

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
FALL 2019**



**NEPSTER
POCKET SHELF**

**UTSAB ACHARYA
SANDESH NAGA
SUDIP KANDEL
TARA GURUNG
BHASKAR ACHARYA**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	9.20.2019	SK, UA, SN, BA, TG	document creation
0.2	9.20.2019	BA, UA, SN, SK, TG	complete draft
0.3	9.21.2019	SN, UA, BA, SK, TG	release candidate 1
1.0	9.22.2019	TG, UA, SN, SK, BA	official release
1.1	9.23.2019	BA, UA, SN, SK, TG	added design review requests
1.2	9.24.2019	BA, UA, SN, SK, TG	complete draft

CONTENTS

1	Introduction	5
2	System Overview	5
3	UI	6
3.1	Layer Hardware	6
3.2	Layer Operating System	6
3.3	Layer Software Dependencies	6
3.4	Subsystem 1	6
4	Business Logic	8
4.1	Layer Hardware	8
4.2	Layer Operating System	8
4.3	Layer Software Dependencies	8
4.4	Business Controller	8
4.5	AR Generator	9
4.6	QR Generator	9
4.7	UI Input Validation	10
5	Database	11
5.1	DB Controller	11
5.2	Login Mgt	12
5.3	Register Mgt	14
5.4	Shelf mgt	15
5.5	Item mgt	16
5.6	Search mgt	18
6	Appendix A	19

LIST OF FIGURES

1	System architecture	5
2	Example subsystem description diagram	6
3	Example subsystem description diagram	8
4	Example subsystem description diagram	9
5	Example subsystem description diagram	10
6	Example subsystem description diagram	10
7	DB Controller description diagram	11
8	Login mgt description diagram	13
9	Register mgt description diagram	14
10	Shelf mgt description diagram	16
11	Item mgt description diagram	17
12	Search mgt description diagram	19

LIST OF TABLES

2	DB Controller Subsystem Interfaces	12
3	Login Mgt Subsystem Interfaces	14
4	Register Mgt Subsystem Interfaces	15
5	Shelf mgt Subsystem Interfaces	16
6	Item mgt Subsystem Interfaces	18
7	Search mgt Subsystem Interfaces	18

1 INTRODUCTION

Inventory management is an important part of running business and managing items around us. Traditional practice of managing items for business cost a lot of time and work force for companies. Customers become frustrated when the items they're looking for aren't available on the shelf. Bad practice of managing items in our home, school, works etc. cost lots of time because it is hard for us to remember what items are needed and where it is located. Pocket-self is the inventory tracking ISO/Android based mobile application which allow users to keep the items in available space, which is created by the user. We called PocketShelf because the idea behind it, we can carry the inventory application in our pocket and can search items in just a click. The function of the app is user can sign up and login then add items or add shelf or search the items. App can help to find the place where to keep the items in shelf, keeps record of it and help to find the items when customers search for it. The app can help the customers to find and add the items in the shelf with the help of the items pictures too. We used Augmented reality as our key feature of the application. Where user can see the item before they approach to grab it. Which might be helpful in the sense that it will save the user effort and time spent and they know that they are getting the right thing. Time frame to notify user can be predefined by the user at the time of its entry. Our app use the online database to save every entry by the user and had to be encrypted before it goes to the database, all the data had to flow through the server. So, user must be online to be able to use the app. We had PIN protection as an extra feature on the app, where if the user wants not to show the item location to another user using the same app on same device. Which we found cool feature if the user wants to secure valuable items somewhere in the shelf. The scope of pocketShelf are prioritize inventory, Track all product information, Track sales, sorts in-venture according to manufacture and best by date of the items. App helps to prioritize inventory on the basic of the customers demand, shape and sizes of items in the shelf. For example, heavy and big items are preferred in he button of the shelf and small or light items in the top the shelf. App will keep the records of the inventory as well as items in stocks. Another important feature of the app is track sales. Update the numbers of the items when add or remove an item from the shelf. App can sort the inventory according to customer preferences based on attributes of the item. The app not only keep track of the item but able to sort the items based on the expiration dates and can notify the user if the item is expiring soon.

2 SYSTEM OVERVIEW

This is the first sub section that user interacts with. When the user opens the mobile application for the first time details for SignUp is displayed. Then the user enters all the information required for SignUp including all the valid user name and password to create a account. Then an account for the user is created.

Figure 1: System architecture

3 UI

UI is the graphic feature of the application mainly responsible for the displaying information and interact with the user. UI consist of login subsystem responsible for taking userid and password from user. Registration subsystem is responsible to take userid, email, password, full name from the user. Add_Shelf subsystem is responsible for taking the shelf description from the user. Add_Item is responsible for taking item description from user. Search subsystem is responsible for taking user input via QR code or input text. After any kind input and options chosen by the user UI_Controller will send all user input and the option chosen by the user to Business_Controller in Business_Logic. UI_Controller also accept the input messages from the Business_Controller, stating successful completion of the chosen task by the user.

3.1 LAYER HARDWARE

Being an android/iOS app, UI will use components underlying hardware for user input and to show output to the user. i.e. user uses phone screen for input and show output in same screen.

3.2 LAYER OPERATING SYSTEM

A description of any operating systems required by the layer.

3.3 LAYER SOFTWARE DEPENDENCIES

A description of any software dependencies (libraries, frameworks, etc) required by the layer.

3.4 SUBSYSTEM 1

Descibe at a high level the purpose and basic design of this subsystem. Is it a piece of hardware, a class, a web service, or something else? Note that each of the subsystem items below are meant to be specific to that subsystem and not a repeat of anything discussed above for the overall layer.

Figure 2: Example subsystem description diagram

3.4.1 SUBSYSTEM HARDWARE

A description of any involved hardware components for the subsystem.

3.4.2 SUBSYSTEM OPERATING SYSTEM

A description of any operating systems required by the subsystem.

3.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

3.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

A description of any programming languages used by the subsystem.

3.4.5 SUBSYSTEM DATA STRUCTURES

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

3.4.6 SUBSYSTEM DATA PROCESSING

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

4 BUSINESS LOGIC

This is another important layer in our design. This part of the system works as a bridge between the User Interface and the Database of the system. If the data is being received from the UI controller that are coming form different subsections of UI, the Business controller will, depending upon what kind of input it gets, either generate an AR, generate a Qr or does an input validation. The Business Logic will send the data to the data base for more validation and verification or to store the data in database. If Business Logic is getting the instruction to fetch data form Database, then it will contact Database to get particular data, which is then passed on to the UI.

4.1 LAYER HARDWARE

A description of any involved hardware components for the layer. For example, if each subsystem is a software process running on an embedded computer, discuss the specifics of that device here. Do not list a hardware component that only exists at the subsystem level (include it in the following sections).

4.2 LAYER OPERATING SYSTEM

iOS/Android

4.3 LAYER SOFTWARE DEPENDENCIES

This layer is build using ReactNative. Dependencies we needed for this part are "dependencies":

```
"expo": "34.0.1",  
"expo-barcode-scanner": "6.0.0",  
"expo-permissions": "6.0.0",  
"firebase": "6.6.0",  
"native-base": "2.13.7",  
"react": "16.8.3",  
"react-dom": "16.8.6",  
"react-native": "https://github.com/expo/react-native/archive/sdk-34.0.0.tar.gz",  
"react-native-datepicker": "1.7.2",  
"react-native-gesture-handler": "1.4.1",  
"react-native-search-bar": "3.4.3",  
"react-native-simple-radio-button": "2.7.3",  
"react-native-vector-icons": "6.6.0",  
"react-native-web": "0.11.4",  
"react-navigation": "4.0.0",  
"react-navigation-stack": "1.5.1",  
"reinput": "3.7.1"]
```

4.4 BUSINESS CONTROLLER

Any kind of data coming and going to Business Logic do so through the Business controller. The data coming to Business Logic will be divided into three different ways, namely AR generation, QR generation or Input Validation related. Depending upon those different input types, Business controller communicates with the Database controller to get particular data or to store that information.

Figure 3: Example subsystem description diagram

4.4.1 SUBSYSTEM HARDWARE

NA

4.4.2 SUBSYSTEM OPERATING SYSTEM

iOS/Andriod

4.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

4.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

JavaScript

4.4.5 SUBSYSTEM DATA STRUCTURES

JSON, Array, Union, List.

4.4.6 SUBSYSTEM DATA PROCESSING

Shorting Algorithm was used.

4.5 AR GENERATOR

AR generator is the part of Business logic that generates AR when a customer wants to store an item to the shelf. The generated AR is then used when the customer wants to search the item. When the customer wants to search an item, he will scan bar-code or enter manually. Then the program will display the AR right on the bar-code of the item.

Figure 4: Example subsystem description diagram

4.5.1 SUBSYSTEM HARDWARE

Phone Camera

4.5.2 SUBSYSTEM OPERATING SYSTEM

iOS/Andriod

4.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Phone camera, images available

4.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

JavaScript

4.5.5 SUBSYSTEM DATA STRUCTURES

JSON, Array, Union, List.

4.5.6 SUBSYSTEM DATA PROCESSING

Shorting Algorithm was used.

4.6 QR GENERATOR

Based on description of the item customer provides, QR generator generate QR. This QR is then store in the database along with the description of the item.

Figure 5: Example subsystem description diagram

4.6.1 SUBSYSTEM HARDWARE

Phone Camera

4.6.2 SUBSYSTEM OPERATING SYSTEM

iOS/Andriod

4.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Phone camera

4.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

JavaScript

4.6.5 SUBSYSTEM DATA STRUCTURES

JSON, Array, Union, List.

4.7 UI INPUT VALIDATION

This section will validate format and types of input from user. It will reject the inputs if user's input is wrong format and suggest the correct format of the input. Provide guidance on how to fix any errors, don't just tell users what they did wrong. It will prevent from unauthorized SQL injection into the data base.

Figure 6: Example subsystem description diagram

4.7.1 SUBSYSTEM HARDWARE

N/A

4.7.2 SUBSYSTEM OPERATING SYSTEM

iOS/Andriod

4.7.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Phone camera

4.7.4 SUBSYSTEM PROGRAMMING LANGUAGES

JavaScript

4.7.5 SUBSYSTEM DATA STRUCTURES

JSON, Array, Union, List.

5 DATABASE

This subsection is used when we have to save or retrieve data from the database. When user tries to login, register, add self, add item or search we have to use database. When somebody wants to register for the app, he provides all the information then these information is taken by the DB controller and saved in organized manner. same process occurs for all the other activities like adding self, adding item or login.

5.1 DB CONTROLLER

This is a type of main controller of the Database system. All the data that needs to be stored in database is first handled by this layer and later stored in the database. When any of the other layer send data to the database layer, first database layer takes the information and figures out what type of data is provided and what to do with the given data.

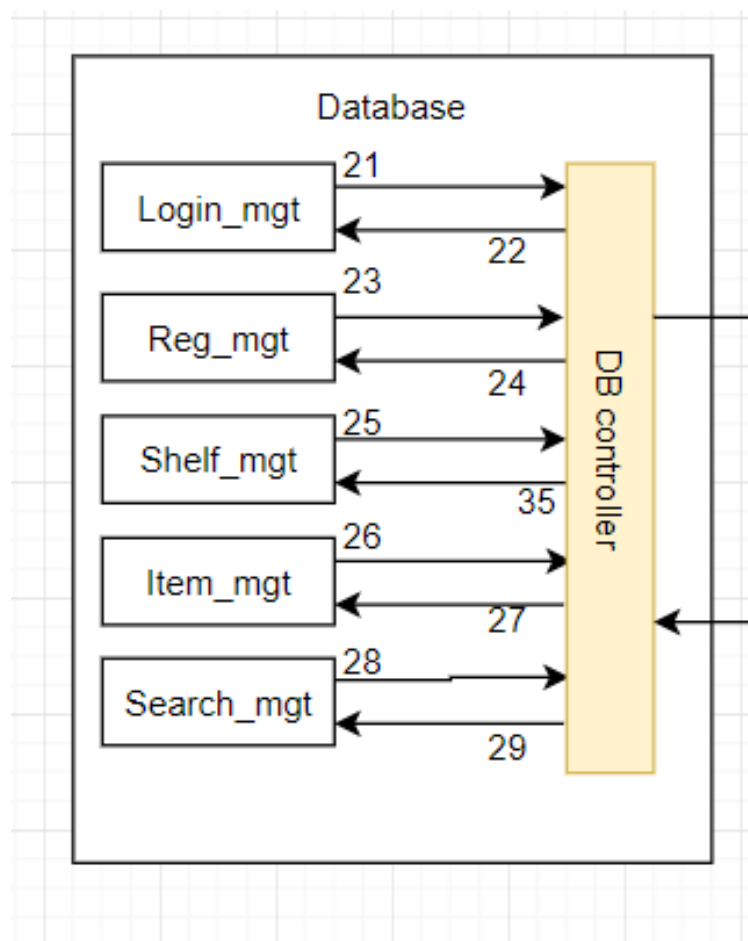


Figure 7: DB Controller description diagram

5.1.1 ASSUMPTIONS

These are the following assumptions made about this subsection:

- All the input provided to be saved in the database is a valid input.
- Device is connected to the internet when saving or retrieving the data.

5.1.2 RESPONSIBILITIES

These are the following responsibilities of the Database Controller.

- It must be able to take input sent from all the other layers.
- When input is captured it must be able to classify if the data is login data or sign up data. Must be able to classify if the incoming data is trying to search the item or save the item in the database.

5.1.3 DB CONTROLLER SUBSYSTEM INTERFACES

ID	Description	Inputs	Outputs
#22	DB controller/Login_mgt	N/A	Username Password
#21	Login_mgt/DB Controller	Success Message Failure Message	N/A
#24	DB Controllere/Reg_mgt	N/A	Username Password email address User Name
#23	Reg_mgt/DB Controller	Success Message Failure Message1	N/A
#35	DB Controller/Shelf_mgt	N/A	Shelf Name Row column Shelf Location
#25	Shelf_mgt/DB Controller	Success Message Failure Message	N/A
#26	DB Controller/Item_mgt	N/A	Item name BarCode(opt) Quantity Expiration Date(opt) Expiration_notifi- cation(opt) PIN(opt) Brewery(if alco- hol) Style(if alcohol)
#27	DB Controller/Item_mgt	Image file	N/A
#28	DB Controller/Search_mgt	N/A	User input label
#29	Search_mgt/DB Controller	Success Message Failure Message Item image	N/A

Table 2: DB Controller Subsystem Interfaces

5.2 LOGIN MGT

This subsection of database just deals with the login information. User first registers for the account in the mobile application. Then when he wants to use the app he will provide the user name and

password for the app. Then the login management layer handles all the data. It checks if the user name and password provided by the user exists in database or not. It should allow user to login if the combination of user name and password exists else it should deny user from using the application itself.

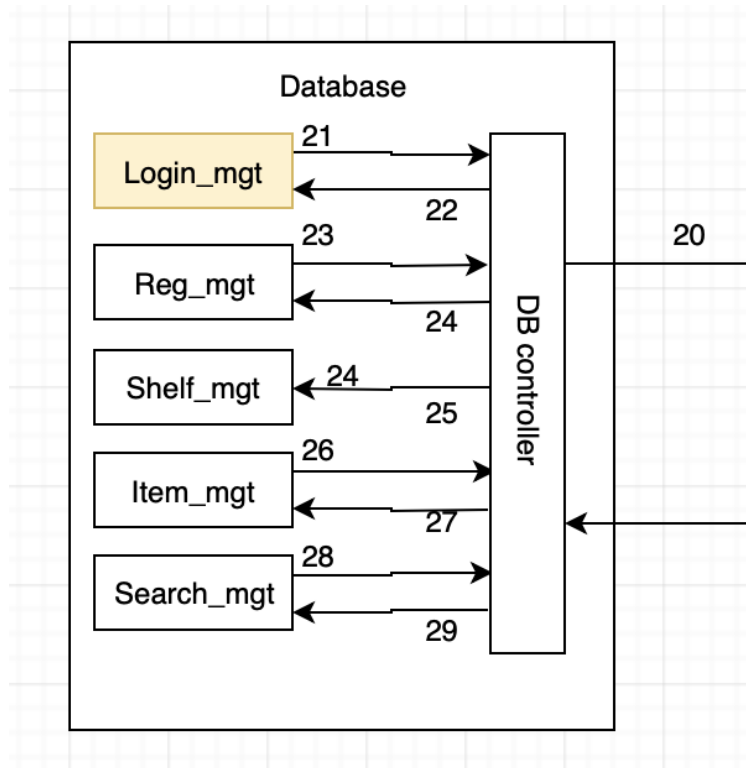


Figure 8: Login mgt description diagram

5.2.1 ASSUMPTIONS

These are the following assumptions made about this subsection:

- Input provided by the user is valid input. Inputs are not malicious code or SQL query.

5.2.2 RESPONSIBILITIES

These are the following responsibilities of this layer;

- Check the provided login against the database login information.
- If provided login data exists in database let user log in to the app else reject from logging in.

5.2.3 LOGIN MGT SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

ID	Description	Inputs	Outputs
#22	DB controller/Login_mgt	N/A	Username Password
#21	Login_mgt/DB Controller	Success Message Failure Message	N/A

Table 3: Login Mgt Subsystem Interfaces

5.3 REGISTER MGT

This subsection of the database deals with registration data like first name, last name, DOB, etc. When user provides the information it is validated for SQL injection and passed into the register management. Then the data is stored in database in organized manner by the register management subsystem.

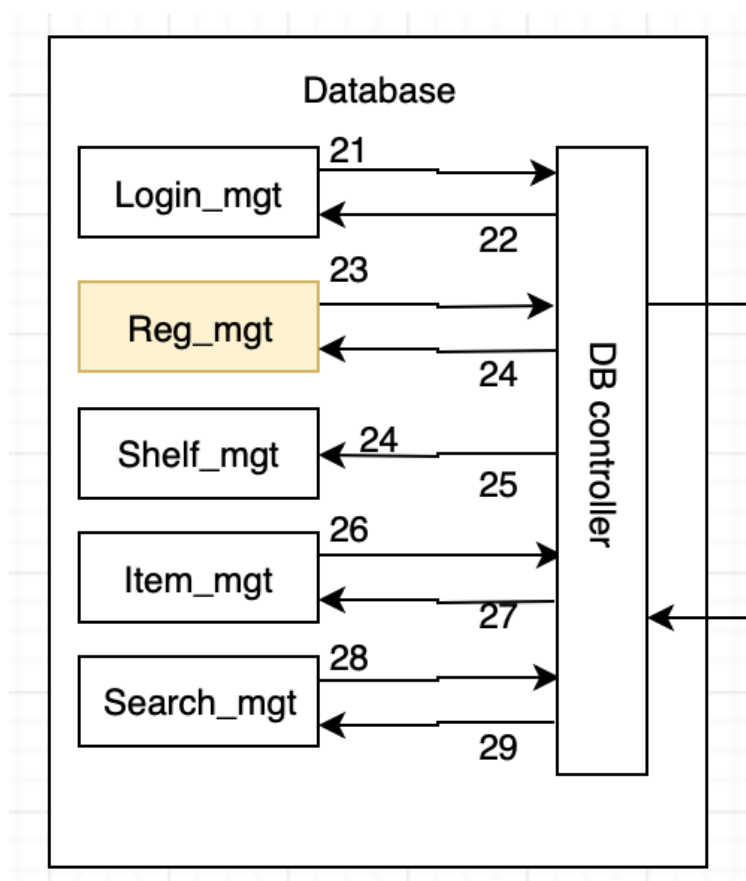


Figure 9: Register mgt description diagram

5.3.1 ASSUMPTIONS

These are the following assumptions made about this subsection:

- User provides all the information needed to register the application.
- Input provided by user is already verified to be the valid input. Eg, malicious code or SQL query is already discarded.

5.3.2 RESPONSIBILITIES

These are the following responsibilities of this subsection.

- Must be able to return error if all of input field is not provided.
- Must be able to store all the information in the database in easy retrieval fashion.

5.3.3 REGISTER MGT SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

ID	Description	Inputs	Outputs
#24	DB Controllere/Reg_mgt	N/A	Username Password email address User Name
#23	Reg_mgt/DB Controller	Success Message Failure Message1	N/A

Table 4: Register Mgt Subsystem Interfaces

5.4 SHELF MGT

This subsystem of database is used whenever user wants to add or remove self. When user provides the specification for the self then the database controller takes the data and passes it to the self management layer. Then the layer takes the information and makes the necessary adjustment in the database.

5.4.1 ASSUMPTIONS

These are the following assumptions made about this subsection:

- Input provided by user is already verified to be the valid input. Eg, malicious code or SQL query's already discarded.
- All the necessary input for adding or deleting the self is provided.

5.4.2 RESPONSIBILITIES

These are the following responsibility's that must be performed by this sub system:

- Must be able to take all the input values and allocate or de-allocate the self space according to the input values.

5.4.3 SHELF MGT SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

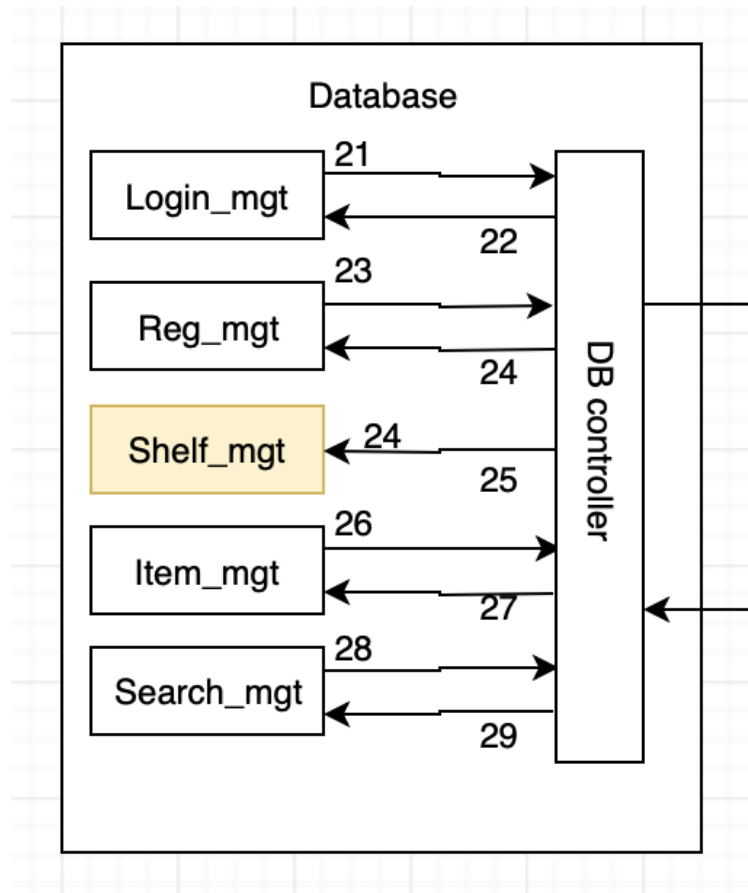


Figure 10: Shelf mgt description diagram

ID	Description	Inputs	Outputs
#35	DB Controller/Shelf_mgt	N/A	Shelf Name Row column Shelf Location
#25	Shelf_mgt/DB Controller	Success Message Failure Message	N/A

Table 5: Shelf mgt Subsystem Interfaces

5.5 ITEM MGT

This subsystem of the database is used whenever user wants to add or remove item from the database. When user provides the data for adding or removing the item, then the DB controller takes the input at first and then passes it to the item Mgmt. Then the layer takes the input value and make necessary change ie. add or remove the item accordingly.

5.5.1 ASSUMPTIONS

These are the following assumptions made about this subsection:

- Input provided by user is already verified to be the valid input. Eg, malicious code or SQL query

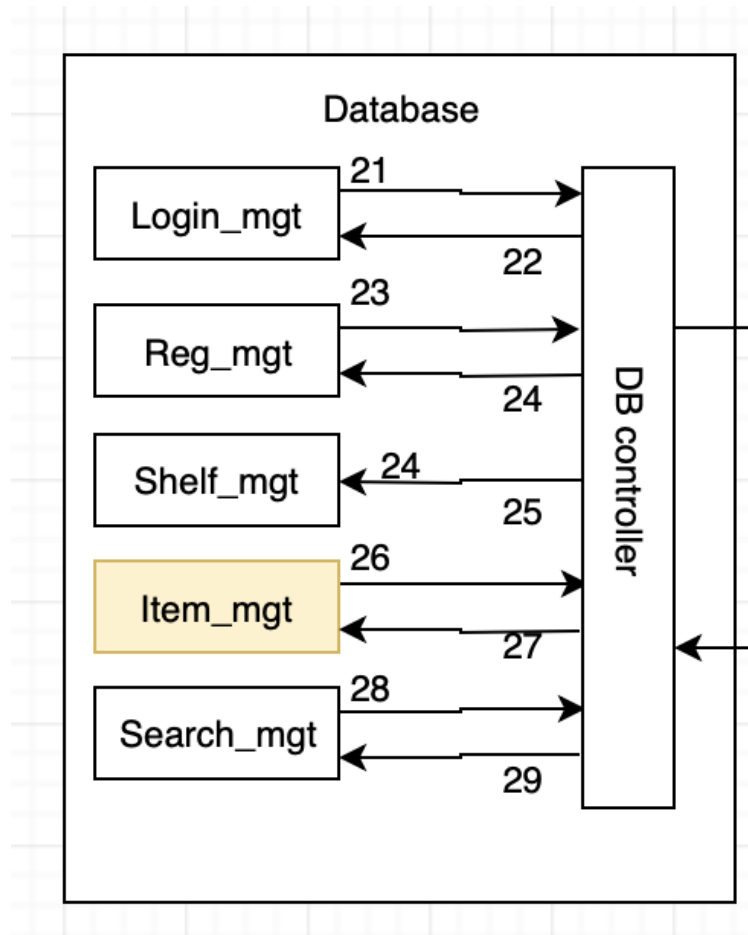


Figure 11: Item mgt description diagram

is already discarded.

- All the input value is provided by the user.

5.5.2 RESPONSIBILITIES

These are the following responsibilities of this subsystem.

- Must be able to take all the input values and add or remove item according to the input provided by the user.

5.5.3 ITEM MGT SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

ID	Description	Inputs	Outputs
#26	DB Controller/Item_mgt	N/A	Item name BarCode(opt) Quantity Expiration Date(opt) Expiration_notifi- cation(opt) PIN(opt) Brewery(if alco- hol) Style(if alcohol)
#27	DB Controller/Item_mgt	Success Message Failure Message	N/A

Table 6: Item mgt Subsystem Interfaces

5.6 SEARCH MGT

This Sub System deals with taking the item description from user and it searches for the item. Variety of options are provided for searching the item. User can use the camera in the cellphone to scan the QR generated for the specific item or users will also be able to search the item by their names.

5.6.1 ASSUMPTIONS

These are the following assumptions made about this subsection:

- Input provided by user is already verified to be the valid input. Eg, malicious code or SQL query is already discarded.
- All the necessary information is provided for the search.
- Device has working camera to scan and search.

5.6.2 RESPONSIBILITIES

These are the following responsibilities of this sub system:

- Must be able to take the input and return the result if it exists in database and return "No Items Found" if nothing is found in the database.

5.6.3 SEARCH MGT SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

ID	Description	Inputs	Outputs
#28	DB Controller/Search_mgt	N/A	User input label
#29	Search_mgt/DB Controller	Success Message Failure Message Item image	N/A

Table 7: Search mgt Subsystem Interfaces

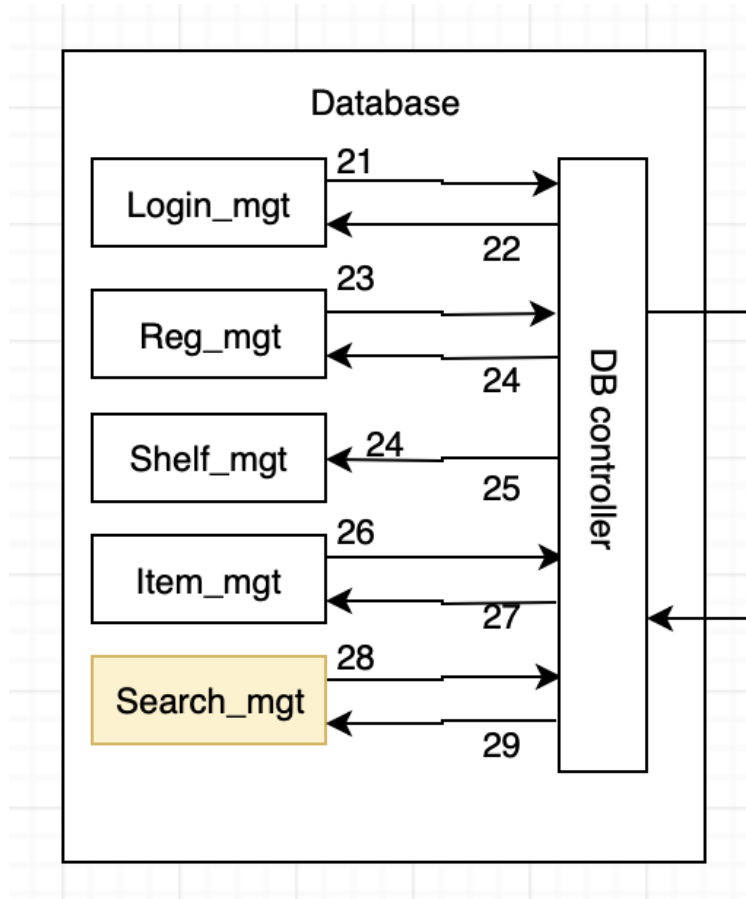


Figure 12: Search mgt description diagram

6 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES