Table of Contents

Document Version

No	Update	Name	Date	Version
1.	Initial version	Joel	11 June 2024	1.0
2.	Flow chart	Jaslyn	11 Jun2 2024	1.1
3.	Non-Functional Requirements	Xin Jun	11 June 2024	1.2
4.	Purpose	Jaslyn	11 June 2024	1.3
5.	Overall System Description	Joel	13 June 2024	1.4
6.	Purpose: scope	Jaslyn	15 June 2024	1.5
7.	Functional Requirements	Joel	17 June 2024	1.6
8.	Software Architecture	Jia Yi	17 June 2024	1.7
9.	Table of contents update	Joel	20 June 2024	1.8
10.	Definition and acronym	Jaslyn	23 June 2024	1.9
11.	Final update for table of contents	Joel	23 June 2024	2.0

Purpose

1.1. Intended Audience

This SRS document describes an IoT Library System Requirements and Software Design for a library system. The target audience is system and software engineers working on developing this project.

1.2. Intended Use

The SRS defines the overall System Architecture and Requirements and 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.

1.3. Scope

This project is an online book reservation system. Users can conveniently reserve desired books from the library's collection through the library's website and choose their desired location for book collection. The system is integrated with existing security measures like verifying user identity via NRIC/SP card upon book retrieval. Furthermore, it automatically tracks loan periods and associated late fees, ensuring efficient library operations and user accountability. This project also helps to reduce the burden on Librarians by automating the process of lending and collecting books. Librarians will then be able to focus more on cataloguing new books on the website.

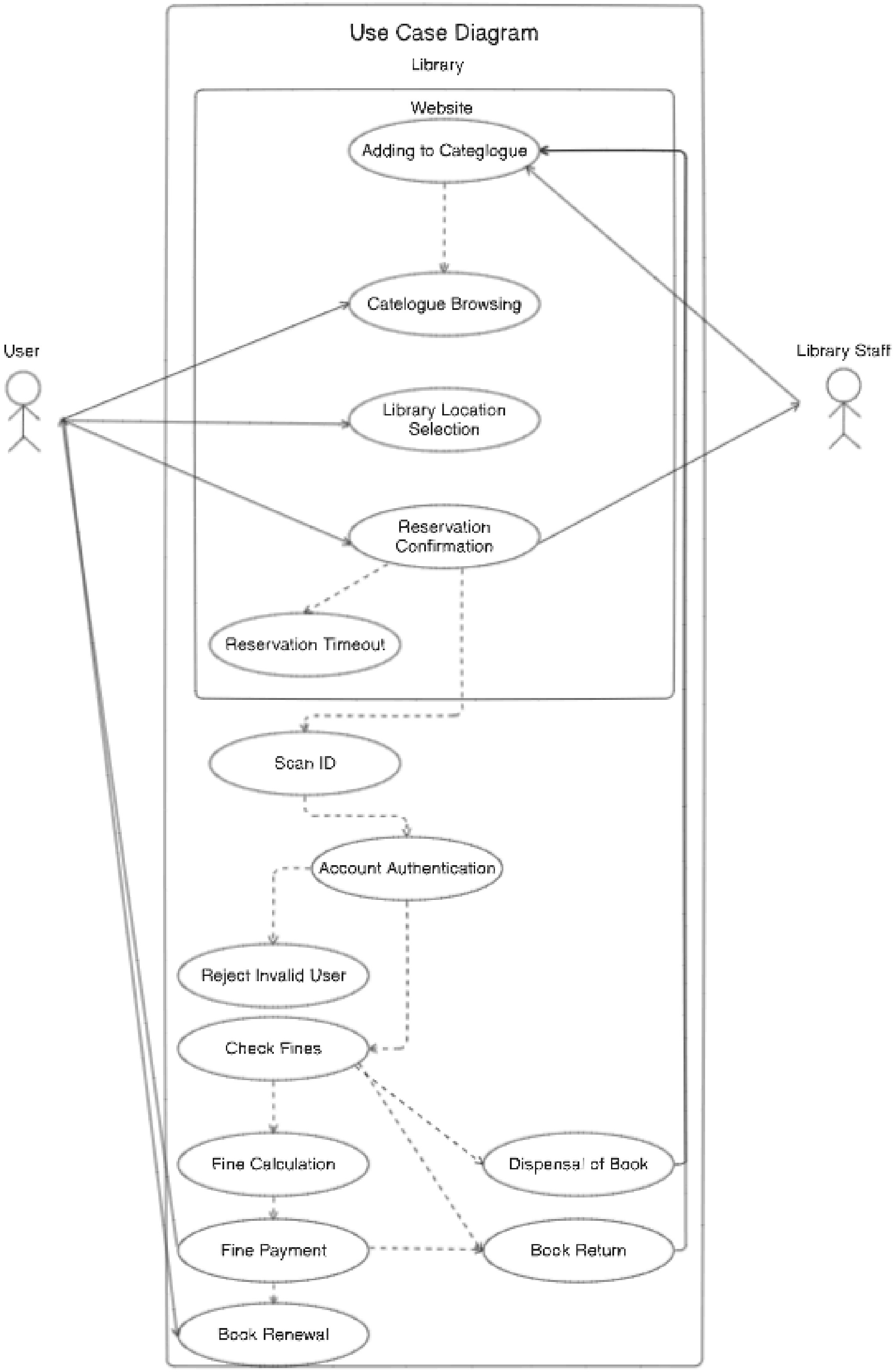
1.4. Definitions and Acronyms

Acronym	**Description**
IR	Infra Red
LED	Light Emitting Diode
NFC	Near Field Communication
SW	Software
HW	Hardware
RFID	Radio-Frequency Identification
NRIC	National Registration Identity Card
SP	Singapore Polytechnic

2. Overall System Description

2.1. Use Case Diagrams

Use Case Diagram

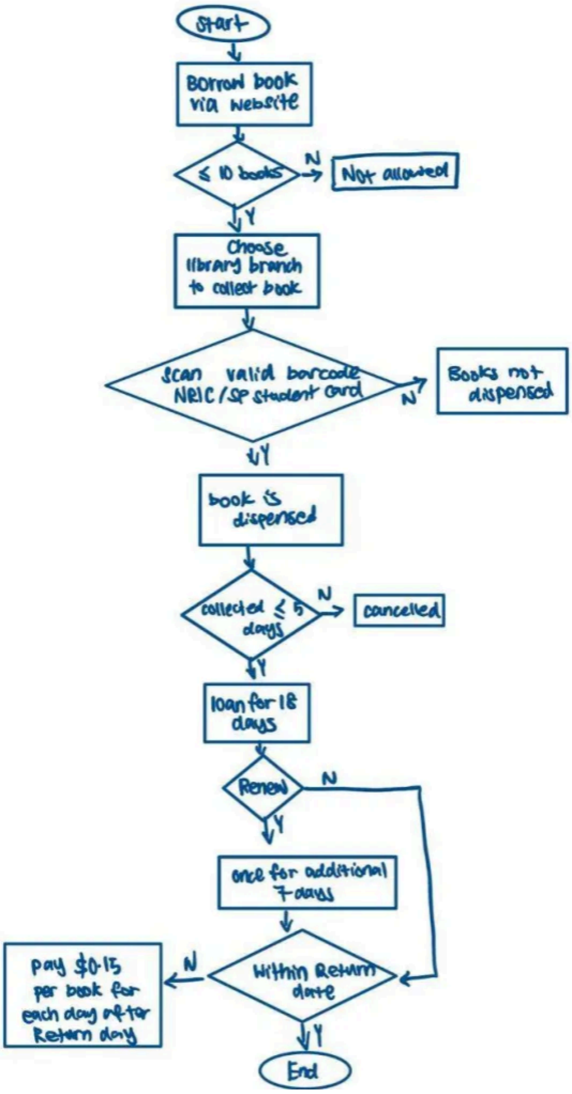


The use case diagram above shows how the Raspberry Pi implementations at physical libraries are supposed to work together with the online website.

The website works to allow users to browse the catalogue of the library online and gets updated every time books are returned or borrowed from the library or the Raspberry Pi implementation at the physical location. If users are interested in any of the books they see in the library catalogue, the user can determine a convenient location of a library where they can reserve a book to collect from the raspberry pi system. If the reservation fails for any reason on the website, they will get timed out of the system.

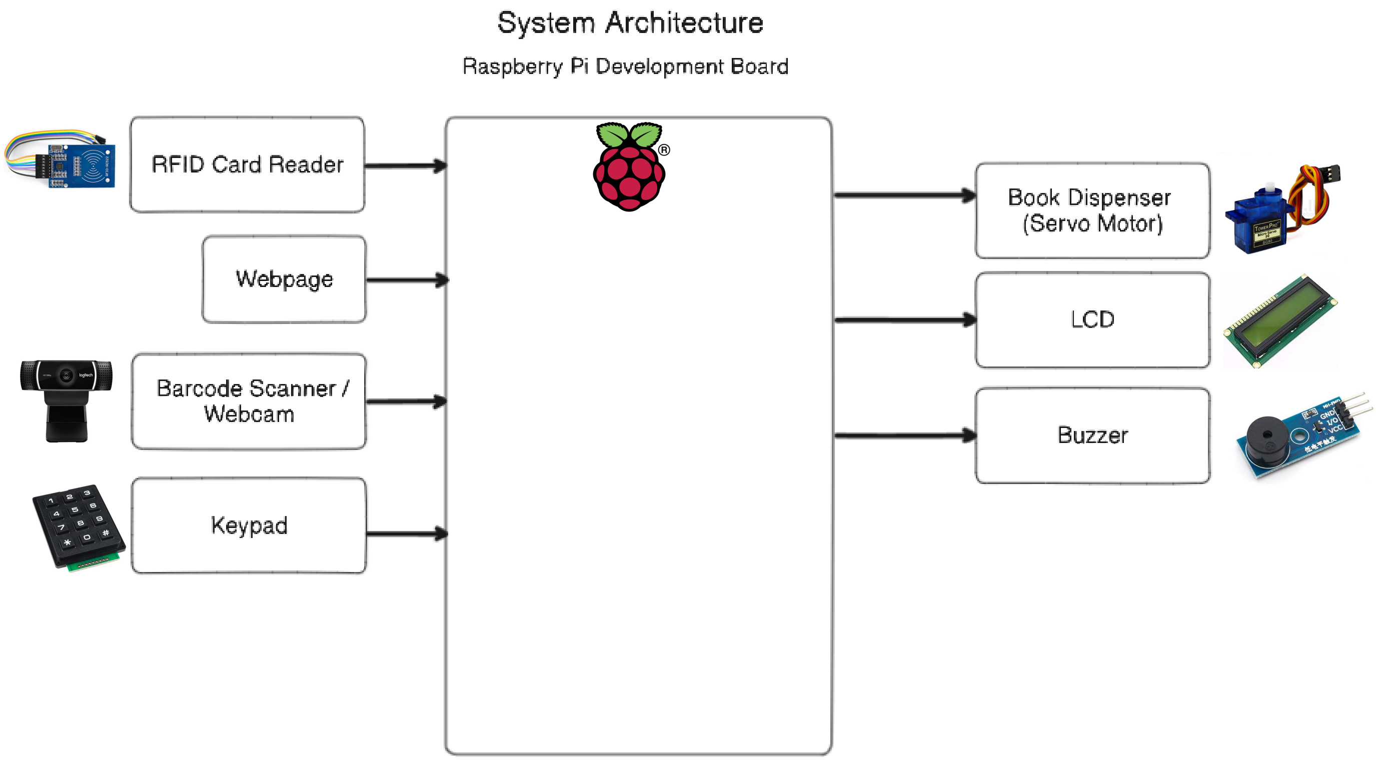
When users visit the physical location, they have to scan their ID which first allows for a second level of security to authenticate the account and the reservation. If the IDs do not match up, they will be rejected and unable to retrieve the book. If their account is authenticated, the system will first check if the account has any outstanding fines which it will deduct should there be any found. After that, the Raspberry Pi implementation will dispense the book to the user. This will then update the website catalogue. This procedure is the same as when the user returns borrowed books or renews the duration they are borrowing a book.

Flowchart



2.2. System Architecture

System Architecture



The Raspberry Pi Development Board uses the RFID Card Reader to scan the user’s ID card that is required for the verification and use of the user’s library account. This is to ensure the security of the system when books are loaned out and returned.

The webpage allows the board to connect to the catalogue available and also connect with the online platform for users who reserve books for collection.

The barcode scanner works in tandem with the RFID Card Reader should the user choose to scan the barcode on their IDs instead of scanning it with the RFID Card Reader.

This works with the Raspberry Pi’s Servo Motor that dispenses the book from the system.

2.3. Functional Requirements

2.3.1. Book Reservation

Online book reservation system via website to make a reservation for the book in which they want to borrow from the library.

REQ_ID	**Requirement**
REQ-01	Database for library catalogue available.
REQ-02	Website for users to interact with to select books for reservation and location for collection
REQ-03	System to send data to Raspberry Pi Board for reservation and Raspberry Pi to send information back to the website database.
REQ-04	System limits total number of books borrowed to 10

2.3.2. Account Verification

Account must be verified before books can be returned or collected in person after reservation online

REQ_ID	**Requirement**
REQ-05	Barcode scanner to scan NRIC or SP Student Card.

2.3.3. Fine System

Make users pay any outstanding fines.

REQ_ID	**Requirement**
REQ-06	RFID scanner for “Library Card”
REQ-07	Fine calculation system that charges $0.15 everyday after the book’s return date
REQ-08	Renewal system that allows users to renew books in advance should they need more time. Max renewal for 7 days after being loaned out for 18 days
REQ-09	Force users to pay outstanding fines via their NRIC or SP Student Card to their accounts before being able to return or borrow books.

2.3.4. Book Borrowing and Dispensal

System allows users to borrow books if all the conditions are met. This also allows the system to recycle the books into circulation if for whatever reason the books are not picked up in the timeframe.

REQ_ID	Requirement
REQ-10	System cancels the reservation if the books are not collected within 5 days.
REQ-11	Dispensing book after account verification function succeeds and the user is confirmed to be verified.
REQ-12	Force users to pay outstanding fines via their”Library Card” to their accounts before being able to return or borrow books.

2.4. Non-Functional Requirements

Non-Functional Requirement are those that refer to requirements that are not functionally, vital, such as reliability and/or stress testing

2.4.1. Renewal Limit

Ensuring that no one permanently renews their borrowed books forever.

REQ_ID	Requirement
REQ-13	If the user enters a renewal value of >7, the book would not be renewed
REQ-14	If the barcode is invalid, the LCD will display an error and ask the user to please try again.
REQ-15	When a book is being dispensed, the system will not process any new request until the current action has been completed.
REQ-16	The buzzer beeps when the barcode is successfully scanned to ensure the user knows the scanning of the barcode was a success.

2.4.2. Book Dispensal

Ensuring that the book dispensal system doesn’t fail when more stress is put onto it.

REQ_ID	Requirement
REQ-14	When a book is being dispensed, the system will not process any new request until the current action has been completed.

2.4.3. Barcode Scanning

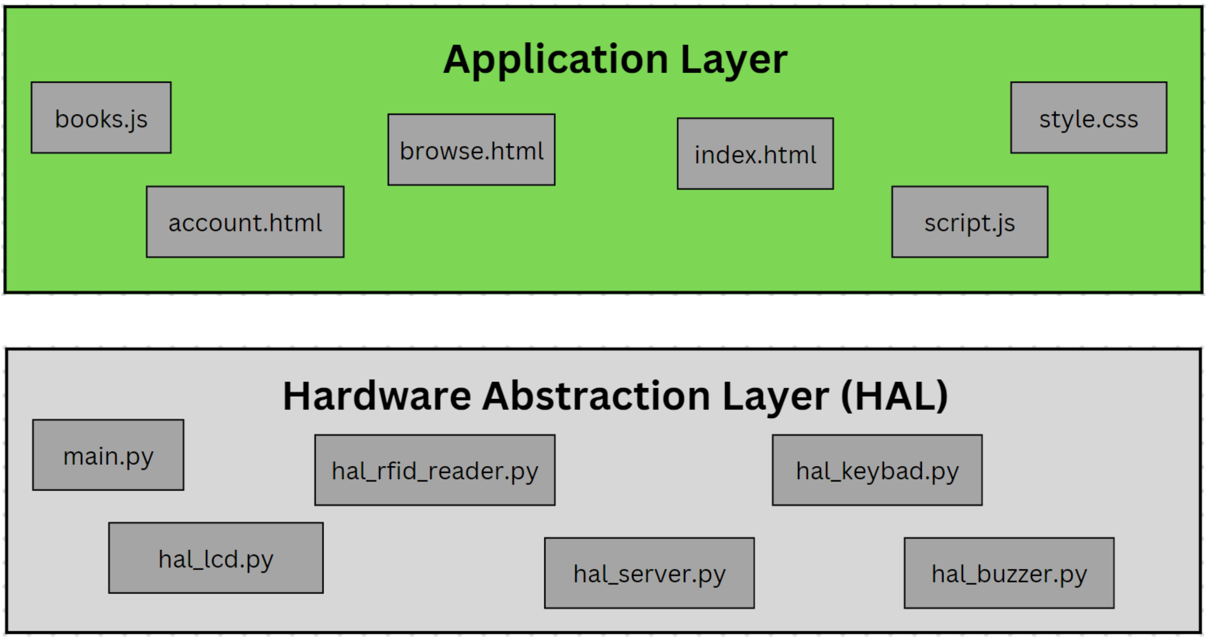
Ensures that the barcode scanning system can operate as smoothly as possible with the user understanding easily if the scanning has been successful or not.

REQ_ID	Requirement
REQ-15	If the barcode is invalid, the LCD will display an error and ask the user to please try again.
REQ-16	The buzzer beeps when the barcode is successfully scanned to ensure the user knows the scanning of the barcode was a success.

3. Software Architecture

3.1. Static Software Architecture

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


Application Layer
books.js
browse.html
index.html
style.css
account.html
script.js
Hardware Abstraction Layer (HAL)
main.py
hal_rfid_reader.py
hal_keybad.py
hal_lcd.py
hal_server.py
hal_buzzer.py