

**CAPSTONE PROJECT REPORT**

**Report 3 – Software Requirement Specification**

– Ho Chi Minh, October 2022 –

**Table of Contents**

[I. Project Report 3](#_heading=h.1ksv4uv)

[1. Status Report 3](#_heading=h.44sinio)

[2. Team Involvements 3](#_heading=h.2jxsxqh)

[3. Issues/Suggestions 3](#_heading=h.z337ya)

[II. Software Requirement Specification 4](#_heading=h.30j0zll)

[1. Overall Description 4](#_heading=h.30j0zll)

[1.1 Product Overview 4](#_heading=h.30j0zll)

[1.2 Business Rules 5](#_heading=h.35nkun2)

[2. User Requirements 6](#_heading=h.d9pnlmo5desl)

[2.1 System Actors 6](#_heading=h.3znysh7)

[2.2 Use Cases 6](#_heading=h.bztj4zlp7cu6)

[3. Functional Requirements 8](#_heading=h.tyjcwt)

[3.1 System Functional Overview 8](#_heading=h.1y810tw)

[3.2 <<Feature Name 1>> 10](#_heading=h.3dy6vkm)

[3.3 <<Feature Name 2>> 10](#_heading=h.1t3h5sf)

[4. Non-Functional Requirements 11](#_heading=h.4d34og8)

[4.1 External Interfaces 11](#_heading=h.2s8eyo1)

[4.2 Quality Attributes 12](#_heading=h.17dp8vu)

[5. Other Requirements 14](#_heading=h.4i7ojhp)

[5.1 Appendix1 - Messages List 14](#_heading=h.2xcytpi)

[5.2 Appendix2 - … 14](#_heading=h.1ci93xb)

[5.3 … 14](#_heading=h.3whwml4)

# I. Record of Changes

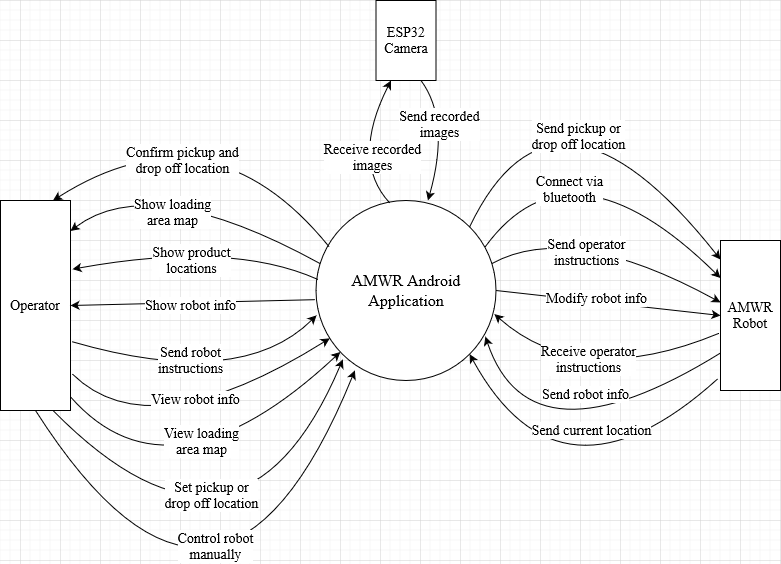
| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 13/10/2022 | A | Nguyễn Huỳnh Nhật Minh | Create initial draft |
| 22/10/2022 | M | Nguyễn Huỳnh Nhật Minh | Modify Context Diagram, Use Case, Screen Flow description |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Software Requirement Specification

## 1. Product Overview

AMWR is a prototype robot similar to the AGV robot. It will use multiple IR sensors as a line detector, a small lifting table, and a simple navigation system to automate the work of lifting products and move them to the designated locations. It will be accompanied by an android application to keep track of empty/occupied space and also the map of the loading area. The android app will provide an optimal path, then the robot will follow the line to its designated location to do its job.

****

***AMWR System Context Diagram***

## 2. User Requirements

### 2.1 Actors

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Operator | An operator is any user who controls the AMWR Robot via the AMWR android application. The operator can send instructions, view robot info, view the loading area map, and send pickup or drop-off locations. The operator can also view real-time images of the loading area map recorded by the ESP32 Camera. |
| 2 | AMWR Robot | AMWR Robot is the robot that the operator control via the AMWR android application. The robot can receive instructions from the operator to include pickup or drop-off of products at a specific location. The robot can also let the operator view its info like connection status, speed, battery capacity, and current location. |
| 3 | ESP32 Camera | ESP32 Camera is responsible for recording images of the loading area map and sending them to the android application to process. The recording is used to keep track of the location of the product as well as the robot's current location. |

### 

### 2.2 Use Cases

#### 2.2.1 Diagram(s)

#### 2.2.2 Descriptions

| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| --- | --- | --- | --- |
| 1 | Send robot instructions | Operator | This use case allows the operator to send instructions to a robot to the product locations. Only activate when choosing pickup location and dropoff location use cases have been performed. |
| 2 | Choose pickup location | Operator | This use case allows the operator to choose a pick-up product location for the robot. |
| 3 | Choose drop-off location | Operator | This use case allows the operator to choose to drop off product location for the robot. |
| 4 | View robot info | Operator | This use case allows the operator to view robot information such as: name, bluetooth connection status, and bluetooth password. |
| 5 | Modify robot info | Operator | This use case allows the operator to modify robot info such as: update name, and update bluetooth password. |
| 6 | View loading area map | Operator | This use case allows the operator to view the map of the loading area. |
| 7 | View real-time operation | Operator | This use case allows the operator to view the current operation that carries out by the robot. It also shows the current robot speed and battery capacity. |
| 8 | View real-time images | Operator | This use case allows the operator to view the real-time images that is being recorded by the ESP32 Camera. |
| 9 | View robot current status | Operator | This use case allows the operator to view the current robot status such as: current speed, current battery capacity, current location. |
| 10 | View product current location | Operator | This use case allows the operator to view the product's current location on the loading area map before and after the product has been by the robot. |
| 11 | Control robot manually | Operator | This use case allows the operator to take control of the robot and control it to move in a specific direction. Also, the operator can adjust the speed of the robot. |
| 12 | Connect via bluetooth | AMWR Robot, ESP32 Camera | This use case allows the AMWR Robot and the ESP32 Camera to connect to an android application via bluetooth |
| 13 | Receive operator instructions | AMWR Robot | This use case allows the AMWR Robot to receive instructions from operator |
| 14 | Send robot info | AMWR Robot | This use case allow AMWR Robot to send its info to the android application via bluetooth |

## 3. Functional Requirements

### 3.1 System Functional Overview

#### 3.1.1 Screens Flow

#### 3.1.2 Screen Descriptions

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Show loading area map | Loading Area map | Show a map of the loading area and real-time images |
| 2 | Show real-time images | Loading Area map | Show the real-time images of the loading area camera that is being captured by the ESP32 camera. |
| 3 | Show robot current location | Loading Area map | Show a red dot indicating the location of the robot on the map. |
| 4 | Show product current location | Loading Area map | Show which space is empty and which is occupied on the loading area map. |
| 5 | Choose pickup or drop off location | Loading Area map | Allow the operator to choose which location the robot can pick up and drop off the product. |
| 6 | Show device connection status | Device Connection | Show the status of a device whether it is connected or not. |
| 7 | Scan for bluetooth device | Device Connection | Allow operator to scan for available bluetooth devices |
| 8 | Take control of the robot | Manual Control | Allow operator to take control of the robot to control manually |
| 9 | Show list of robot | Manual Control | Show a list of robots that are available |
| 10 | Control in robot difference directions | Control Panel | Allow operator to control the robot in specific directions |
| 11 | Adjust robot speed | Control Panel | Allow operator to adjust the speed of the robot on the fly |
| 12 | Show list of robot | Robot Info | Show a list of robot that available |
| 13 | Show robot detail infos | Robot Details | Show robot detail informations such as: name, bluetooth connection, bluetooth password, speed, battery capacity |
| 14 | Modify robot info | Robot Details | Allow operator to modify robot info such as: set name, update name, set bluetooth password, update bluetooth password |

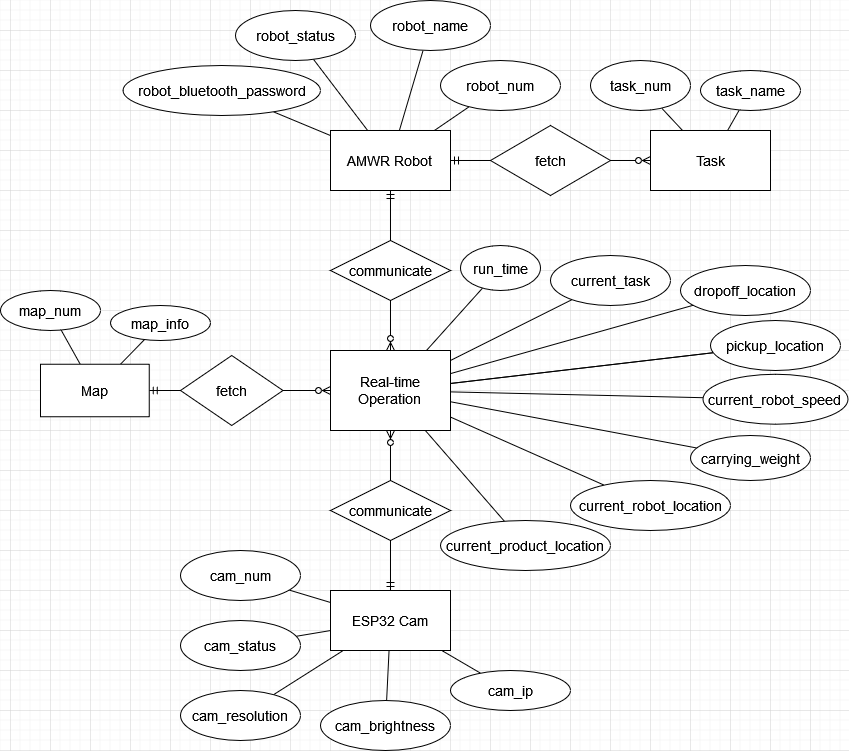
#### 3.1.3 Screen Authorization

| **Screen** | **Operator** |
| --- | --- |
| Home Screen | X |
| Loading Area map | X |
| Show loading area map | X |
| Show real-time images | X |
| Show robot current location | X |
| Show product current location | X |
| Choose pickup and drop off location | X |
| Device Connection | X |
| Show device connection status | X |
| Scan for bluetooth devices | X |
| Manual Control | X |
| Show list of robot | X |
| Take control of the robot | X |
| Control Panel | X |
| Control robot in difference directions | X |
| Adjust robot speed | X |
| Robot Info | X |
| Show list of robot | X |
| Robot Details | X |
| Show robot name | X |
| Show robot battery status | X |
| Show robot speed | X |

#### 3.1.4 Non-Screen Functions

| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | Bluetooth connection | Reduce latency, improve signal strength, increase signal range | Implement Bluetooth 5.0 technology |
| 2 | images processing | Process loading area images recorded from ESP32 Camera. Determine the current location of the product and robot. | Implement images processing algorithm |

#### 3.1.5 Entity Relationship Diagram



**Entities Description**

| **#** | **Entity** | **Description** |
| --- | --- | --- |
| 1 | AMWR Robot | Information of the robot. |
| 2 | Map | Map information of the loading area |
| 3 | Task | Include tasks that the robot can carry out. |
| 4 | ESP32 Camera | Recorded images of the loading area. |
| 5 | Real-time Operation | Show information of loading area map, robot info, and products location in real-time |

### 3.2 Loading Area map

#### 3.2.1 Show loading area map

*[A function can be a screen or a non-screen function (listed in the part 3.1.5 above). In this part, you need to provide the details on the related function, focus on mentioning below information*

*• Function trigger: how this function is triggered (navigation path, a timing frequency, etc.*

*• Function description: actors/roles, purpose, interface, data processing, etc.*

*• Screen map: mock-up prototype of the screen, sample below is for Manage Products screen*

**

Function Details: provide explanation for the data, validation, business rules, functionalities (for both normal cases and abnormal cases), etc. of the function so that the reader can imagine how it work.

* Function trigger: this function is triggered when the operator navigates from the Home screen to the Loading Area map screen.
* Function description:
  + Actor: Operator.
  + Purpose: Show a map of the loading area and real-time images.
  + Interface: Android application.
  + Data: Map info, robot info, robot location, product location, camera images
* Screen map:
* Function Details:

#### 3.2.2 Show real-time images

* Function trigger: this function is triggered when the operator navigates from the Home screen to the Loading Area Map screen.
* Function description:
  + Actor: Operator.
  + Purpose: Show the real-time images of the loading area camera that is being captured by the ESP32 camera.
  + Interface: Android application.
  + Data: camera images.
* Screen map:
* Function Details:

#### 3.2.3 Show robot current location

* Function trigger: this function is triggered when the operator navigates from the Home screen to the Loading Area Map screen.
* Function description:
  + Actor: Operator.
  + Purpose: Show a red dot indicating the location of the robot on the map.
  + Interface: Android application.
  + Data: robot current location.
* Screen map:
* Function Details:

#### 3.2.4 Show product current location

* Function trigger: this function is triggered when the operator navigates from the Home screen to the Loading Area Map screen.
* Function description:
  + Actor: Operator.
  + Purpose: Show which space is empty and which is occupied on the loading area map.
  + Interface: Android application.
  + Data: camera images.
* Screen map:
* Function Details:

#### 3.2.5 Choose pickup and drop off location

* Function trigger: this function is triggered when the operator chooses the pickup and drop off location on the map.
* Function description:
  + Actor: Operator.
  + Purpose: Allow the operator to choose which location the robot can pick up and drop off the product.
  + Interface: Android application.
  + Data: pickup location, drop off location.
* Screen map:
* Function Details:

### 3.3 Device connection

#### 3.3.1 Show device connection status

* Function trigger: this function is triggered when the operator navigates to the Device connection screen from the Home screen.
* Function description:
  + Actor: Operator.
  + Purpose: Show the status of a device whether it is connected or not.
  + Interface: Android application.
  + Data: bluetooth connection status.
* Screen map:
* Function Details:

#### 3.3.2 Scan for bluetooth device

* Function trigger: this function is triggered when the operator chooses the scan bluetooth device button.
* Function description:
  + Actor: Operator.
  + Purpose: Allow operator to scan for available bluetooth devices.
  + Interface: Android application.
  + Data: bluetooth connection status.
* Screen map:
* Function Details:

### 3.4 Manual Control

#### 3.4.1 Take control of the robot

* Function trigger: this function is triggered when the operator press confirm when choosing the robot on the Manual Control robot screen.
* Function description:
  + Actor: Operator.
  + Purpose: Allow operator to take control of the robot to control manually
  + Interface: Android application.
  + Data: robot info.
* Screen map:
* Function Details:

#### 3.4.2 Show list of robot

* Function trigger: this function is triggered when the operator navigates from Home screen to Manual Control screen.
* Function description:
  + Actor: Operator.
  + Purpose: Show a list of robots that are available.
  + Interface: Android application.
  + Data: robot info.
* Screen map:
* Function Details:

### 3.5 Control Panel

#### 3.5.1 Control in robot difference directions

* Function trigger: this function is triggered when the operator press directional arrows on the Control Panel screen.
* Function description:
  + Actor: Operator.
  + Purpose: Allow the operator to control the robot in specific directions.
  + Interface: Android application.
  + Data: N/A.
* Screen map:
* Function Details:

#### 3.5.2 Adjust robot speed

* Function trigger: this function is triggered when the operator drags the speed value slider on the Control Panel screen.
* Function description:
  + Actor: Operator.
  + Purpose: Allow the operator to adjust the speed of the robot on the fly.
  + Interface: Android application.
  + Data: robot current speed.
* Screen map:
* Function Details:

### 3.6 Robot Info

#### 3.6.1 Scan for bluetooth device

* Function trigger: this function is triggered when the operator chooses the specific robot on the robot list
* Function description:
  + Actor: Operator.
  + Purpose: Show the status of the robot information.
  + Interface: Android application.
  + Data: bluetooth connection status.
* Screen map:
* Function Details:

### 

### 3.7 Robot Details

#### 3.7.1 Show robot detail Infos

* Function trigger: this function is triggered when the operator knows the status of the robot.
* Function description:
  + Actor: Operator.
  + Purpose: Show the status of the robot information.
  + Interface: Android application.
  + Data: bluetooth connection status.
* Screen map:
* Function Details:

#### 3.7.2 Modify robot info

* Function trigger: this function is triggered when the operator allows modifying robot info.
* Function description:
  + Actor: Operator.
  + Purpose: Allow operator to modify robot info.
  + Interface: Android application.
  + Data: bluetooth connection status.
* Screen map:
* Function Details:

## 4. Non-Functional Requirements

### 4.1 External Interfaces

*[This section provides information to ensure that the system will communicate properly with users and with external hardware or software/system elements.]*

The application is usually used with Bluetooth connection with the ARWM robot in a specific location which is in the loading area. The connection with the ARWM robot is required to gain the best experience using the application. The application can be used in the Android OS system.

### 4.2 Quality Attributes

*[List all the required system characteristics (quality attributes) specification. Some of the possible attributes are provided with the guide/descriptions are mentioned here]*

#### 4.2.1 Usability

*[This section includes all those requirements that affect usability. For example, specify the required training time for a normal users and a power user to become productive at particular operations; specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like; specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]*

* The training time for the system have different requirements for a different type of user. For a normal user, it’ll take about 15 minutes, and for a power user, it’ll take about 5 minutes to become productive on all operations.
* For each task, the time needed to perform can be between 5ms to 10ms.
* There will be error messages to notify the operator what task can’t be performed. The user can back out and choose another option easily.
* When the operator wants to take control of the robot or update its information, a dialog box will pop up to confirm.
* The interface is easy to learn and navigate allowing the operator to accomplish their goals in 3 clicks.
* Error messages will be easy to understand and can be used to instruct the operator on how to correct them.

#### 4.2.2 Reliability

*[Requirements for reliability of the system should be specified here. Some suggestions follow:*

*Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.*

Availability: The android application will run until that phone/tablet runs out. The ARWM robot has a use time of 90 minutes.

*Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.*

*Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?*

*Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.*

*Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).*

*Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]*

#### 4.2.3 Performance

*[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.*

*Response time for a transaction (average, maximum)between put, for example, transactions per second*

*Capacity, for example, the number of customers or transactions the system can accommodate*

*Resource utilization, such as memory, disk, communications, and so forth.]*

* Response time of the robot is vary from 100ms to 50ms.
* Bluetooth pairing is between 5 - 10 seconds depending on your android device.

## 5. Requirement Appendix

*[Provide business rules, common requirements, or other extra requirements information here]*

### 5.1 Business Rules

*[Provide common business rules that you must follow. The information can be provided in the table format as the sample below]*

| **ID** | **Rule Definition** |
| --- | --- |
| BR-01 | The limit carrying weight of the robot is 10 kilograms. |
| BR-02 | The robot must follow a line to navigate throughout the journey. |
| BR-03 | All meals in a single order must be delivered to the same location. |
| BR-04 | All meals in a single order must be paid for by using the same payment method. |
| BR-11 | If an order is to be delivered, the patron must pay by payroll deduction. |
| BR-12 | Order price is calculated as the sum of each food item price times the quantity of that food item ordered, plus applicable sales tax, plus a delivery charge if a meal is delivered outside the free delivery zone. |
| BR-24 | Only cafeteria employees who are designated as Menu Managers by the Cafeteria Manager can create, modify, or delete cafeteria menus. |
| BR-33 | Network transmissions that involve financial information or personally identifiable information require 256-bit encryption. |
| BR-86 | Only regular employees can register for payroll deduction for any company purchase. |
| BR-88 | An employee can register for payroll deduction payment of cafeteria meals if no more than 40 percent of his gross pay is currently being deducted for other reasons. |

### 5.2 Common Requirements

*[Fill all the common requirements here..]*

### 5.3 Application Messages List

| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| --- | --- | --- | --- | --- |
| 1 | MSG01 | In line | There is not any search result | *No search results.* |
| 2 | MSG02 | In red, under the text box | Input-required fields are empty | *The \* field is required.* |
| 3 | MSG03 | Toast message | Updating asset(s) information successfully | *Update asset(s) successfully.* |
| 4 | MSG04 | Toast message | Adding new asset successfully | *Add asset successfully.* |
| 5 | MSG05 | Toast message | Confirming email of asset hand-over is sent successfully | *A confirmation email has been sent to {email\_address}.* |
| 6 | MSG06 | Toast message | Resetting asset information successfully | *Return asset(s) successfully.* |
| 7 | MSG07 | Toast message | Deleting asset information successfully | *Delete asset(s) successfully.* |
| 1 | MSG01 | In line | There is not any search result | *No search results.* |
| 2 | MSG02 | In red, under the text box | Input-required fields are empty | *The \* field is required.* |

### 5.4 Other Requirements…