

Nepster

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

**ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
SUMMER 2019**



POCKET SELF

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

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	08.01.2019	SK	document creation
0.2	08.02.2019	SK, UC, SN, BA, TG	complete draft
0.3	08.02.2019	SK, UC	release candidate 1
1.1	08.02.2019	TG, UC	added design review requests
1.0	08.06.2019	SK, SN, BA	official release

CONTENTS

- 1 Introduction** **5**

- 2 System Overview** **6**
 - 2.1 UI 6
 - 2.2 Business_Logic 6
 - 2.3 Database 6

- 3 Subsystem Definitions & Data Flow** **8**

- 4 User Interface (UI)** **9**
 - 4.1 UI Controller 9
 - 4.2 Login 10
 - 4.3 Register 11
 - 4.4 Add Shelf 13
 - 4.5 Add Item 14
 - 4.6 Search 15

- 5 Business Logic** **17**
 - 5.1 Business Controller 17
 - 5.2 AR Generator 18
 - 5.3 QR Generator 19
 - 5.4 UI Input Validation 21

- 6 Database** **23**
 - 6.1 DB Controller 23
 - 6.2 Login Mgt 24
 - 6.3 Register Mgt 26
 - 6.4 Shelf mgt 27
 - 6.5 Item mgt 28
 - 6.6 Search mgt 30

LIST OF FIGURES

1	A simple architectural layer diagram	6
2	A simple data flow diagram	8
3	UI Controller description diagram	9
4	Login description diagram	11
5	Register description diagram	12
6	Adding Shelf description diagram	13
7	Adding items description diagram	14
8	searching item description diagram	16
9	Business Controller Subsystem Interfaces	17
10	AR generator description diagram	19
11	QR generator description diagram	20
12	UI Input Validation description diagram	21
13	DB Controller description diagram	23
14	Login mgt description diagram	25
15	Register mgt description diagram	26
16	Shelf mgt description diagram	28
17	Item mgt description diagram	29
18	Search mgt description diagram	31

LIST OF TABLES

2	UI_Controller Subsystem Interfaces	10
3	Login Subsystem Interfaces	11
4	Register Subsystem Interfaces	12
5	Add Shelf Subsystem Interfaces	14
6	Add Item Subsystem Interfaces	15
7	Search Subsystem Interfaces	16
8	Business Controller Subsystem Interfaces	18
9	AR Generator Subsystem Interfaces	19
10	QR Generator Subsystem Interfaces	20
11	UI Input Validation Subsystem Interfaces	22
12	DB Controller Subsystem Interfaces	24
13	Login Mgt Subsystem Interfaces	26
14	Register Mgt Subsystem Interfaces	27
15	Shelf mgt Subsystem Interfaces	28
16	Item mgt Subsystem Interfaces	30
17	Search mgt Subsystem Interfaces	30

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 are 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 IOS/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 item's 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 pre-defined 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 inventory 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 the bottom 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

PocketShelf is an inventory IOS/Android application. Our application consist of many features among which adding item/shelf, search item for each user with Augmented reality experience is the key feature of our application. Diagram below show the overall high level feature of our application.

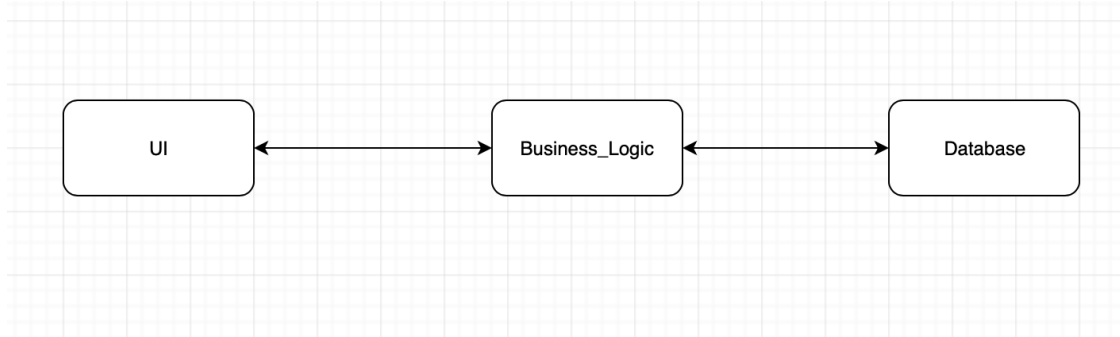


Figure 1: A simple architectural layer diagram

2.1 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.

2.2 BUSINESS_LOGIC

Business_Logic is the logical controller for the application. It is responsible for taking input from UI_Controller, work accordingly based on the input given by UI_Controller and work with DB_Controller. If the UI_Controller get any kind of input from user it validate it in order to prevent SQL injections and prevent all sort of the invalid input that can be given by the user before sending it to DB_Controller. If UI_Controller sends the item description to add to the shelf, it will create QR for the entry, append it with the description and forward it to DB_Controller to save it to Database. If UI_Controller sends any item information to search from database, Business_Logic will validate the input and the sends the item description to the DB_Controller, once item is retrieve and received by Business_Logic it fetch the image associated to the item and sends that image to UI_Controller which is shown to the user.

2.3 DATABASE

Database the is core functioning elements of our application, which will handles every thing relate to the database. DB_Controller will accepts input from the Business_Controller and work according to that. DB_Controller also return the Boolean value to the Business_Controller to indicate successful or unsuccessful completion of the queries. Along with the item needed if asked by Business_Controller. Login_mgt will checks the user input against the input in the database, if no result found it return false. Reg_mgt will add the new user to the database. Shelf_mgt will add item to the database along with the

available shelf. Once added DB_Controller will return true with the shelf description to the Business_Controller. Search_mgt will take input from Business_Controller, then it will look up in the database, if found it will return the item the Business_Controller.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

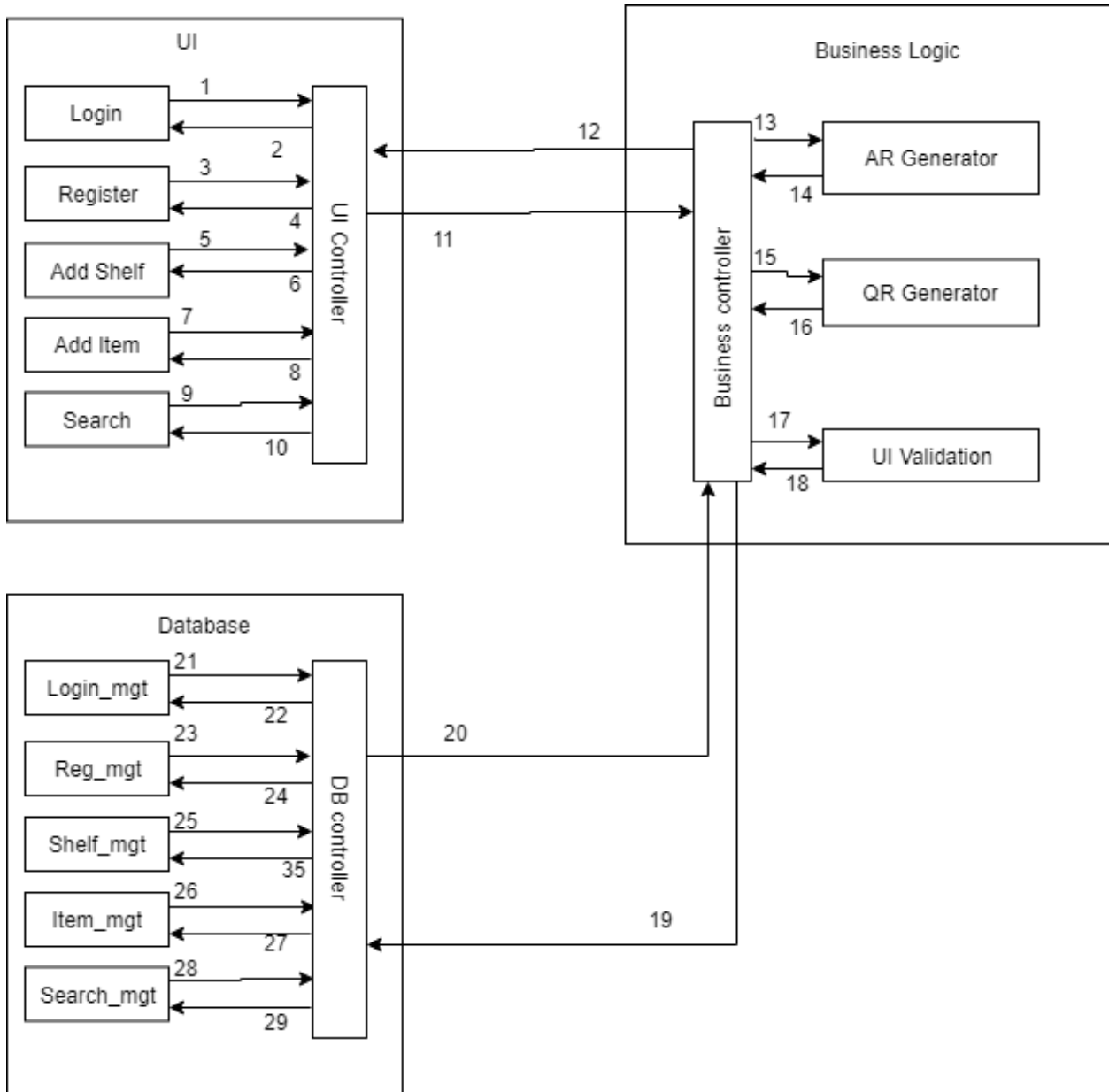


Figure 2: A simple data flow diagram

4 USER INTERFACE (UI)

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.

4.1 UI CONTROLLER

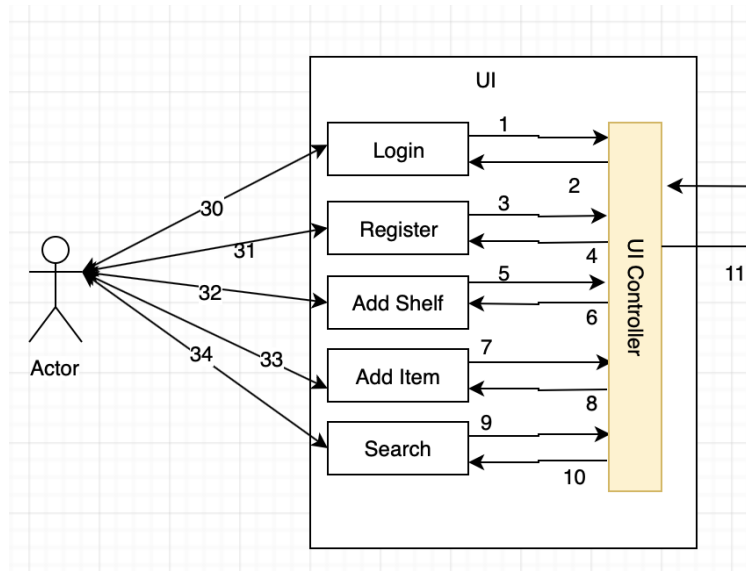


Figure 3: UI Controller description diagram

4.1.1 ASSUMPTIONS

Following assumptions are made for this SubSystem:

- User select the specific function with some input.

4.1.2 RESPONSIBILITIES

Following are the responsibilities of this SubSystem:

- To send the user request to Business_Controller.
- To receive Messages and various images returned by Business_Controller

4.1.3 UI_CONTROLLER SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here.

ID	Description	Inputs	Outputs
#1	Login/UI_Controller	Userid Password	N/A
#2	UI_Controller/Login	N/A	Success Message Failure Message
#3	Register/UI_Controller	Userid Password User Name Email	N/A
#4	UI_Controller/Register	N/A	Success Message Failure Message
#5	Add Shelf/UI_Controller	Shelf Name Row Column Shelf Location	N/A
#6	UI_Controller/Add Shelf	N/A	Success Message Failure Message
#7	Add Item/UI_Controller	Item name BarCode(opt) Quantity Expiration Date(opt) Expiration_notifi- cation(opt) PIN(opt) Brewery(if alco- hol) Style(if alcohol)	N/A
#8	UI_Controller/Add Item	N/A	Success Message Failure Message
#9	Search/UI_Controller	Item Description/ QR Code	N/A
#10	UI_Controller/Search	N/A	Image file
#11	UI_Controller/Business_Controller	User input label	N/A
#12	Business_Controller/UI_Controller	N/A	Success Message Failure Message Item image

Table 2: UI_Controller Subsystem Interfaces

4.2 LOGIN

4.2.1 ASSUMPTIONS

Following assumptions are made about this SubSystem:

- User clicked on the login or signup button in the home screen.
- Users cellphone is connected to the internet.
- User has a valid Email address.
- User provides the password that satisfies the basic password constrain

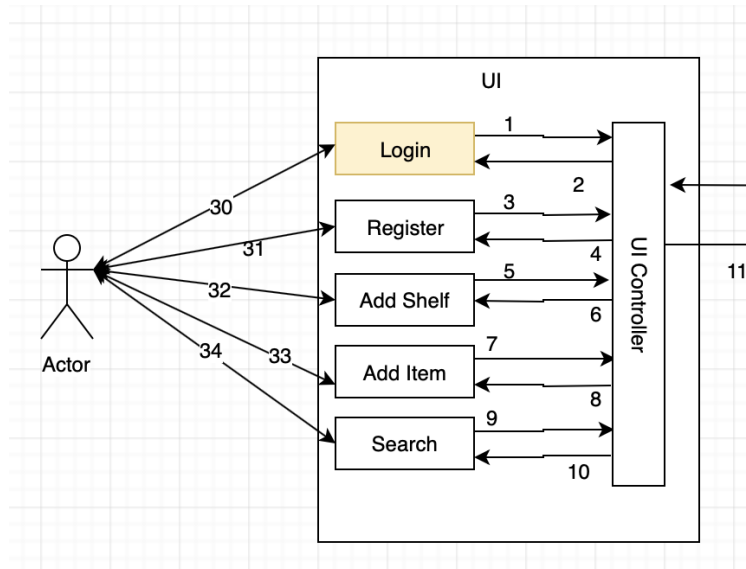


Figure 4: Login description diagram

4.2.2 RESPONSIBILITIES

These are the following responsibilities of this subsystem:

- Email address must be validated. If Email address doesnât exist or is not valid, SubSystem must prevent from creating or accessing the account.
- If the password doesnât meet minimum criteria, SubSystem must prevent user form creating account.

4.2.3 LOGIN SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. An entry for each labelled interface that connects to this subsystem is shown in the table below:

ID	Description	Inputs	Outputs
#30	Actor/ Login	Username Password	Success message Failure Message
#1	Login/UI_Controller	Userid Password	N/A
#2	UI_Controller/Login	N/A	Success Message Failure Message

Table 3: Login Subsystem Interfaces

4.3 REGISTER

4.3.1 ASSUMPTIONS

Following are the assumption of SubSection Email/Password Validation:

- User clicked on the register button.
- Users cellphone is connected to the internet.

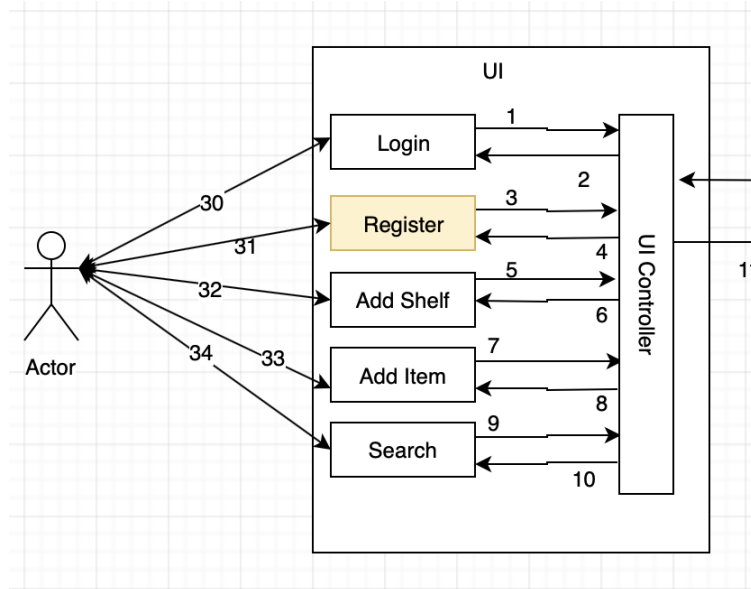


Figure 5: Register description diagram

- The user inputs all the information in the sign up form.
- The User ID and Password input is valid meeting the minimum criteria.

4.3.2 RESPONSIBILITIES

Following responsibilities must be carried out by this SubSystem:

- Email address, user name and the password is validated. If the provided email address is not valid then user is not allowed to register for the account.
- The system must only accept a unique username and a unique email address.
- User password must satisfy minimum criteria for the password constrain.

4.3.3 REGISTER 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
#31	Actor/Register	Username Password Email	Success Message Failure Message
#3	Register/UI_Controller	Userid Password User Name Email	N/A
#4	UI_Controller/Register	N/A	Success Message Failure Message

Table 4: Register Subsystem Interfaces

4.4 ADD SHELF

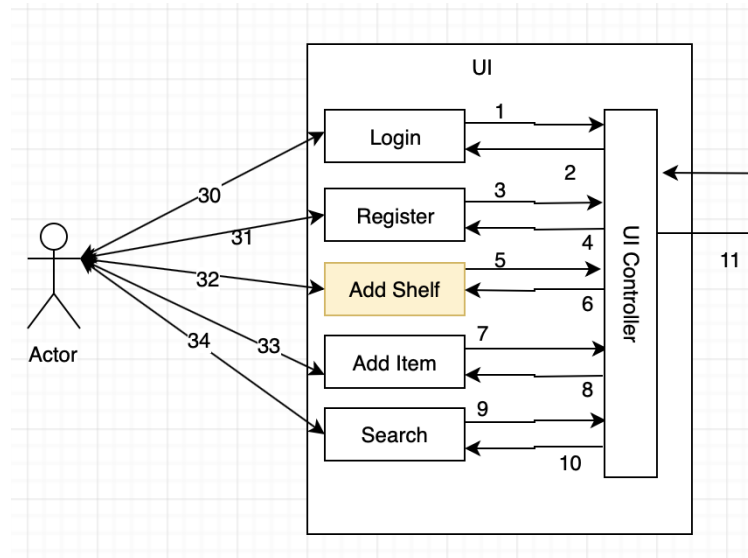


Figure 6: Adding Shelf description diagram

4.4.1 ASSUMPTIONS

Following assumptions are made about this SubSystem:

- User clicked on the add shelf button
- Users cellphone is connected to the internet.
- User inputs all the information required to create a new shelf
- User inputs a valid measurement dimensions.

4.4.2 RESPONSIBILITIES

These are the following responsibilities of this subsystem:

- The system validate the measurement.

4.4.3 ADD SHELF SUBSYSTEM INTERFACES

Each of the inputs and outputs for the Add shelf subsystem are defined here.

ID	Description	Inputs	Outputs
#32	Actor/Add shelf	Shelf Name Row Column Shelf Location	Success Message Failure Message
#5	Add Shelf/UI_Controller	Shelf Name Row Column Shelf Location	N/A
#6	UI_Controller/Add Shelf	N/A	Success Message Failure Message

Table 5: Add Shelf Subsystem Interfaces

4.5 ADD ITEM

Add Item allows the user to add the items with the description of its different attributes. Attributes like name of product, brand name, manufacture date, best by date and size of product.

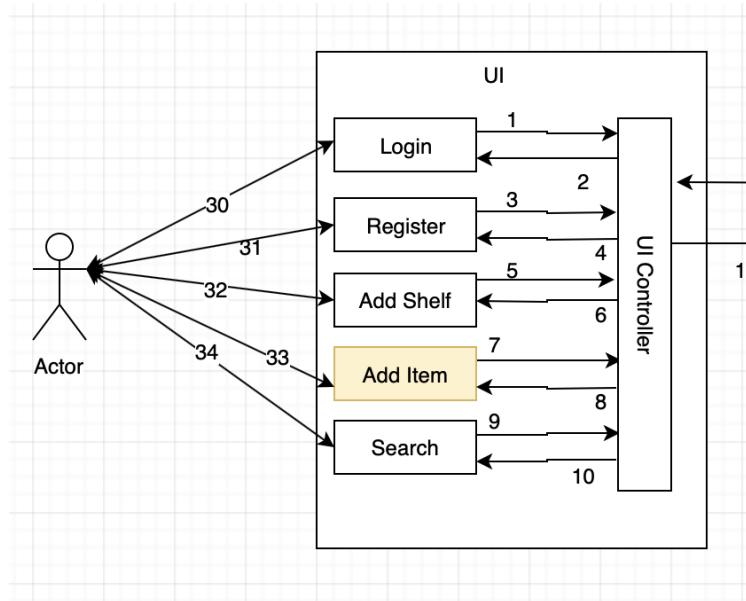


Figure 7: Adding items description diagram

4.5.1 ASSUMPTIONS

Following assumptions are made about this SubSystem:

- User clicked on add item button
- Users cell phone is connected to the internet
- User input detailed description about the item.

4.5.2 RESPONSIBILITIES

These are the following responsibilities of this subsystem:

- The system must get the detailed information from the user and send it to the UI Controller.

4.5.3 ADD ITEM 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
#33	Actor/ Add Item	Item name BarCode(opt) Quantity Expiration Date(opt) Expiration_notification(opt) PIN(opt) Brewery(if alcohol) Style(if alcohol)	Success Message Failure Message
#7	Add Item/UI_Controller	Item name BarCode(opt) Quantity Expiration Date(opt) Expiration_notification(opt) PIN(opt) Brewery(if alcohol) Style(if alcohol)	N/A
#8	UI_Controller/Add Item	N/A	Success Message Failure Message

Table 6: Add Item Subsystem Interfaces

4.6 SEARCH

4.6.1 ASSUMPTIONS

Following assumptions are made about this SubSystem:

- User clicked on the search button
- Users cell phone is connected to the internet
- User search for the item with either bar code or manually. If search is with the bar code camera is accessed to scan the bar code and if the search is manual then a text entry field is available.

4.6.2 RESPONSIBILITIES

These are the following responsibilities of this subsystem:

- The system should be able to get the data from user and send it to the UI Controller.

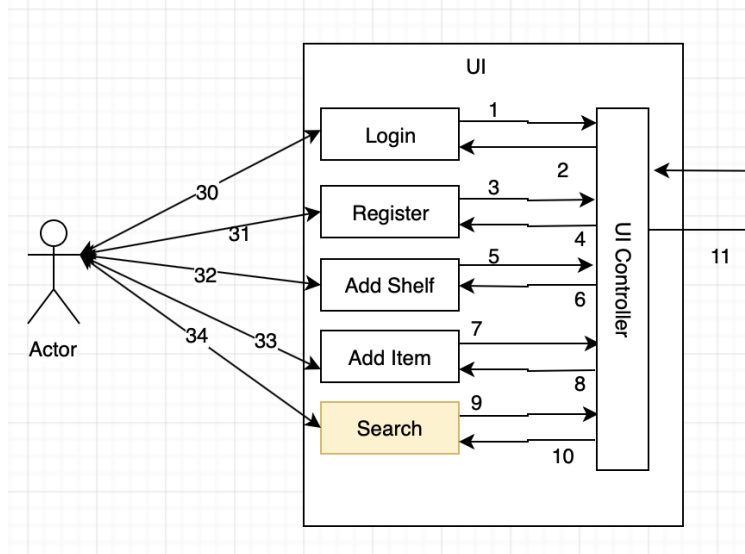


Figure 8: searching item description diagram

4.6.3 SEARCH 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
#34	Actor/Search	Item Description/ QR Code	Image file
#9	Search/UI_Controller	Item Description/ QR Code	N/A
#10	UI_Controller/Search	N/A	Image file

Table 7: Search Subsystem Interfaces

5 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.

5.1 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.

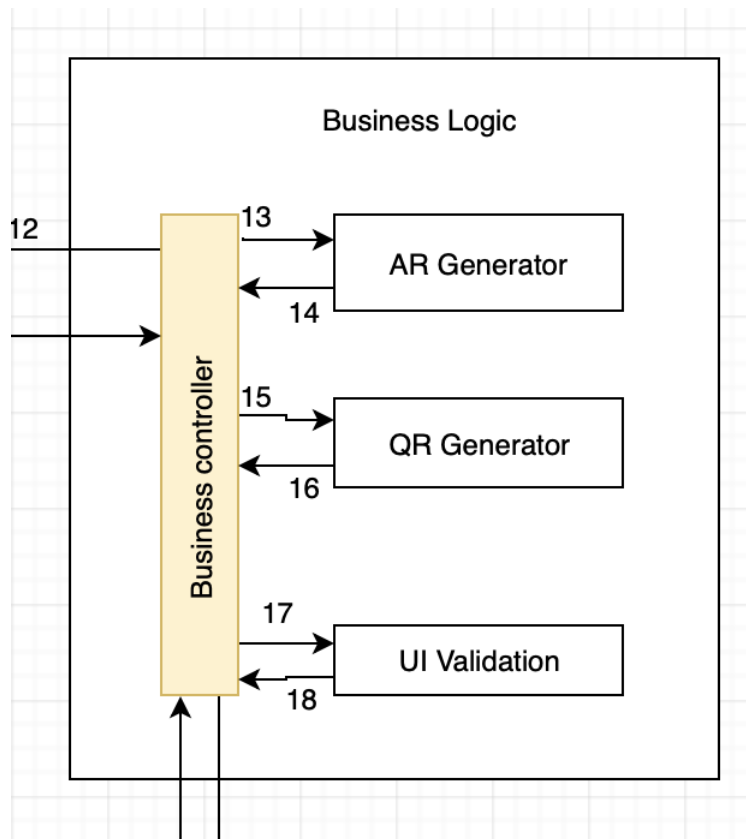


Figure 9: Business Controller Subsystem Interfaces

5.1.1 ASSUMPTIONS

- UI is working fine and UI controller has well established connection with the Business Logic.
- Database is working fine and DB controller has well established connection with the Business Logic.

5.1.2 RESPONSIBILITIES

Following are the responsibilities of the Business Logic:

- Creating QR for any new item to be stored in the inventory.
- Creating AR for any new item to be stored in the inventory.
- Validate the input coming from the UI.
- Send QR generated to be stored in Database.
- Send AR generated to be stored in the Database.
- Fetch required data from Database that are required by the UI.

5.1.3 BUSINESS CONTROLLER SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here.

ID	Description	Inputs	Outputs
#11	UI_Controller/Business_Controller	User input label	N/A
#12	Business_Controller/UI_Controller	N/A	Success Message Failure Message Item image
#13	Business Controller/AR Generator	Item from database	
#14	AR Generator/Business Controller	N/A	Item Image
#15	Business Controller/QR Generator	Item Description form UI_Controller	N/A
#16	QR Generator/Business Controller	N/A	Item description QR code
#17	Business Controller/UI Validation	Any input from UI_Controller	N/A
#18	UI Validation/Business Controller	N/A	Valid or Invalid
#19	Business Controller/DB_Controller	Any input from UI_Controller if valid	N/A
#20	DB_Controller/Business Controller	N/A	Item Success or failure Message

Table 8: Business Controller Subsystem Interfaces

5.2 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.

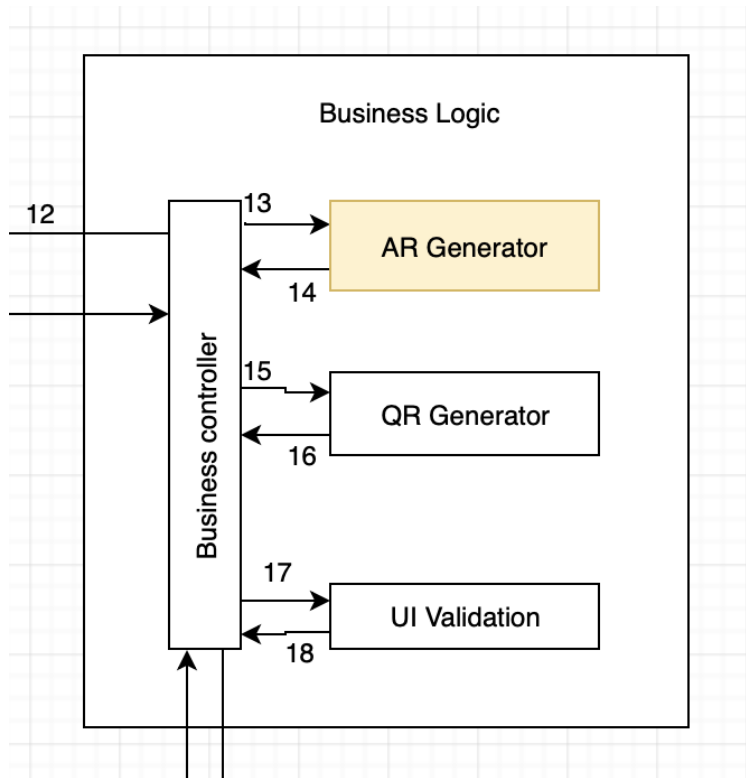


Figure 10: AR generator description diagram

5.2.1 ASSUMPTIONS

- item is in the database
- User has a device with working camera.
- User has stored valid item image for the item.

5.2.2 RESPONSIBILITIES

AR generator produces the AR and when a customer scans a bar-code for the item he is searching, the program will display the AR on the Bar-code.

5.2.3 AR GENERATOR SUBSYSTEM INTERFACES

ID	Description	Inputs	Outputs
#13	Business Controller/AR Generator	Item from database	N/A
#14	AR Generator/Business Controller	N/A	Item Image

Table 9: AR Generator Subsystem Interfaces

5.3 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.

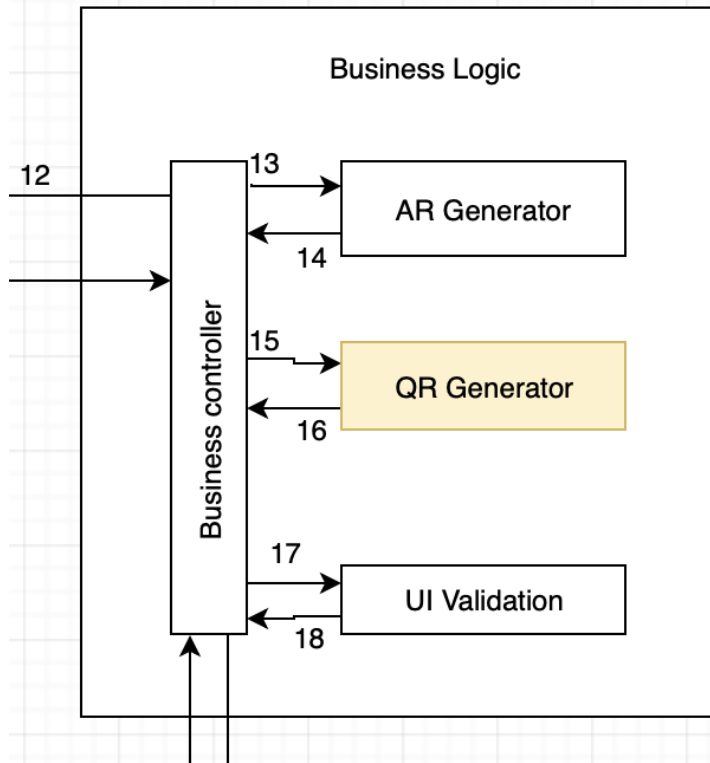


Figure 11: QR generator description diagram

5.3.1 ASSUMPTIONS

- Customer inputs the right description of the item to that is to be stored in inventory.
- Item should be new to the inventory.

5.3.2 RESPONSIBILITIES

QR generator is responsible of generating new QR for each new item that is being added to the inventory. This shall help in giving new identity to the item that is to be stored as well as make it easy to access the item stored in the inventory.

5.3.3 QR GENERATOR SUBSYSTEM INTERFACES

ID	Description	Inputs	Outputs
#15	Business Controller/QR Generator	Item Description form UI_Controller	N/A
#16	QR Generator/Business Controller	N/A	Item description QR code

Table 10: QR Generator Subsystem Interfaces

5.4 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.

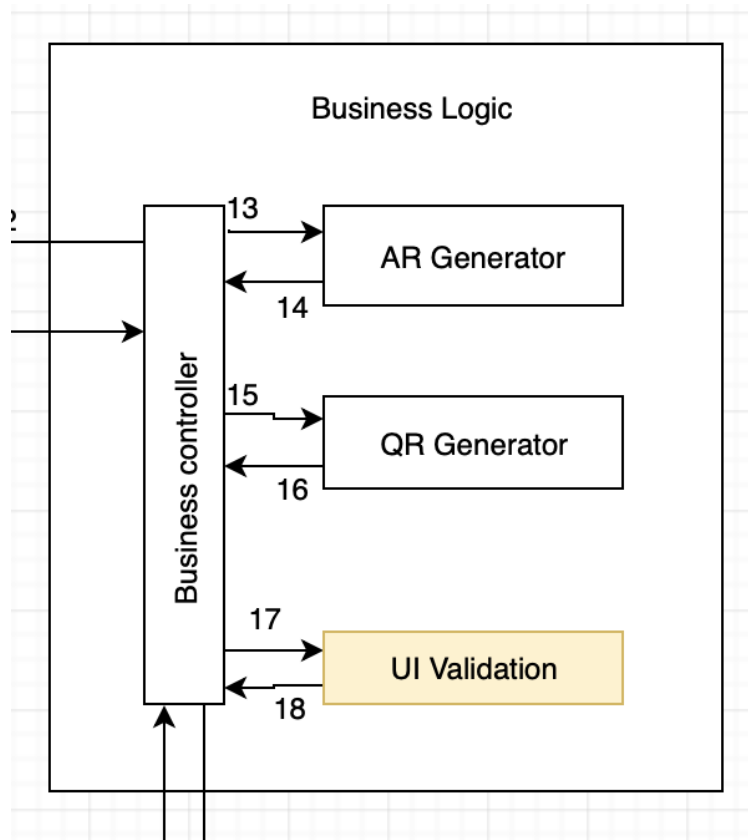


Figure 12: UI Input Validation description diagram

5.4.1 ASSUMPTIONS

- User can input item information.
- If team's assumption was correct, move forward to the next assumption.

5.4.2 RESPONSIBILITIES

- Conduct an experiment like the one listed to see if they were correct.
- Reject the SQL injection

5.4.3 UI INPUT VALIDATION SUBSYSTEM INTERFACES

Inputs and outputs for the UI validation are defined here.

ID	Description	Inputs	Outputs
#17	Business Controller/UI Validation	Any input from UI_Controller	N/A
#18	UI Validation/Business Controller	N/A	Valid or Invalid

Table 11: UI Input Validation Subsystem Interfaces

6 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.

6.1 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.

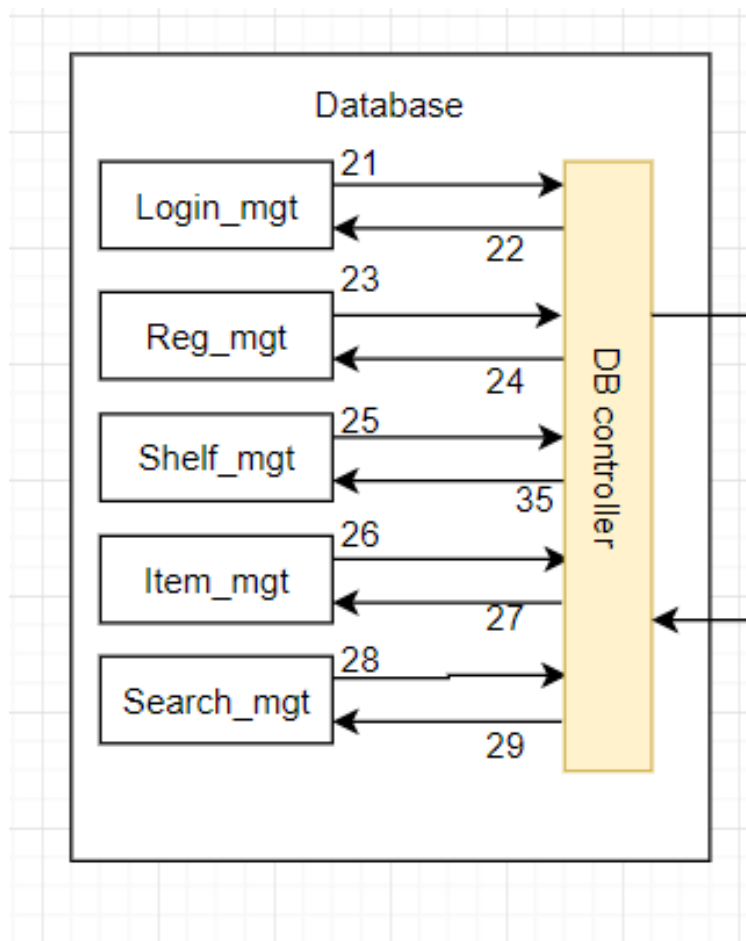


Figure 13: DB Controller description diagram

6.1.1 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.

6.1.2 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.

6.1.3 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 12: DB Controller Subsystem Interfaces

6.2 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.

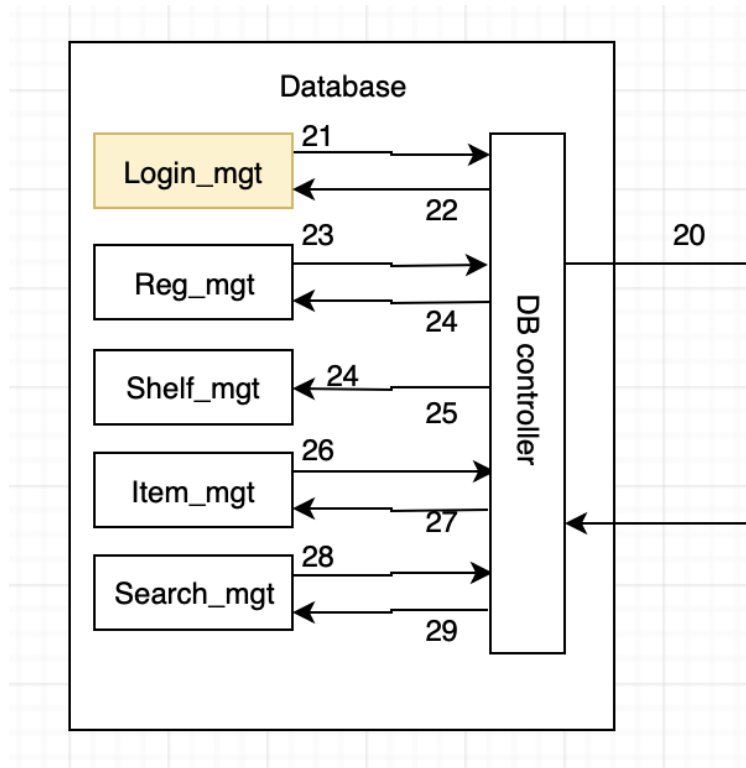


Figure 14: Login mgt description diagram

6.2.1 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.

6.2.2 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.

6.2.3 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 13: Login Mgt Subsystem Interfaces

6.3 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.

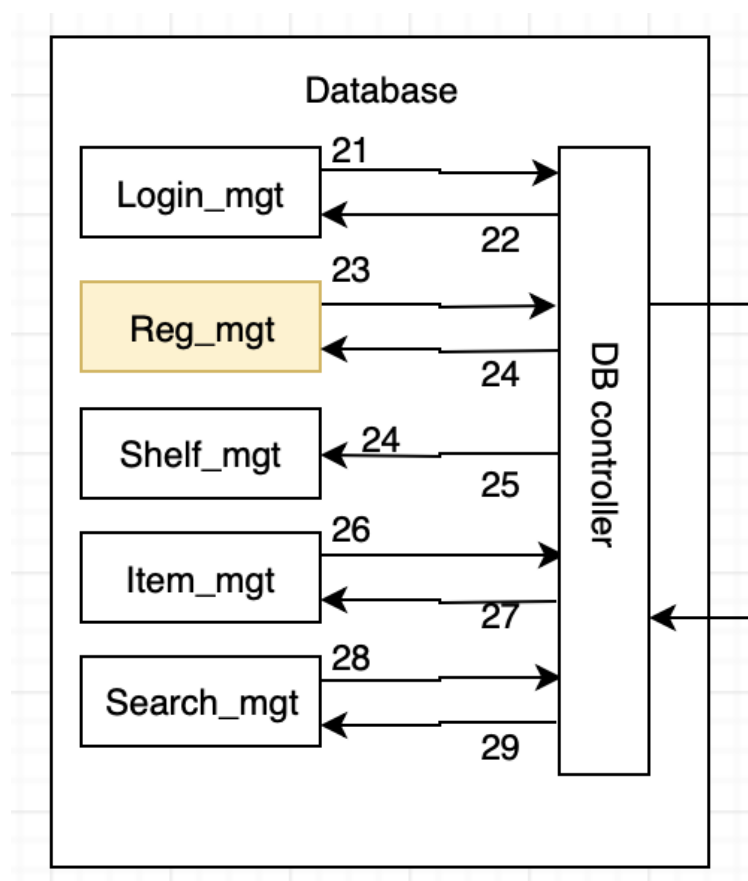


Figure 15: Register mgt description diagram

6.3.1 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.

6.3.2 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.

6.3.3 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 14: Register Mgt Subsystem Interfaces

6.4 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.

6.4.1 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.

6.4.2 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.

6.4.3 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.

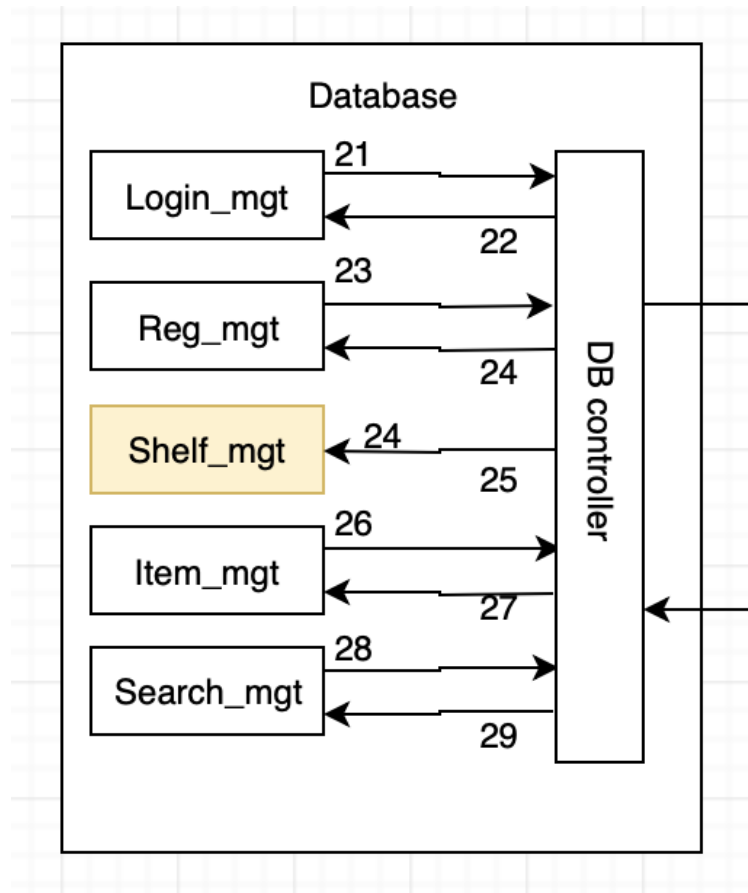


Figure 16: Shelf mgt description diagram

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 15: Shelf mgt Subsystem Interfaces

6.5 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.

6.5.1 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

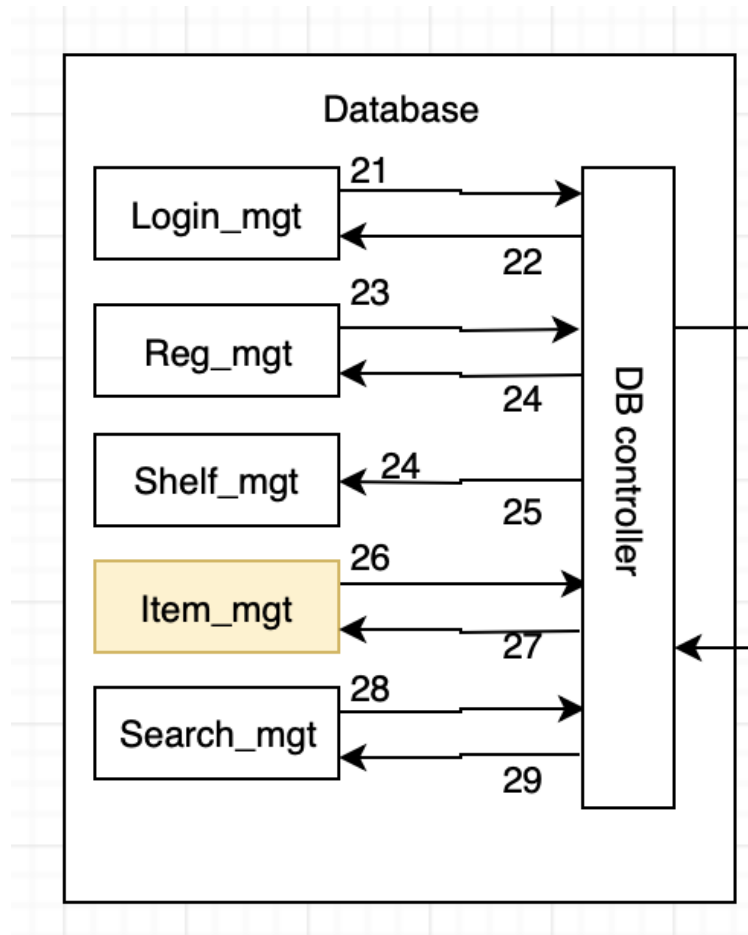


Figure 17: Item mgt description diagram

is already discarded.

- All the input value is provided by the user.

6.5.2 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.

6.5.3 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 16: Item mgt Subsystem Interfaces

6.6 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.

6.6.1 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.

6.6.2 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.

6.6.3 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 17: Search mgt Subsystem Interfaces

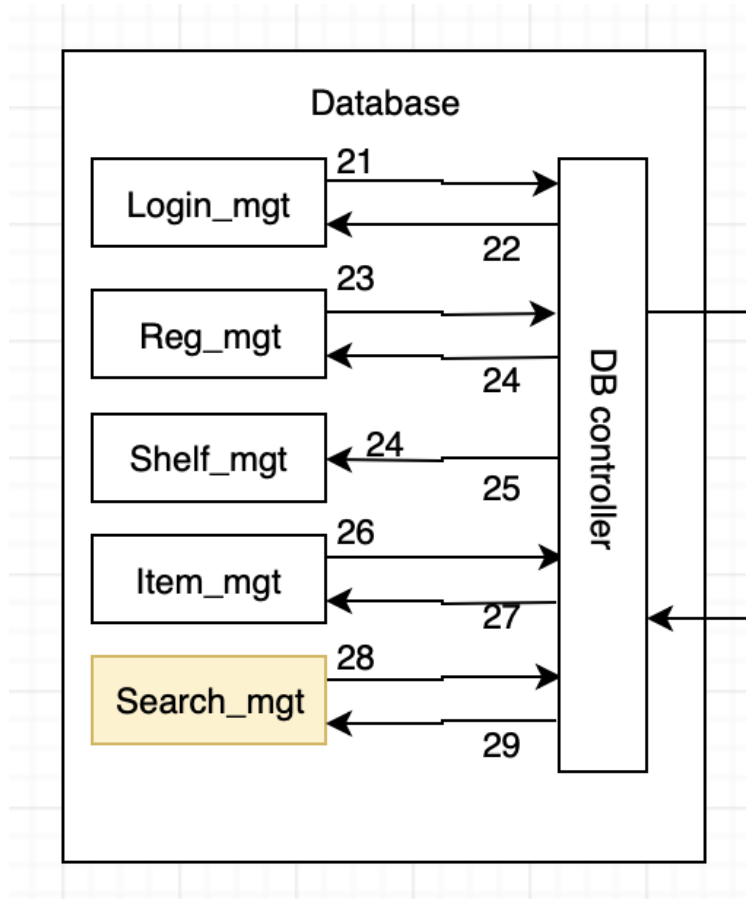


Figure 18: Search mgt description diagram

REFERENCES