Table of Contents

1. [Purpose 2](#_30j0zll)
   1. [Intended Audience 2](#_3znysh7)
   2. [Intended Use 2](#_tyjcwt)
   3. [Scope 2](#_1t3h5sf)
   4. [Definitions and Acronyms 2](#_2s8eyo1)
2. [Overall System Description 3](#_3rdcrjn)
   1. [Use Case Diagrams 3](#_lnxbz9)
   2. [System Architecture 4](#_1ksv4uv)
   3. [Functional Requirements](#_2jxsxqh) 6
      1. [Start Up and Main Menu](#_3j2qqm3) 6
      2. [Dispense Drink](#_4i7ojhp) 7
      3. [Drink Payment](#_1ci93xb) 8
      4. [Anti-Burglar System](#_2bn6wsx) 9
      5. [Remote Monitoring 9](#_3as4poj)
      6. [Servicing and Restocking 1](#_jah60nttyqed)1
   4. [Non-Functional Requirements 1](#_49x2ik5)2
      1. [Vending Machine Management 1](#_147n2zr)2
3. [Software Architecture 1](#_23ckvvd)3
   1. [Static Software Architecture 1](#_32hioqz)3

| **Name:** | **Contributions:** |
| --- | --- |
| Drew | 1. Burglar\_system and passcode 2. Merging all functions into main.py 3. Scrum master 4. Bug fixes 5. Docker 6. Website template |
| Yue Hang | 1. 70% of Testwebsite.py 2. Main\_menu.py 3. Some bug fixes 4. Software pytest 5. Stock json file. |
| Manbir | 1. Dispensing of drinks 2. Restocking 3. Update stock to json file |
| Zi Qiong | 1. payment.py 2. 30% of Testwebsite.py 3. Sales json file. |

# Purpose

## Intended Audience

This SRS document describes the System Requirements and Software Design for a Smart Vending Machine and the target audience are System and Software Engineers working on the development of this project.

## Intended Use

The SRS defines the overall System Architecture and Requirements as well as the Software Architecture and Design. This document also contains the definition of the System Requirements which shall be used as the input for System Test cases and Software Unit Test cases.

## Scope

**Brief Project Description**

* Smart Vending Machine which allows customers to purchase a drink physically from the machine and pay using an RFID Tag and Scanner. It includes an anti-burglary system which detects whether the machine is being forcefully opened. A valid user code can be used to open the vending machine without triggering the alarm in order to service or restock.

**Components Used:**

* RFID scanner
  + Used to simulate payment for the drinks
* LCD
  + Used as a display
* Keypad
  + Used as input for drink selection and password input for servicing
* Buzzer
  + Alarm when burglary is detected (attempt in forcefully opening the machine)
* Servo Motor
  + Used to dispense drinks
* 3 Axis Accelerometer
  + Detects whether the machine is being forcefully opened

**Timeline**

**Sprint 4:**

* Write code main body (Make a vending machine) :
  + Start Up and Main Menu (Yue Hang)
  + Dispense Drinks (Manbir)
  + Drink Payment (Zi Qiong)
  + Set-up RFID Database (Zi Qiong)
* Fix Bugs

**Sprint 5:**

* Code :
  + Burglar system (Drew)
  + Remote Monitoring (All of us)
  + Servicing and Restocking (Manbir)
* Create test cases for codes in Sprint 4
* Test the test cases above and fix any bugs

**Sprint 6:**

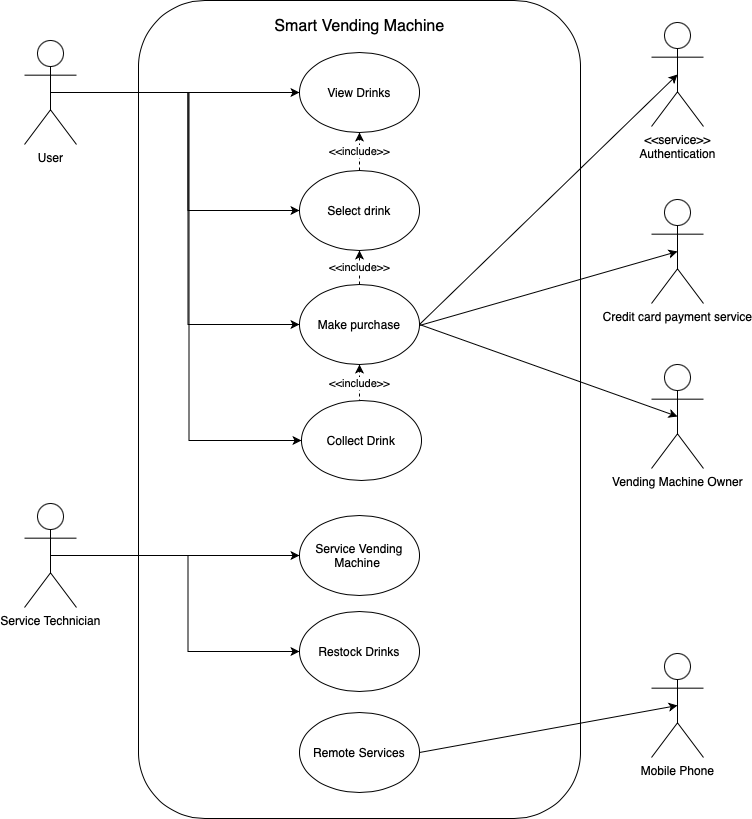
* Create test cases for codes in Sprint 5
* Fix more Bugs
* Test the entire system together

## Definitions and Acronyms

| **Acronym** | **Description** |
| --- | --- |
| RFID | Radio-Frequency IDentification |
| LED | Light Emitting Diode |
| LCD | Liquid Crystal Display |
| SW | Software |
| HW | Hardware |

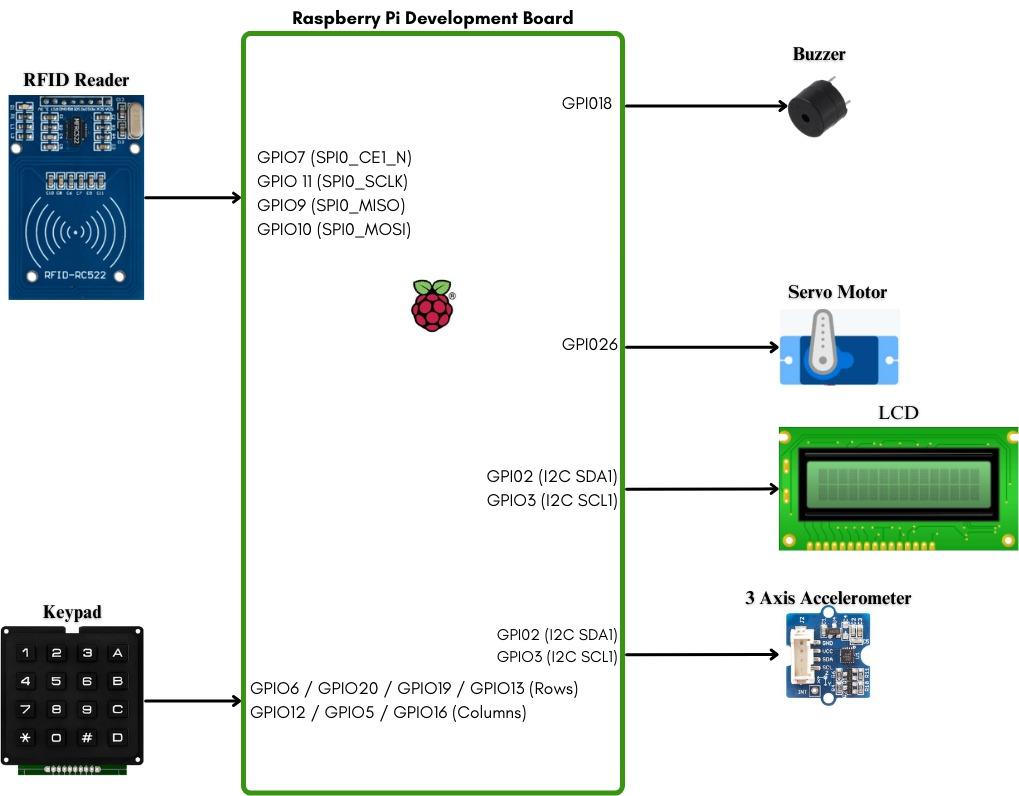
# Overall System Description

## Use Case Diagrams



## System Architecture

The System Architecture

****

## Functional Requirements

### Start Up and Main Menu

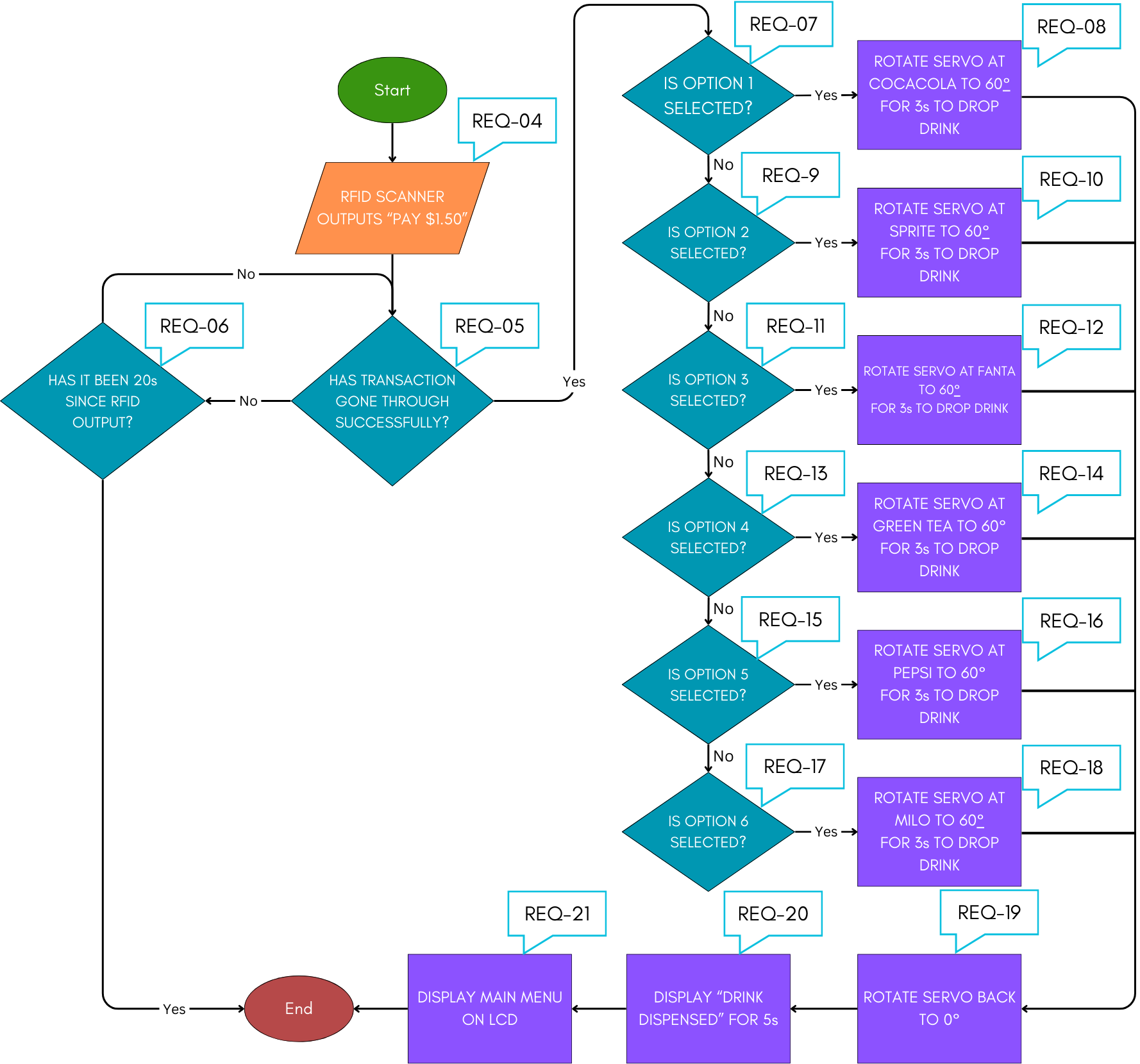
This is how the smart vending machine will work from start up, showing the main menu.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-01 | When the Vending Machine is first powered ON, the main menu with the text below shall be displayed on the LCD screen  Line 1 = “1.Coca-Cola”  Line 2 = “2.Sprite”  After 5 seconds, the main menu with the text below shall be displayed on the LCD screen  Line 1 = “3. Fanta”  Line 2 = “4. Green Tea”  After another 5 seconds, the main menu with the text below shall be displayed on the LCD screen  Line 1 = “5. Pepsi”  Line 2 = “6. Milo”  It will loop every 5 seconds. |
| REQ-02 | In the main menu defined in REQ-01, if any option from 1-6 is selected on the keypad, will go to REQ-22. |

### Dispense Drink

This is the process of how the drink is dispensed after selection has been made by the user.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-03 | From the main menu, if the user selects “1.Beverage”and selects an option from 1 to 6, then the flowchart defined in Figure 1 shall be implemented. |



#### Figure 1

### Drink Payment

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-22 | If the RFID reader detects a RFID card that has been registered in the internal database then the vending machine will go to REQ-03 |
| REQ-23 | Deduct $1.50 from the specific RFID database for that ID tag after the order has been  processed. |
| REQ-24 | Once payment is made, the sale will be updated on the website. |

### Anti-Burglar System

The Smart Vending Machine includes an integrated Anti-Burglar System which triggers an alarm and notifies the owner when there has been an attempt of the vending machine being forcefully opened.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-25 | Check the status of the accelerometer, whether it's shaking (vibrating) or if it's in a stationary state. |
| REQ-26 | If the accelerometer detects shaking (vibrating), it would trigger an alarm using a buzzer and send a notification to the owner through SMS (Telegram). |
| REQ-27 | During an alarm, the buzzer will be activated based on the timing below |
| REQ-28 | If the accelerometer is in a stationary state (does not detect shaking), nothing should happen. Operation will continue as normal. |

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### Servicing and Restocking

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-29 | To restock, user will need to press “#”. After pressing “#”, they will be directed to REQ-30. |
| REQ-30 | The user will then be prompted to enter the password. |
| REQ-31 | If the password entered is correct :   * Enter amount restocked   After entering the amount, the LCD will display “Restocking done.” |
| REQ-32 | If the password entered is incorrect :   * LCD will display:   + Line 1: “Wrong Pin!”   + Line 2: “Pls Try Again” * Vending machine will return to main menu |

### Remote Monitoring

The Smart Vending Machine supports “Remote Monitoring” which allows the vending machine owner to monitor the Sales of each item, Total Earnings and the Inventory remaining through the use of an internal Web Server.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-33 | The user shall be able to login to the IP address of the Smart Vending Machine to view a web page. |
| REQ-34 | The internal Web Server on the Smart Vending Machine shall allow the user to monitor the following,   * Totals amount of sales for each drink * Amount of drinks remaining |
| REQ-35 | A telegram message will be sent to the Vending Machine Owner if there is a burglary attempt. |

## Non-Functional Requirements

### Vending Machine Management

The Smart Vending Machine has 3 states as defined in the State Machine diagram in Figure 2 below. The transitions between Sales Mode, Service Mode, Alarm Mode are triggered by the events labelled “EnterSalesMode”, “EnterServiceMode”, “EnterAlarmMode”

Conditions for triggering the events are defined in the requirements below.



#### Figure 2

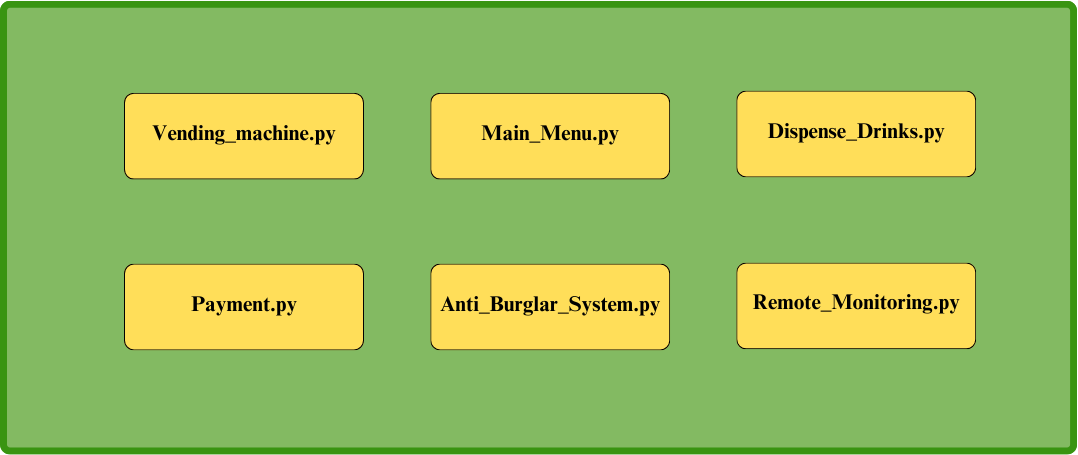
| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-36 | **“EnterSalesMode” Trigger Condition 1**   * When vending machine is not being forcefully pried open (3 axis sensor does not detect movement) |
| REQ-37 | **“EnterServiceMode” Trigger Condition 1**   * When the correct password has been keyed into keypad |
| REQ-38 | **“EnterAlarmMode” Trigger Condition 2**   * When 3 Axis Sensor detects violent movement as described in REQ-26 |
| REQ-39 | **Store Stock data to JSON file**   * Stores data into a json file which to be used to store data for stock counts of drinks and used to display. |
| REQ-40 | **Store Sales data to JSON file**   * Stores data into a json file which to be used to store sales data of the individual drinks with sales id, timestamps,price,name of drink |
| REQ-41 | **Show Profit at the sales page**   * Calculate total sales collected from the json and show it in the Sales page. |
| REQ-42 | **Show Stock count in the inventory page**   * Decrement and increment stock count and update stock when a sale or restocking is performed. |

# Software Architecture

## Static Software Architecture

The Software Architecture defines the various Software Components that are developed to realise the implementation of the system requirements.

#### Application Layer

****

**Hardware Abstraction Layer (HAL)**

****