CS383 Group Project

Semester #431

Software Requirements Specification (SRS)

For Robot Librarian

Version 1.1

Prepared by Group 4

Table of content

Change history	
version	**Changes**
0.1	Provisional version
1.0	Initial document release
1.1	Added this change history and fixed typos

1 Introduction

This document was written by Abdullah Alhabib 371111359 for the robot librarian project of the course Software engineering CS383 by the request of dr.Faisal Alhwikem.

1.1 Scope

This is a software requirement specification document for the robot librarian project.

The project is a requirement for the software engineering course at the college of computer in AlQassim university. It aims to create a software interface between a robot and a library, the specifics of the robot and the library are out of the scope, instead we're focusing on the interface between them, and the application of a robot librarian.

1.2 Document structure

This document goes over the overview on the design of the robot librarian project, focusing more on requirement engineering rather than software structure specifics, for the latter, the viewer of this document can go to the Software Design Document (SDD).

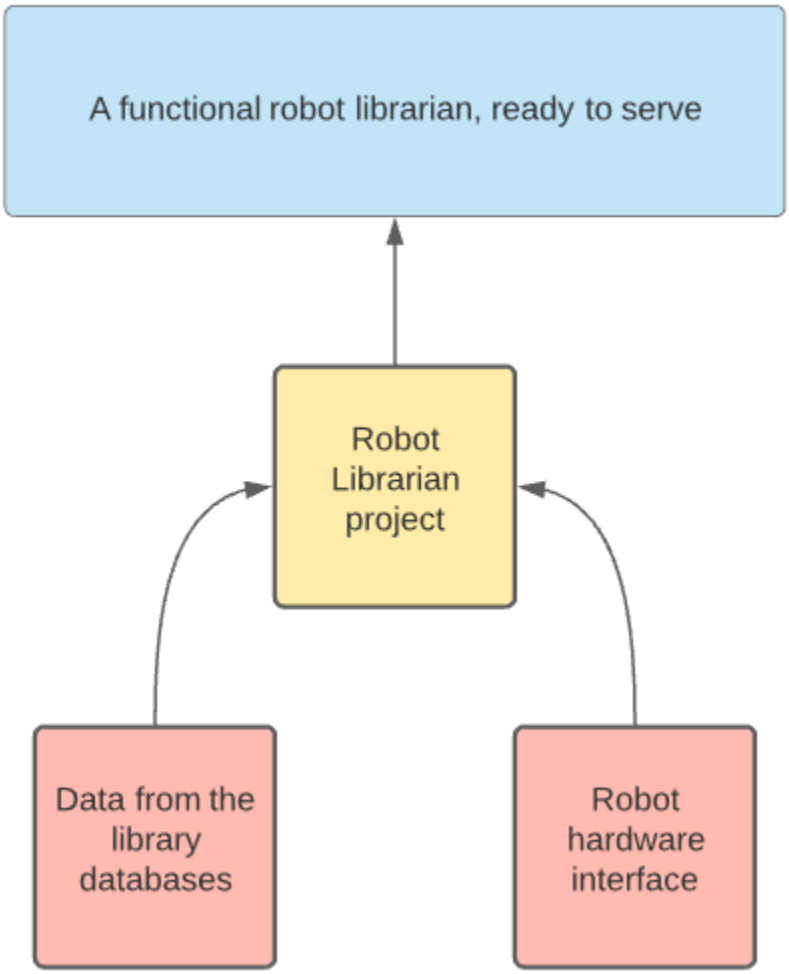
In this document we will walk through the requirements and use cases of this project, as well as the database and external interface dependencies.

2 Overview description

2.1 Product perspective

The software is an interface that allows a robot to do some librarian tasks, such as the use cases illustrated down in section 4 of this document.

It needs an established robot interface and library databases, both of which are specified in section 2.5 of this document.



2.2 Product functions

This software will allow a robot to do the following functionalities in the library:

- Lend books.
- Put returned books back into the correct shelf.

- Check if a book is available.
- Bring books for local reading, and return them to the proper place.

2.3 User classes and characteristics

Users are divided into three classes according to this software:

1. User: the average customer using the library, limited access (find, borrow and return books) and should not require any technical knowledge more than what the average person has.
2. Admin: Library staff, has more access (direct commands to bots, access to some customer information, add but not delete information from databases) and can do minor harm to the system if misused, Still does not require high technical knowledge, though might need some training on how to properly use the software's administrative abilities.
3. SuperAdmin: Assigned the IT staff, this class can assign admins and has complete access, if misused or maliciously used it can do major damage to the databases, should only be assigned to very few trusted staff members, preferably technologically apt.

2.4 Design and implementation constraints

- We are creating a general interface, we can not cater to a specific robot or a specific library structure, this software may or may not need tweaking for specific applications.
- Our knowledge in robotics is limited, since we are focusing on computer science and information technology and not computer engineering.

2.5 Assumptions and dependencies

For this product to work properly, we assume that:

- There must be a database for books, using barcodes generated using ISBN as a unique ID, and containing title, genre, version and production year.
- The library must physically have barcodes assigned on shelves, according to a shelf database, where the barcode is the unique ID and each shelf belongs to a subgenre, which belongs to a genre.

- There must be designated extra temporary shelves, with name "temporary" as the genre and "extra" as sub-genre in the shelf database, these shelves are used in case a shelf belonging to a certain genre/subgenre is full, the robot will instead put these books in the temporary shelves.

3 User Requirements

3.1 User requirement definitions

3.1.1 Robots shall organize returned or new books according to genre then name using Barcode.

3.1.2 The customer can request to borrow a book through automated means, and be able to know the books available, and find books through title, author(s), or genre.

3.1.3 books must be dusted regularly.

3.2 User requirement specification

3.2.1 There must be a database for books, using barcodes generated using ISBN as unique ID, and contain title, genre, version and production year.

3.2.2 There must be barcodes for genres and sub-genres, in order to organize books through genre, then sub-genre, then shelf number.

3.2.3 There must be designated extra temporary shelves, with name "temporary" as the genre and "extra" as sub-genre.

3.2.4 If the extra shelves have a book, the librarians must be notified, so as to either expand the needed genre/sub-genre or cull it.

3.2.5 The robot will have to organize books according to the title alphabet after finding the genre.

3.2.6 In case the book must be placed in between two books that already are placed,
the robot will have to safely shift the books that come after the held book one spot until meeting a free spot, then put the held book in the correct position.

3.2.7 In case the shelves designated for the genre are full, the book will then be placed in the extra temporary shelf and send notification of doing so to librarians notification center.

3.2.8 Using the robot functionality, put the book on the shelf after finding the right spot.

3.2.9 There must be a user interface for the customer.

3.2.10 The tasks the customer can request using the user interface are:

a: Borrow book: which will ask for the customer to log in, if the customer did not register or have forgotten his/her details, he/she will be redirected towards the help desk with human employees.

b: Return book: which will open a tray and request the customer to put the book in there, then automatically process the return using ISBN (use ISBN to find customer account and process "return success").

c: Find if book is available: Customer can use title, author, genre or ISBN to find a list of books with specified filters, information on each book includes: title, author, version, genre and year.

d: Request book for local reading: which will run the same process of finding books, then after that the bot will lead a customer to a free reading table, then retrieve the book and give it to the customer.

3.2.11 Every day, the bots will use their dusting functionality at the beginning of work hours, cleaning books and shelves of dust.

3.2.12 The process goes like this:

a: At the start of work hours, X amount of bots are assigned cleaning duty.

b: The number of bots will be assigned by the librarians and should be easily changeable.

c: The software will divide the number of shelves (taken from the database) by the number of cleaning bots, then be assigned the result as shelves to clean.

d: The software will then assign each bot to the shelves starting from the shelf with the number (amount of shelves already assigned)

and ending with the shelf with the number (amount of shelves already assigned + the result of the division shelves/bots), this new stopping point is the starting point assigned to the next bot.

3.3 User Requirements

3.3.1 IO functions

IO (Input-Output) functions that are the interface between the robot and the GUI and remote GUI.

R0001: Find book

Rationale: It finds the book based on search criteria (one or more of: title, author, genre or ISBN)

Source:User input matched with book database.

Status: implemented.

R0002: Borrow book

Rationale: when the book is found, the customer will be able to request to borrow.

Source: Output from R0001 Find book, book location in book database.

Status: implemented.

R0003:Read locally

Rationale: When the book is found, the customer will be able to request to read locally.

Source: Output from R0001 Find book, book location in book database.

Status: implemented.

R0004: Return book

Rationale: Customer wishes to return borrowed book

Source: Book tray matched with user database.

Status: implemented.

R0005: Register new customer

Rationale: New customer registration.

Source: User input.

Status: implemented.

3.3.2 Robot functions

R0006: Move

Rationale: Invoke hardware to move to a specified shelf /table.

Source: Robot hardware and shelf/table location from library database

Status: implemented.

R0007: Add/remove book to/from storage

Rationale: The robot grabs a book and adds it to storage, or takes it out of storage

Source: Hardware.

Status: implemented.

R0008: Organize shelf

Rationale: Robot will move books in specific manner

Source: DB.

Status: implemented.

R0009: Put book on shelf

Rationale: Remove book from storage and put it on the shelf

Source: Book DB, shelf DB.

Status: implemented.

R0010: Clean

Rationale: Robot will clean a book

Source: Book db, shelf DB.

Status: implemented.

R0011: Register location

Rationale: Registers current location (in order to return to it in case of cleaning interrupt or

Source: Table barcode OR shelf barcode.

Status: implemented.

R0012: Change lane

Rationale: Avoid potential robot collision by moving to a different lane if lane is occupied.

Source: Robot database.

Status: implemented.

3.3.3 Library functions

R0013: Add book

Rationale: Add book to subgenre

Source: admin input

Status: implemented.

R0014: Remove book

Rationale: Tag a book that it is completely removed from the library

Source: Admin input, library DB

Status: implemented.

R0015: Add genre

Rationale: Add a genre to the library

Source: Admin input

Status: implemented.

R0016: Add subgenre

Rationale: Add a subgenre to a genre

Source: Admin input

Status: implemented.

R0017: Edit book info

Rationale: Allow for a book info to be edited

Source: Admin input, library DB

Status: implemented.

R0018: Edit genre

Rationale: Allow for a genre to be edited

Source: Admin input

Status: implemented.

R0019: Edit subgenre

Rationale: Allow for a subgenre to be edited

Source: admin input

Status: implemented.

R0020: remove genre

Rationale: Mark a genre as completely removed from the library

Source: superAdmin input, library DB

Status: implemented.

R0021: remove subgenre

Rationale: Mark a subgenre as completely removed from the library

Source: superAdmin input, library DB

Status: implemented.

3.3.4 User management

R0022: Promote user

Rationale: Increase the level of the user from customer to admin.

Source: superAdmin input.

Status: implemented.

R0023: Demote user

Rationale: Decrease the level of the user from admin to customer.

Source: superAdmin input.

Status: implemented.

R0024: Delete customer

Rationale: Delete a customer and all info associated.

Source: admin/superAdmin input.

Status: implemented.

R0025: Edit user info

Rationale: Edit the info of a user of the same level or lower

Source: admin/superAdmin input.

Status: implemented.

R0026: Add user

Rationale: Add a user through the user management module, similar to R0005.

Source: admin/superAdmin input.

Status: implemented.

4 Use cases

UC0001 Find book:

Description:

Find a specified book for the customer to be borrowed or read locally.

Basic Flow:

A: When find book option is selected from the GUI's main menu, ask for the customer to type in **any** of the following:

- Book title
- Book author
- Book genre
- Book ISBN

B: Return any book that matches the search criteria, along with all of its information (title, author, version, number of copies available, genre, ISBN and year).

C: Provide the following options: Borrow book (UC0002), Request for local reading (UC0004) and Return to main menu.

UC0002 Borrow book:

Description:

After finding the desired book, the customer is able to request to borrow the book

Basic Flow:

A: From UC0001, the customer clicks on the option to borrow the book

B: customer is asked to login or register:

- In case of login, the customer is asked to either enter the login information or swipe the library card for faster automatic login.
- In case of registration request, then the use case moves on to UC0003.

C: After recognizing the customer information, the robot will move to the shelf that has the book (using R0006), and add the desired book to its storage (R0007), then move back to the customer (R0006), and bring the book out so the customer can take it (R0007).

UC0003 customer registration:

Description:

Register customer information.

Basic Flow:

A: After the customer clicks on registration, take them to the registration menu and ask for the following information:

- Name
- Phone
- Email
- Address
- Username and password for future login

B: upload that information to the database using R0026.

UC0004 Request a book for local reading:

Description:

After selecting a book the customer is able to request for it to be brought for local reading, no login or registration required for this.

Basic Flow:

A: After finding book in UC0001, the customer clicks on read locally

B: The robot will move to the shelf that has the book (using R0006), and add the desired book to its storage (R0007), then move back to the customer (R0006), and bring the book out so the customer can take it (R0007).
C: after finishing the customer puts the book on a specified tray on their table.

UC0005 Return borrowed book:

Description:

Customer returns their borrowed book.

Basic Flow:

A: Customer selects the return borrowed book option from the GUI.

B: A tray will open so the customer can put the book/books.

C: The robot will automatically scan the book's information to finalize its return procedure (tag the book returned, tag the specific customer as no longer borrowing and have returned the book/s, and proceed to return the book to its place).

UC0006 organize books:

Description:

The robot organizes the books once new books arrive according to their info.

Basic Flow:

A: The new book must already have a barcode sticker with its database information.

B: The robot will put the book on the right shelf (according to genre/subgenre) and then calculate the position of the held book (using the shelf database) according to the alphabet by default (which can be changed to ISBN).

C: In case the book can not fit in the shelf due to it being full, the robot will put it on the temporary shelf

D: Otherwise the Robot will shift the positions of the books to put the new book according to what was calculated in step B.

UC0007 Clean books

Description:

The robot cleans the books when it's idle , or outside of the library's work hours. It must have more than one hour of battery life.

Basic Flow:

A: The robot is idle and has more than one hour of battery life remaining (can be changed).

B: The robot checks its registry for "last cleaned location", which should be done by the function in R0011, in case it doesn't have any registry, it will start cleaning from the first book in the library databases.

C: The robot cleans using its functionality (see R0010), then after every time it cleans a book the robot will register the book's location (R0011).

D: Any commands will interrupt the cleaning and the robot will immediately go to process the command.

UC0008 Robot direct control:

Description:

Admins can give the robot direct commands.

Basic Flow:

A: The admin selects the option from the admin GUI.

B: The admin then can override the robot's automatic functionality momentarily, issuing any of the robot's functions manually, or change the settings of one (like the battery life limit for cleaning).

5 External Interface Requirements

Robot hardware interface:

Robot hardware interface must support the functionalities provided in the robot functions requirements in section 3.3.2 .

6 Nonfunctional requirements

R0001 Usability

Description:

The project should provide Text-To-Speech, easily changeable font sizes and GUI should be clear enough for any technical background.

Rationale:

Library visitors come from many backgrounds and are from a large range of ages, so accessibility options should be available.

R0002 scalability

Description:

The project must support small and large uses, from a singular robot to many.

Rationale:

The usual expected use of the project is gradual increase in reliance on robot librarians, starting from single digit robots (possibly singular) and expanding to more as the need increases.

R0003 manageability

Description:

While not directly requested by the user, the project supports user and library management functionality (sections 3.3.3 and 3.3.4).

Rationale:

To make the project scalable, we need to also make it complete so the user would not need more programs that run things this project closely use anyway, since the project has access (and needs) user, library and shelf databases, then supporting management for those makes it comprehensive and removes the need of external database management software.

R0004 Security

Description:

Make the program secure by limiting what users can do without being admins, and to make sure this happens we have separated the customer GUI class from the admin GUI class, discussed in further detail in the SDD document.

Rationale:

While there isn't a large incentive to maliciously attack the robot librarian, since there isn't any monetary value to the information stored and the robots cannot be ordered to move OUT of the library and be stolen that way, we still must still make it harder to do so to prevent any petty cybercrime (like phishing customers by using their info) or attempts at causing chaos.