (to complete the outputs as defined in the above section), in the format as the sample below].*

*D - Do; R - Review; S - Support; I - Informed; <blank> - Omitted*

| **Responsibility** | **MinhNHN** | **TuanCQ** | **MinhNH** | **LongNLT** |
| --- | --- | --- | --- | --- |
| Project planning & tracking | D | S | S | R |
| Prepare project report 1 | D | S | S | R |
| Prepare project report 2 | D | S | S | R |
| Prepare project report 3 | D | S | S | R |
| Prepare project report 4 | D | S | S | R |
| Prepare project report 5 | D | S | S | R |
| Prepare project report 6 | D | S | S | R |
| Prepare project final report | D | S | S | R |
| Prepare Use Case Diagram | S | S | D | I |
| Prepare Context Diagram | D | I | I | I |
| Prepare ERD | D | R | R | R |
| Design UI for mobile app | R | D | I | S |
| Design Database | S | D | S | I |
| Find robot materials | D | R | S | I |
| Build robot | S | S | D | I |
| Code robot functions | D | R | S | I |
| Code mobile app functions | R | S | I | D |
| Prepare test cases | D | S | S | I |
| Do testing | R | D | S | S |

## 5. Project Communications

| **Communication** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| --- | --- | --- | --- | --- |
| Kick-off meeting | Project team members and instructor | - Introduce project.  - Define objectives  - Find a suitable approach. | Once at the start of the project. | In-person meeting |
| Project progress review | Project team members and instructor | - Review project progress.  - Discuss the next steps.  - Review documents. | Once a week | In-person meeting or Google Meet |
| UI design planning | Project team members | - Prepare mobile app  UI layout. | Once at the start of the UI design planning phase | Google Meet |
| Build and fix robot | Project team members | - Build robot  - Fix robot if defect | Every Monday, Wednesday, and Friday | In-person meeting |
| Test robot | Project team members | - Test robot functions  - Test robot to mobile app communication | Every Monday, Wednesday and Friday | In-person meeting |

## 6. Configuration Management

### 6.1 Document Management

*[Describe how you would manage project documents & their changes/versions]*

### 6.2 Source Code Management

*[Describe how you would manage project source codes & their changes/versions]*

### 6.3 Tools & Infrastructures

| **Category** | **Tools / Infrastructure** |
| --- | --- |
| **Technology** | Flutter (FrontEnd), Android Java (App/Backend) |
| **Database** | FireBase Real-time Database |
| **IDEs/Editors** | Android Studio, Arduino IDE |
| **Diagramming** | DrawIO |
| **Documentation** | Ms Office, Google Docs/Sheets/Slides |
| **Version Control** | GitHub (Source Codes), Google Drive (Documents) |
| **Deployment server** | N/A |
| **Deployment Enviroment** | Android |
| **Project management** | Zalo (Schedule), GitHub (Tasks, Defects) |