Table of Contents

[Document Version 2](#_Toc170141326)

[1. Purpose 3](#_Toc170141327)

[1.1. Intended Audience 3](#_Toc170141328)

[1.2. Intended Use 3](#_Toc170141329)

[1.3. Scope 3](#_Toc170141330)

[1.4. Definitions and Acronyms 3](#_Toc170141331)

[2. Overall System Description 4](#_Toc170141332)

[2.1. Use Case Diagrams 4](#_Toc170141333)

[2.2. System Architecture 5](#_Toc170141334)

[2.3. Functional Requirements 6](#_Toc170141335)

[2.3.1. Function 1 – Start Up and Main Menu 6](#_Toc170141336)

[2.3.2. Function 2 – On-site Purchase 7](#_Toc170141337)

[2.3.3. Function 3 – Online Purchase 8](#_Toc170141338)

[2.3.4. Function 4 – Authentication Services 9](#_Toc170141339)

[2.3.5. Function 5 – Burglar Alarm 10](#_Toc170141340)

[2.4. Non-Functional Requirements 11](#_Toc170141341)

[2.4.1. Non-Functional Requirement – Activation Management 11](#_Toc170141342)

[3. Software Architecture 12](#_Toc170141343)

[3.1. Static Software Architecture 12](#_Toc170141344)

# Document Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Update | Name | Date | Version |
| 1. | Outline + Requirements | SRS\_project\_xxx | 1 June 2024 | 1.0 |
| 2. | Flowcharts | SRS\_project\_xxx | 3 June 2024 | 2.0 |
| 3. | Edited Requirements | SRS\_project\_xxx | 19 June 2024 | 3.0 |
| 4. | Final touch up | SRS\_project\_01\_SmartDrinkVM | 22 June 2024 | 4.0 |
| 5 | Updated SRS | SRS\_project\_01\_SmartDrinkVM | 9 August 2024 | 5.0 |

# Purpose

## Intended Audience

This SRS document describes the System Requirements and Software Design for an IoT Smart Drink 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

## Definitions and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| LCD | Liquid-Crystal Display |
| RFID | Radio-Frequency Identification |
| VM | Vending Machine |
| IR | Infra-Red |
| USONIC | Ultrasonic |

# Overall System Description

## Use Case Diagrams

A diagram of a customer service

Description automatically generated

## System Architecture

The System Architecture

A diagram of a computer hardware development board

Description automatically generated

## Functional Requirements

### Function 1 – Start Up and Main Menu

The Smart Drink VM will prompt users on their needs when they use the VM.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-01 | When the VM is active, the file to be called is based on the slide switch position.  Left position: Projectmain.py file is called (for purchasing of drinks) Right position: Authentication.py file is called (for service technicians and drinks suppliers) |
| REQ-02 | In the main menu defined in REQ-01, if the slide switch is in the left position, the Projectmain.py file will run, and the following menu shall be displayed on the LCD Screen      Line 1 : “1. Key Selection”      Line 2 : “2. Online Payment” |
| REQ-03 | In the main menu defined in REQ-01, if the slide switch is in the right position, the Authentication.py file will run, and the following menu shall be displayed on the LCD Screen  Line 1 : “Authentication”  (wait for a while)  Line 1 : “Key user code:”  Line 2 : The LCD shall display “X” for each key pressed  The valid user code is made up of 6 characters, including numbers & special characters. (\*73524) |

### Function 2 – On-site Purchase

The Smart Drink VM allows users to purchase their drink physically from the VM.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-04 | Each drink in the VM is given a corresponding number based on our CSV file for drinks (drinks.csv).  From REQ-02, if the user selects “1. Key Selection”, the VM will read the drink.csv provided by the vendor with the list of drinks and their corresponding price the vending machine is selling. |
| REQ-05 | The VM will prompt the user to select a drink afterwards. The user is allowed to enter any number between 0 to 99. When a number is keyed in, the following text shall be displayed on the LCD Screen.      Line 1 : “Enter Number:”  Line 2 : *[Display number entered]* :  “X” or “XX”, where X represent a digit from 0 to 9 |
| REQ-06 | If the user is unsatisfied with his selection, he can choose ‘\*’ to re-enter his drink number as and when he wants.  If the user is satisfied with his selection, he can choose ‘#’ to confirm his number. Afterwards, the flowchart defined in Figure 1 shall be implemented.  [“\*” still works when the user has accidentally press “#”] |

A diagram of a flowchart

Description automatically generated

### Function 3 – Online Purchase

The Smart Drink VM supports “Online Purchase” remotely via smartphones or an external website.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-12 | The user shall be able to access and view the website to purchase drinks from the VM without being physically there. |
| REQ-13 | The user should make payment via the app or website, and a QR code or barcode will then be generated for the user to scan at the VM, to collect their drink. |
| REQ-14 | From REQ-02, if the user selects “2. Online Payment”, then the flowchart defined in Figure 2 shall be implemented. |

A diagram of a flowchart

Description automatically generated

### Function 4 – Authentication Services

For service technicians and drink suppliers, they will need to enter a valid user code on the keypad to open the VM door without triggering the burglar alarm.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-19 | Service technicians and/or drink suppliers have only 3 attempts to key in the valid code before the buzzer rings. For each invalid attempt, the following text shall be displayed on the LCD screen.      Line 1 : “Invalid code”  Line 2 : *[Display attempts left] :*  “*(integer)* attempts left” |
| REQ-20 | If the 6 character code corresponds to the valid user code, then the flowchart defined in Figure 3 shall be implemented. |

A diagram of a machine

Description automatically generated

### Function 5 – Burglar Alarm

To avoid any theft of the drinks, a buzzer will be activated if the VM door has been forcefully pried open.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-24 | From REQ-19, if an invalid code has been keyed in wrongly for the 3rd time, the buzzer shall be activated based on the timing diagram in Figure 4 |

A diagram of a graph

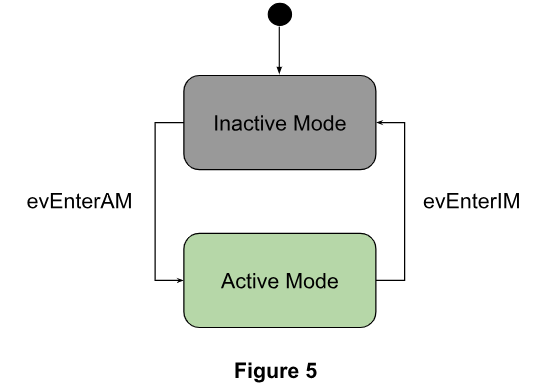
Description automatically generated

## Non-Functional Requirements

### Non-Functional Requirement – Activation Management

The Smart Drink VM has 2 modes as defined in the State Machine Diagram in Figure 4 below. The transitions between the Inactive Mode and Active Mode are triggered by the events labelled “evEnterIM” and “evEnterAM”.

Conditions for trigger events are defined in the requirements below.



|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-25 | **“evEnterIM” Trigger Condition 1**  When the keypad has not been pressed for at least 1 minute. |
| REQ-26 | **“evEnterIM” Trigger Condition 2**  When the PiCam has not detected any payment for at least 1 minute. |

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-27 | **“evEnterAM” Trigger Condition 1**  When the user presses any button on the keypad. |

# Software Architecture

## Static Software Architecture

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

A screenshot of a computer application

Description automatically generated