Contents

[1. Purpose 3](#_Toc137991535)

[1.1. Intended Audience 3](#_Toc137991536)

[1.2. Intended Use 3](#_Toc137991537)

[1.3. Scope 3](#_Toc137991538)

[1.4. Definitions and Acronyms 3](#_Toc137991539)

[2. Overall System Description 4](#_Toc137991540)

[2.1. Use Case Diagrams 4](#_Toc137991541)

[2.2. System Architecture 5](#_Toc137991542)

[2.3 Functional Requirements 6](#_Toc137991543)

[2.3.1. Startup and Main Menu 6](#_Toc137991544)

[2.3.2. Selection of drinks 7](#_Toc137991545)

[2.3.3 Money counting and Payment 9](#_Toc137991546)

[2.3.4. QR Code Transaction System (online) 11](#_Toc137991547)

[2.3.5 Drink Dispensing + Display 13](#_Toc137991548)

[2.3.6 Burglar Detection System 14](#_Toc137991549)

[2.3.7. Maintenance Services 16](#_Toc137991550)

[2.4 Non-Functional Requirements 18](#_Toc137991551)

[2.4.1 Power management 18](#_Toc137991552)

[3. Software Architecture 19](#_Toc137991553)

[3.1. Static Software Architecture 19](#_Toc137991554)

[4. Additional notes 20](#_Toc137991555)

# Purpose

This SRS document describes the System Requirements and Software Design for an IoT Smart Vending Machine.

## Intended Audience

The target audience of this document shall be for System and Software Engineers working on the development, testing, and deployment of this project.

## Intended Use

The SRS defines the overall System Architecture and Requirements as well as the Software Architecture and Design. This document is 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

This document includes the system architecture of the product, use case diagrams, as well as requirements for the smart vending machine. It also includes software architecture, and requirements for perform system, integration, and unit testing.

## Definitions and Acronyms

Stock Array – Remaining Stock of Drinks  
Layout Array – Allocated slots of the drinks

# Overall System Description

## 2.1. Use Case Diagrams

## A picture containing diagram, text, line Description automatically generated

## 2.2. System Architecture

The System Architecture of the project is as stated in the diagram below.  
“requirements” shall refer to a group/section of requirements and “requirement” shall refer to a single requirement.

A picture containing text, screenshot, font, parallel

Description automatically generated

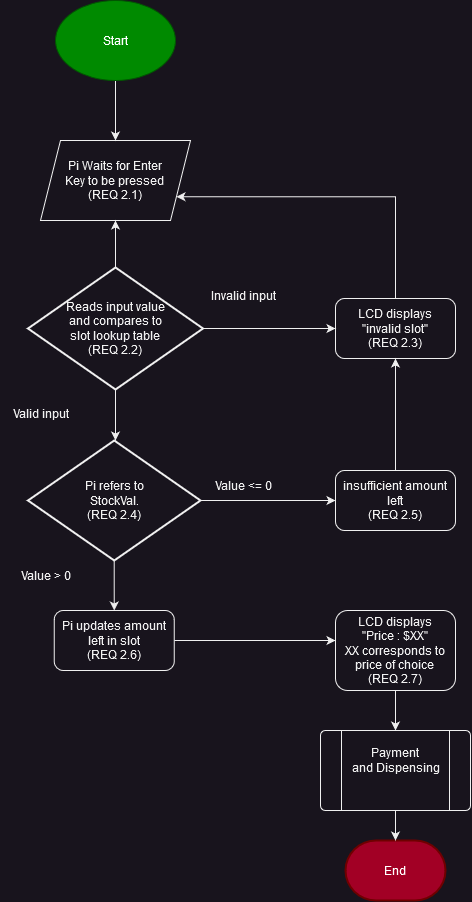
## 2.3 Functional Requirements

### 2.3.1. Startup and Main Menu

|  |  |
| --- | --- |
| REQ\_ID | Requirement |
| REQ\_1.1 | LCD displays idle message  L1 : “ Input selection code” |
|  | When Config command is inputed via ssh,  REQ\_1.2 LCD displays idle message  (Python interrupt thread via input())  L1: “ Config mode”  L2 :” awaiting SU input”  NetStat inputted  REQ\_1.3 > Show current connection with internet + (status of website)  ShowSlot inputted  REQ\_1.4 > Array showing reference codes of items in their respective slots are serial printed  ShowStock inputted  REQ\_1.5 > Array showing remaining amount of items left along side their reference code are serial printed  REQ\_1.6 > Pi returns back to idle config state (REQ\_2) upon finishing the previous command  REQ\_1.7 > Exitconf returns pi back to initial idle message ( REQ\_1.1) |
|  |  |

2.3.2. Selection of drinks  
For physical implementation:

|  |  |
| --- | --- |
| REQ 2.0 | The selection of drinks shall follow the flowchart in Figure 2.1. |

  
Figure 2.1. Flowchart for physical implementation (# button is used as a substitute for enter)  
  
For online implementation:

#

Start

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENT** |
| 2.1o | Website fetches and displays StockVal. of each different drinks (updated whenever user opens the website) |
| 2.2o | User selects their drink and website moves to payment screen |
| 2.3o | After successful payment, a QR code is generated and a png is available for the user to download/preview ( this is to be scanned later at the vending machine)  This is discussed in section 2.3.4. |
|  |  |

### 2.3.3 Money counting and Payment

The vending machine shall provide options for payment and counting/validation of money deposited for purchase of a drink. This shall be made available physically at the machine, or remotely via an online website/app.

Payment shall be made after the steps in requirement 2, where the user has selected their preferred drink.

The detail of each implementation is shown below:

#### 2.3.3.1. Physical implementation

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| 3.1 | The payment interface shall be made available to the user after selection of their drink, along with activation of the sensors (RFID) |
| 3.2 | The user shall be shown the amount to be paid before payment |
| 3.3 | The user shall be able to pay via credit/debit card which shall be read via the RFID card reader |
| 3.4 | The user shall be presented with an option to retry and an option to cancel if payment is unsuccessful |
| 3.5 | The flow of payment is to be executed as per Figure 3. |
| 3.6 | The user shall be able to interrupt the process at any point, and cancel the transaction. |

A picture containing text, diagram, screenshot, design

Description automatically generated

Figure 3.1. Flowchart for payment (physical)

#### 2.3.3.2. Online implementation

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| 3.1o | The payment interface shall be made available to the user after selection of their drink on the website/app. |
| 3.2o | The user shall be shown the amount to be paid before payment |
| 3.7 | The user shall be able to pay via credit/debit card by entering the card details |
| 3.4o | The user shall be presented with an option to retry and an option to cancel if payment is unsuccessful |
| 3.8 | If payment is successful, the drink shall be reserved for 24 hours; after which it will be released to existing stock if not collected. |
| 3.9 | If payment is successful, the user shall be redirected to QR generation. |
| 3.10 | The flow of payment is to be executed as per Figure 4. |
| 3.6o | The user shall be able to interrupt the process at any point, and cancel the transaction. |

A diagram of a payment process

Description automatically generated with low confidence

Figure 3.2. Flowchart for payment (online)

### 2.3.4. QR Code Transaction System (online)

#### 2.3.4.1. QR Generation (remote counterpart)

The online/remote counterpart shall include a QR generation capability after the payment of the drink, for the collection at the vending machine.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| 4.1 | The QR code shall be able to convey the purchase information and proof of payment to the vending machine. |
| 4.2 | The QR code shall be for one time use; any subsequent use of the QR code shall be invalid. |
| 4.3 | The program shall follow the flowchart in Figure 4.1 |

A picture containing text, diagram, screenshot, design

Description automatically generated

Figure 4.1. The flowchart for QR generation

#### 2.3.4.2. QR Detection (vending machine counterpart)

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| 4.4 | The machine shall present an option for the user to scan QR codes. |
| 4.5 | The machine shall redirect the user to collection if the payment is verified. |
| 4.6 | The machine shall be able to scan for QR codes with its camera, retrieving it's value |
| 4.7 | The machine shall not redirect to collection if payment is not successful. |
| 4.8 | The machine shall follow the flowchart in Figure 4.2 |
| 4.9 | The QR Detection code shall run in the background and will be always on as long as the machine is in Normal Operation mode (also known as High power mode) |

A picture containing text, diagram, line, origami

Description automatically generated

Figure 4.2. The flowchart for QR detection

### 2.3.5 Drink Dispensing + Display

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ\_5.1 | When a drink option is selected (a drink slot number is entered) and payment is completed, the dispensing begins and fetches the corresponding drink name to the slot number entered, the program will have an array containing the corresponding drink and Refcode information, example:  drink\_inventory = [  {"refcode": "0001", "drink\_name": "Coca-Cola", "slots": [1, 2, 3], "stock": [5, 8, 3], "price": 1.50},  {"refcode": "0002", "drink\_name": "Sprite", "slots": [4, 5, 6], "stock": [5, 8, 3], "price": 1.25},  {"refcode": "0003", "drink\_name": "Fanta", "slots": [7, 8, 9], "stock": [5, 8, 3], "price": 1.30},  {"refcode": "0004", "drink\_name": "Pepsi", "slots": [10, 11, 12], "stock": [5, 8, 3], "price": 1.20},  # Add more drinks and their corresponding slot numbers, stock, and price information here  ]  The smart vending machine will have a total of 70 slots and offers 12 different drink options. Each element within the drink\_inventory array includes the drink's reference code, name, the slots where the drink is available, and the current stock level. This organized structure allows for efficient retrieval and management of drink information during the dispensing process. |
| REQ\_5.2 | The display then shows the message "Currently dispensing [Drink Name - #Refcode]". |
| REQ\_5.3 | The gyro will then turn 90 degrees to simulate the dispensing of the selected drink and return to its original position (0 degrees) after a few seconds. Gyro refers to the servo motor powering the door. |
| REQ\_5.4 | Once the dispensing is complete, the display will show the message “Dispensing completed, enjoy your drink!”. |

A diagram of a flowchart

Description automatically generated with low confidence

### 2.3.6 Burglar Detection System

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ\_6.1 | The smart vending machine shall be equipped with a potentiometer on the hinge of the door to detect any forceful opening of the vending machine door, ensuring enhanced security measures. |
| REQ\_6.2 | The vending machine will constantly monitor the potentiometer sensor readings in real time to detect changes in the machine's position and detect any forceful movements. |
| REQ\_6.3 | To determine a break-in event, the potentiometer readings will be compared against a predetermined threshold value specifically set to identify forceful break-in attempts.  This threshold value will be carefully calibrated based on the potentiometer 's sensitivity and the expected force required to open the vending machine, ensuring accurate and reliable detection. |
| REQ\_6.4 | Upon surpassing the threshold value, indicating a break-in, the smart vending machine's burglar alarm will be immediately activated.  The burglar alarm will emit a loud and attention-grabbing buzzer sound, designed to deter potential theft or vandalism and alert nearby individuals of the unauthorized access attempt. The camera on the vending machine will also take a picture of the burglar.  The buzzer shall be activated based on the timing diagram below: |
| REQ\_6.5 | An alert will be sent out to the engineer’s smartphone through SMS or other means about the break-in. |
| REQ\_6.6 | The buzzer will continue sounding continuously until the engineer’s access code is entered into numeric keypad on the vending machine, causing the buzzer alarm to stop. |

A picture containing text, diagram, plan, screenshot

Description automatically generated

### 2.3.7. Maintenance Services

In order to be able to access the internals of a vending machine, the user must enter a specific code, such in a “key” to be able to access the system of the vending machine.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ\_7.1 | The following will be displayed to the user as the LCD screen is powered on.  Line 1: “Input selection code”  Line 2: Displays time for when the key is entered “hh:mm:ss” |
| REQ\_7.2 | As the user keys in the code, the system will detect if the code is input and registered into the internal database system, the alarm/burglar system will turn off and allow the user access to the internals by displaying the main menu in REQ-1.  \*The code must be a specific key for the authentication to works, not the code to access the slots but the internals of the vending machine. |
| REQ\_7.3 | If the system detects an invalid code, the following shall be implemented.  -LCD shall display the following:  Line 1:“Invalid code”  -Buzzer shall be activated for a duration of time to alert the user of invalid authentication, following the timing diagram below: |

## 2.4 Non-Functional Requirements

### 2.4.1 Power management

The vending machine has two Power Modes as defined in the State Machine diagram in Figure xx below. The transitions between the Low Power Mode and High Power Mode are triggered by the events labeled “evEnterLPM” and “evEnterHPM”.

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

A diagram of a power saving mode

Description automatically generated with medium confidence

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ\_8.1 | **“evEnterLPM” Trigger Condition 1**  The default state of the vending machine is Power Saving Mode and when the button is pressed, it turns on Normal Operation Mode. |
| REQ\_8.2 | **“evEnterLPM” Trigger Condition 2**  When the keypad for slot selection is not pressed for 10 min, Low Power Mode is turned on. |
| REQ\_8.3 | **“evEnterLPM” Trigger Condition 3**  When the items inside the slots are being restocked |
| REQ\_8.4 | **“evEnterHPM” Trigger Condition 1**  When the button for slot selection is pressed |
| REQ\_8.5 | The LED shall be turned off in Power Saving Mode and turned on in Normal Operation Mode. |

2.4.2 Extra features

|  |  |
| --- | --- |
| REQ\_9.1 | The burglar detection system shall be triggered when the accelerometer senses heavy shaking. |

# 3. Software Architecture

## 3.1. Static Software Architecture

**Application Layer (Physical)**

**inventory.py**

**anti\_theft.py**

**dispensing.py**

**qr.py**

**payment.py**

**selection.py**

**power\_mgt.py**

**main\_menu.py**

**Application layer (online)**

qr\_gen.py

main\_selection.py

inventory.py

payment.py

**Hardware Abstraction Layer**

hal\_keypad.py

hal\_rfid\_reader.py

qr\_reader.py

hal\_camera.py

hal\_accelerometer.py

hal\_servo.py

hal\_buzzer.py

hal\_led.py

hal\_lcd.py

hal\_adc.py

# 4. Additional notes



The vending machine shall look like this. There will be multiple slots for the same drink, and the user shall press the relevant slot for the drink physically. Since the labels are visible, the small LCD will focus on displaying price instead of printing all drink names which will not be feasible.

In the online implementation, the slots will be automatically managed by the system, and a slot with stock will be used.