Jordan University of Science and Technology
College of Computer Sciences & Information Technology

smart Shopper

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor in Software Engineering

by

karam Khaled GHARAYBEH (118980)

Ahmad Hani Al_Mahameed (122153)

Fears Moufeed Masoud (122227)

Jiyan Mohammad Malkawi (122011)

Supervised by

Dr. Mohammad Al-Zinati

Committee Members

Dr. Khaldoon T. Alzoubi

June 2020

ACKNOWLEDGEMENT

In the name of of Allah the Merciful

Praise be to God, there is much good, blessed, full of heavens and earth, full of what is in it, and peace and blessings be upon the Messenger of God, the Seal of the Messenger, and the master of fraud.

We extend our sincere thanks to Dr. Muhammad Al-Zanati:

To accept him to supervise our project. It also contributed to directing us with all available expertise to complete this project in full. He has many thanks and gratitude

We also extend our sincere thanks to everyone who contributed to the production of this project, beginning with the members of the project and everyone who contributed, from near or far, with our encouragement.

I also thank the faculty members of the Software Department, especially Dr. Khaldoon T. Alzoubi

The College of Engineering, Computer and Information Technology is general in

Jordan University of Science and Technology.

UNDERTAKING

This is to declare that the project entitled “Smart Shopper” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “Bachelor in Software Engineering” at Software Engineering Department, College of Computer and Information Technology, Jordan University of Science and Technology.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Karam Khaled GHARAYBEH (118980)

Ahmad Hani Mahameed (122153)

Fears Moufeed Masoud (122227)

Jiyan Mohammad Malkawi (122011)

ABSTRACT

The project provides “Smart Shopper “in the implementation of the technical idea which is to get the best possible options to complete the shopping process. The project relies on generating a shopping cart that provides the customer with the best products and offers that have been built depending on the features and priorities determined by the customer and the recommendations that our system collects based on the wishes of the customer. where specifically the best product is made used as evidence in product identification and value proposition, market research and evaluation and finally the commercial viability of strategic product evaluation. The purpose of the project is to implement an idea that can be seen in our world of practical benefit to the user as it is based on helping the user to get the best options to suit his own requirements.

Table Contacts

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 : Introduction

1.1 Overview

Smart Shopper is an easy to use grocery shopping assistance system. It provides the customer with a user-friendly interface to organize the shopping list. It analyzes the uploaded shopping list and provides a recommendation that directs the customer to the best places for shopping based on the customer preferences, other user recommendations, and the distance traveled to complete the shopping.

The proposed system allows subscribed stores to upload their products and prices to the system. The proposed system sorts your grocery list fitting to the shop and gives you recommendations on the total price and the distance and time required to complete shopping. As more customers use the Smart Shopping app, as better it works. It is possible to sync the grocery list with other phones

1.2 Project Motivation

Shops promote deals on specific products but sell other products with expensive prices. Customers have to go search in multiple places and waster time and effort We develop Smart Shopper system based on the previous problems and to achieve the needs of users and their desires that we have seen as we will provide solutions that facilitate the shopping process for the user intended to provide the best advice to the customer (before shopping) about the best shopping decisions and this makes our system important where we will provide the best products to the user based on recommendations, evaluation, distance traveled, quality and cost Overall, in order to achieve a technique we will use it to take and consider all relevant conditions for each user separately

1.3 Problem Statement

During shopping times Shops promote deals on specific products but sell other products with expensive prices. Customers must go search in multiple places and waste time and effort and money after a stressful workday to get daily needs and products, especially consumer products.

The “Smart Shopper” system solves many problems facing any user who wants an appropriate way to obtain the desired product by identifying all the needs that must be taken into consideration by products based on a formula that fits on a specific user to provide the best advice to the customer (before shopping) about the best shopping decisions.

It also improved user access to the product by providing it with an advanced mapping system

1.4 Project Aim and Objectives

The main goal of this system is to provide a seamless system for a comprehensive shopping plan. The information includes details of where to buy, the expected total cost, and the customer's specific desires.

“Smart Shopper” can achieve the goal by reducing the effort and cost that can be lost, as the system looks at the needs of the user who chose, arranged, and used as inputs that will provide the optimal product for the user.

1.5 Project Scope

The Smart Shopper system is a system that depends on providing products to the user through stores that provide consumables and stationery only to the user and facilitates the store's communication with its customers: the project primarily serves the store customer like (application user) to provide the best shopping you plan for. Our system then serves stores that provide consumables and stationery that provide us with offers and products to promote their products and offers. As for the rest of the stores, our system does not support them.

1.6 Project Software and Hardware Requirements

- The Internet is important to run our system
- Hardware Requirements: PC, Smart Phone, server on the cloud
- Software Requirements: any windows version, android 5.0 and newer

1.7 Project Limitations

Control restrictions:

As for the store admin, he may have difficulty providing enough information about all the information. Products owned within the store, which reduces the effectiveness of the process of displaying results and reduces the quality of work "Smart Shopper"

Smart Shopper limitations are caused by data differences (some products will not be priced), the product does not exist, expires, or the store administrator does not provide us with an appropriate description of the products. Only the difference or lack of information can affect the desired results of the searches, and some deficiencies.

Technical Limitation:

Technical restrictions: Inability of computer or mobile software or hardware for customers or the store administrator to achieve some functions in the system

1.8 Project Expected Output

The ability to present a shopping plan in a manner that is consistent with the requirements of the customer, where the Smart Shopper is intended to provide the best advice to the customer (before shopping) about the best shopping decisions. Marketing the products used as evidence in identifying the product and providing value.

Marketing heavily, making offers as possible, eliminating lost costs on traditional advertising, and investing in them for example.

Commercial viability of strategic product evaluation.

1.9 Project Schedule

Table 1-1 Project Schedule

Task Name	Start Date	Duration	End Date
Introduction	2/23/2020	1/7/1900	3/1/2020
Related Existing System	3/2/2020	1/7/1900	3/9/2020
Requirement Engineering and Analysis	3/10/2020	1/30/1900	4/9/2020
Architecture and Design	4/10/2020	1/30/1900	5/10/2020
Implementation Plan	5/11/2020	1/10/1900	5/21/2020
Testing Plan	5/22/2020	1/10/1900	6/1/2020
Conclusion and Results	6/2/2020	1/5/1900	6/7/2020
References	6/8/2020	1/3/1900	6/11/2020

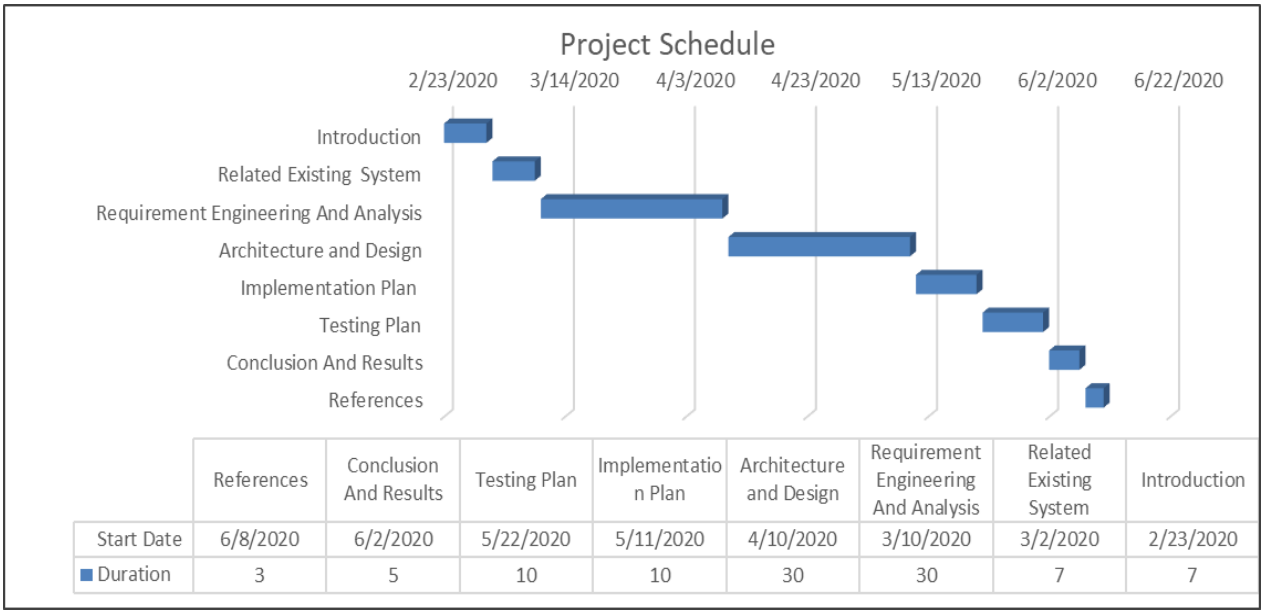


Figure 1-1 Gantt Chart of Project Schedule.

CHAPTER 2 : Related Existing System

2.1 Introduction

Smart Shopping was created to buy and sell products locally and get the best product in less time and effort.

The program is characterized by providing the product in terms of 1. Reviews 2. barcode 3. price comparison 4. website 5. offers 6. expiration date 7. goodwill 8. rating 9 quality.

The purchase list is categorized according to the customer's choice in 1. Reviews 2. Price Comparison 3. Location 4. Offers 5. Expiration Date 6. Goodwill 7. Evaluation 8. Quality.

The program supports sharing the purchase list to other destinations, saving and modifying any changes to the product

2.2 Existing Systems

Google Shopper [1]

Google Shopper provides information like prices, reviews, videos and more of millions of products right on your smartphone, recognizing them via barcode, voice, text search or even cover art. You can easily make the best purchasing decision by comparing prices in different online stores. that works on Android and IO

Groupon [2]

Groupon app is a popular mobile commerce app which brings you fresh local deals with real-time discounts every day, and it's customizable to fit what you're looking for. You can purchase and redeem deals straight from your smartphones and easily track your coupons by expiration date and location. that works on Android ,IOS ,Blackberry and Windows

PriceGrabber [3]

Another app that allows you to find product information, compare prices and get merchant ratings. Searched results can be sorted by price, rating or popularity. Biolab" that works on Android and IOS

Target [4]

With Target, you can create a product Target Lists to stay alert with its pricing. You can also design your shopping list, get special offers, check prices and buy a product using Gift Cards and redeem coupons right from your smartphone. Biolab" that works on Android and IOS.

Table 2-1 Existing Systems

	Google Shopper	Groupon	PriceGrabber	Target	smart shopper
reviews	x		x		x
barcode	x				x
compare prices					x
location		x			x
offers					x
expiration date		x			x
popularity			x		x
rating			x		x
quality					x

2.3 Overall Problems of Existing Systems :

- The first problem is that previous systems do not provide all the features
- The second problem is that previous systems define features
- The accuracy of these systems was lower in product selection

2.4 Overall Solution Approach :

- The system provides all features
- The system allows the customer to define features
- The system is more accurate by specifying the product

Table 2-2 The table shows the features of our system and other systems

Existing Systems	**Overall Problems**	**Overall Solution Approach**
Google Shopper	Products cannot be compared	The product must be compared when searching for a product
Groupon	Vouchers are tracked by expiration and location only	The current tracking must be done through evaluation, quality, price and other important characteristics
PriceGrabber	Search results cannot be sorted by nearest product	Searched results can be sorted by price, rating or popularity.
Target	The product location could not be obtained	The product can be accessed through listings

CHAPTER 3 : Requirement Engineering and Analysis

3.1 Stakeholders

Primary stakeholders:

- System Admin.
  who develop the system and monitor the system.
- Store Admin.

Those who increase products and offers (the system collects its information from the Store Admin) through the interfaces and the tools that will be available to them

- Customer.
  Who will use the app to provide him the service

3.2 Use Case Diagram

3.2.1 USE CASE SECTION

Figure 3-3-1 Use Case Diagram

Table 3-1 Flow of events for (Organize shopping list) use case

Use Case Name: Organize shopping list
Actor: Customer
Description: provide the best advice to the customer (before shopping) about the best shopping decisions.
Preconditions: The Complete login, select location information and shopping Cart not empty.
Normal Flow: 1. The user chooses to click the "organized shopping list" button. 2. The system gets the product from the customer Shopping Cart 3. The system request from the user to arrange products based on price, review, path length, arrival time selection. 4. The User price, review, path length, arrival time selection 5. The system will calculate all the options and information related to the current user to give him the best option 6. The system displays the best advice to the customer on the best shopping decisions as a list in the form of groups (baskets). 7. The user selects the best basket that includes (total cost, path, products, stores) to complete the shopping process based on the desires and priorities he previously specified.
Alternative Flow: There is No Alternative Flow
Post Conditions: The system provides the best option depending on his choices.

Table 3-2 Flow of events for (Search Product) use case.

Use Case Name: Search Product
Actor: Customer
Description: Find the Product That the Customer Search on It.
Preconditions: There Is No Preconditions for Search Product Use Cases.
Normal Flow: **1.** The User Clicks on Search Option. **2.** The User should Enter search data based on his choose **3.** For each search result **3.1.** The System Get the Result from Database **3.1.1.** The system displays a window containing a summary of product details (product name, sales center, and price) **3.1.2.** The customer scroll moves into the list up and down **4.** The customer selects "Next" or "Previous" to view other results **5.** The user chooses the product that he wants and add it to shopping Cart **6. Extend (Provide Review)**
Alternative Flow: If the Product That Get Searched Not Exist a message will appear informing the user that the product does not exist
Post Conditions: View Products

Table 3-2 Flow of events for (Share List) use case.

Use Case Name: Share Shopping Cart
Actor: Customer
Description: The customer can share the list via a public URL
Preconditions: The app should have access to Contact
Normal Flow: 1. The user clicks on Shopping Cart icon 2. The user will select the contact 3. The system Generate a public URL and send it to the previous selected contact
Alternative Flow: There is No Alternative Flow
Post Conditions: Get a public URL

Table 3-3 Flow of events for (Add Favorite) use case.

Use Case Name: Add Favorite
Actor: Customer
Description: The Customer Can Add Product on His Favorite List.
Preconditions: The Complete login
Normal Flow: 1. Include (Search Product). 2. The User clicks on Add Favorite icon. 3. The Add Favorite Icon will Change To Red Color. 4. The System Add the Product to Favorite List.
Alternative Flow: There is No Alternative Flow
Post Conditions: Add to Favorite List.

Table 3-4 Flow of events for (View Offer) use case.

Use Case Name: View Offer
Actor: Customer
Description: The Customer Can View Offer
Preconditions: The Complete login
Normal Flow: 1. The User Click on Profile. 2. The User Choose the section to which the product belongs 3. For each Offers 3.1. System Will Display the Details of The Offers (Offer Products, Expire Date, Available, Number Of Copy, Package Price, image)
Alternative Flow: There is No Alternative Flow:
Post Conditions: View section Offers

Table 3-5 Flow of events for (View Shopping Cart) use case

Use Case Name: View Shopping Cart
Actor: Customer
Description: The Customer Can View All the Products in The Shopping Cart.
Preconditions: The Complete login
Normal Flow: 1. The user clicks on the Shopping Cart icon. 2. The system displays 3. For each product in the Shopping Cart 3.1. The system displays (product name, seller, price, price, quantity) 4. If the user no longer wants a product, they can press remove button 5. For every modification in the Shopping Cart 5.1. The expected total amount is calculated and displayed 6. The customer moves between products according to the effect he has selected
Alternative Flow: 1. If the Shopping Cart is empty, the system displays an empty shopping cart 2. If the product is no longer available for some reason the system disables the product he added, and the product be out of stuck
Post Conditions: View All the Product in The Shopping Cart.

Table 3-6 Flow of events for (Find Store) use case.

Use Case Name: Find Store
Actor: Customer
Description: The Customer Can Display Store Information
Preconditions: The Complete login
Normal Flow: 1. For each Store in the Store List 1.1. The system displays (Store name, description, Hours works , open closed option) as List 2. The User Select The Store 3. The System Display Store **Extend (Provide Review)**
Alternative Flow: if the result of searching for a Store name that does Not Exist ,a message will appear informing the user that the Store does not exist
Post Conditions: View Store Information

Table 3-7 Flow of events for (View Favorite) use case.

Use Case Name: View Favorite
Actor: Customer
Description: The Customer Can View the Favorite List
Preconditions: The Complete login
Normal Flow: 1. The user clicks on Profile 2. The system displays in My Favorite container 2.1. For each product in the View Favorite list 2.1.1. The system displays (product name, price, Type , Discount, Description) 3. If the user no longer wants a product, they can press Red Favorite icon
Alternative Flow: If the user no longer wants a product, they can press Delete icon
Post Conditions: Display the favorites list that to custom

Table 3-8 Flow of events for (Sign Up) use case.

Use Case Name: Sign Up
Actor: Customer
Description: Register use case enables the customer to create a profile in the system.
Preconditions: The Customer Does Not Have an Account.
Normal Flow: 1. The User Clicks Sign Up. 2. The User Fills the Sign Up Form Correctly. 3. The User Clicks on Sign Up Button to Create an Account. 4. The System Will Validate the Input Data to Confirm Input. Data Will Be Saved into Database to Use It in Another Process
Alternative Flow: 1. If the User Enter Invalid Information, Then Refill Sign Up Form. 2. If the User already exists will have the option to recover his password or username by email.
Post Conditions: Login to Account.

Table 3-9 Flow of events for (Manage Offer) use case.

Use Case Name: Manage Offer
Actor: Store Admin
Description: The Store Admin Can Manage Offer (Create, Edit or Delete, View)
Preconditions: The Complete login
Normal Flow: 1. Choices Operation (Create, Edit , Delete ,View). 2. If the Store Admin (Create) 2.1. The Store Admin Click On Create New Offer Button 2.2. The System will display a blank form of Offer information that the store admin must fill in each field with the required information 3. If the Store Admin chooses (Edit) chooses 3.1. The store admin select the Offer. 3.2. The system will display an Offer information form that contains the information previously stored and then modify the store admin in the required field. 4. If the Store Admin chooses (Delete) 4.1. The store admin select the Offer . 4.2. the store admin selects the Offers he wants to delete and clicks on the delete icon. 5. If the Store Admin chooses (View) 5.1. The store admin select the Offer. 5.2. The System Display Offer Detail (Offer Products, Expire Date, Available, Number Of Copy, Package Price, image) 6. Store Admin clicks on the save icon.
Alternative Flow: if store admin miss mandatory field System will not complete the Operation and alarm him
Post Conditions: The system will update the Offer information.

Table 3-10 Flow of events for (Manage Product) use case.

Use Case Name: Manager Product
Actor: Store Admin
Description: The Store Admin Can Manage Product (Add, Edit, Delete, View).
Preconditions: The Complete login
Normal Flow: 1. Choose Operation (Add, Edit, Delete, View). 2. If the Store Admin Choose (Add) 2.1. the system will display a blank form of product information that the store admin must fill in each field with the required information 3. If the Store Admin Choose (Edit) 3.1. the system will display a product information form that contains the information previously stored and then modify the store admin in the required field. 4. If the Store Admin Choose (Delete) 4.1. The store admin select the Product 4.2. the store admin selects the Product he wants to delete and clicks on the delete icon. 5. If the Store Admin Choose (View) 5.1. The store admin select the Product 5.2. The System Display The Page Of Product 6. When the system admin applies all changes then the system updates the product list.
Alternative Flow: if store admin miss mandatory field System will not complete the Operation and alarm him
Post Conditions: View section Offers

Table 3-11 Flow of events for (Provide Review) use case

Use Case Name: Provide Review
Actor: Customer
Description: Customer can provide Review about a product or store
Preconditions: The Complete login
Normal Flow: 1. The user Set Review by Rate and text. 2. The System calculates the new average Review and updates the product rate. 3. The System Post the Review on Review list window.
Alternative Flow: There is No Alternative Flow
Post Conditions: Add To provide Review

Table 3-12 Flow of events for (Manage Advertisement) use case.

Use Case Name: Manage Advertisement
Actor: System Admin
Description: The authentication system administrator can display the offer or confirm the type of offer content, for posting it or blocking the display and re-editing it to the store administrator (the advertiser) and specifying the location of the display and the duration of the display
Preconditions: The Complete login. Add new ads from the store admin.
Normal Flow: 1. The system provides notification of a new announcement to the system administrator, when it has been added or edit 2. The system administrator makes sure that the ad meets the agreed terms. 3. The system administrator accepts the duration and location of the advertisement. 4. The system administrator clicks on the approval icon 5. The system provides notification of a new announcement to users interested in a particular 6. category that has been modified or added to the system administrator.
Alternative Flow: If the ad does not meet the agreed terms, the system administrator will stop publishing, blogging a comment, and returning it to the store administrator.
Post Conditions: system displays ads according to the predetermined location and time.

Table 3-13 Flow of events for (Manage Shopping Cart) use case.

Use Case Name: Manage Shopping Cart
Actor: Customer
Description: The Customer Can Add Products to Shopping Cart
Preconditions: The Complete login
1. Normal Flow: 2. Choose Operation (Add, Edit, remove, View). 3. If The User want (Add) product 3.1. Include (Search Product). 3.2. The User Select Quantity. 3.3. The User Click Add Button to add the product Shopping Cart Icon. 3.4. The System Added the Product to Shop List. 4. If The User want (remove) product 4.1. The User Click on My Cart From Navbar 4.2. The User Select Quantity . 4.3. The User Click remove Button to delete the product from Shopping Cart List. 5. If The User want (update) product 5.1. The User Click on My Cart From Navbar 5.2. The User Select Quantity. 5.3. The system update the new Quantity. 6. If The User want (View) product 6.1. The User Click on My Cart From Navpar 6.2. For each product in the Shopping Cart list 6.2.1. The system displays (product name, price, Type , Discount, Description)
Alternative Flow: If the product is no longer available for some reason the system disables Add Button and the product
Post Conditions: View an updated Shopping List

Table 3-14 Flow of events for (Manage Accounts) use case.

Use Case Name: Manage Accounts
Actor: System Admin
Description: The System admin is responsible for the management of customers and store admin
Preconditions: The Complete login.
Normal Flow: 1. The system admin chooses an Manage Account. 2. The system will view the accounts in the list. 3. The system admin search by Email or Name. 4. The system displayed the specified account. 5. The system admin chooses operations (Add, Deactivate, Activate, View information, Modify).
Alternative Flow: There is No Alternative Flow :.
Post Conditions: Save all changes.

3.3 Non-functional requirements

- **Availability**

Availability The system should be available to service based on user request at any required time. This means that the user can access our website by entering the URL of the site in any browser or through the program from any device that supports the requirements to display the content or provide the required service

And we achieve this by making our database on server on cloud and make our app running on rented Server and have backup on system.

- **usability**

usability system should be easy to handle and learn to serve all types of users by having a clear interface and ease of use for each feature and easy to navigate. For example, the customer can add any product he wants in 3 steps, and he can get the full plan in 4-5 steps.

We followed the HCI rules in designing and identifying user oriented.

- **Performance**

The system must be responsive, and we accomplish this by

Increase the number of online users at the same time

And the response to a user operation in real-time

- **Security**

The passwords are hashed in the database.

restriction on accounts and their access rights.

- **Maintainability**

The system should be easy to maintain, such as editing or adding more features. We accomplish this by designing the classes based on

low Cohesion

high Coupling

follow SOLID principles

- **Scalability**

The System should be able to accommodate additional 100,000 users.

The system can handle 10,000 requests at a time

3.4 Constraints

Domain Constraints:

There are a few restrictions that the system must follow: all entries must be validated to validate, and messages must be given incorrect data. Invalid data should be ignored, and error messages should be given. The details provided by the seller while being registered should be stored in the database. While adding products to the system, mandatory fields must be checked to verify if the store administrator has filled out the appropriate data in these mandatory fields. If not, an appropriate error message will be displayed otherwise data will be displayed

Cost Constraints:

The main cost constraint of the system is the method of financing the system and providing enough money to modify and maintain, increase the volume of data assimilation and add other future services and keep pace with the amount of options and alternatives.

Time constraints:

The project may not be delivered on time; Additional time may be required, the main difficulty facing the project team is finding the time to carry out the meetings without interruption in communication and times not inconsistent with the lectures and tests for other materials.

CHAPTER 4 : Architecture and Design

4.1 Overview

- The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data.
- The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template.
- Although Django follows MVC pattern but maintains its own conventions. So, the framework handles control itself.
- There is no separate controller, and complete application is based on Model View and Template. That's why it is called MVT application. See the following graph that shows the MVT based control flow.

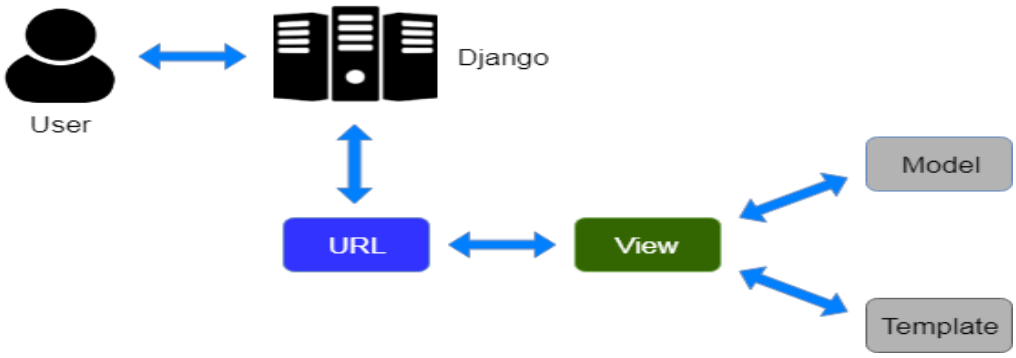


Figure 4-1 Three Layers (3-Tiers) Architectural pattern [5]

2 Software architecture

4.2.1 LOGICAL VIEW

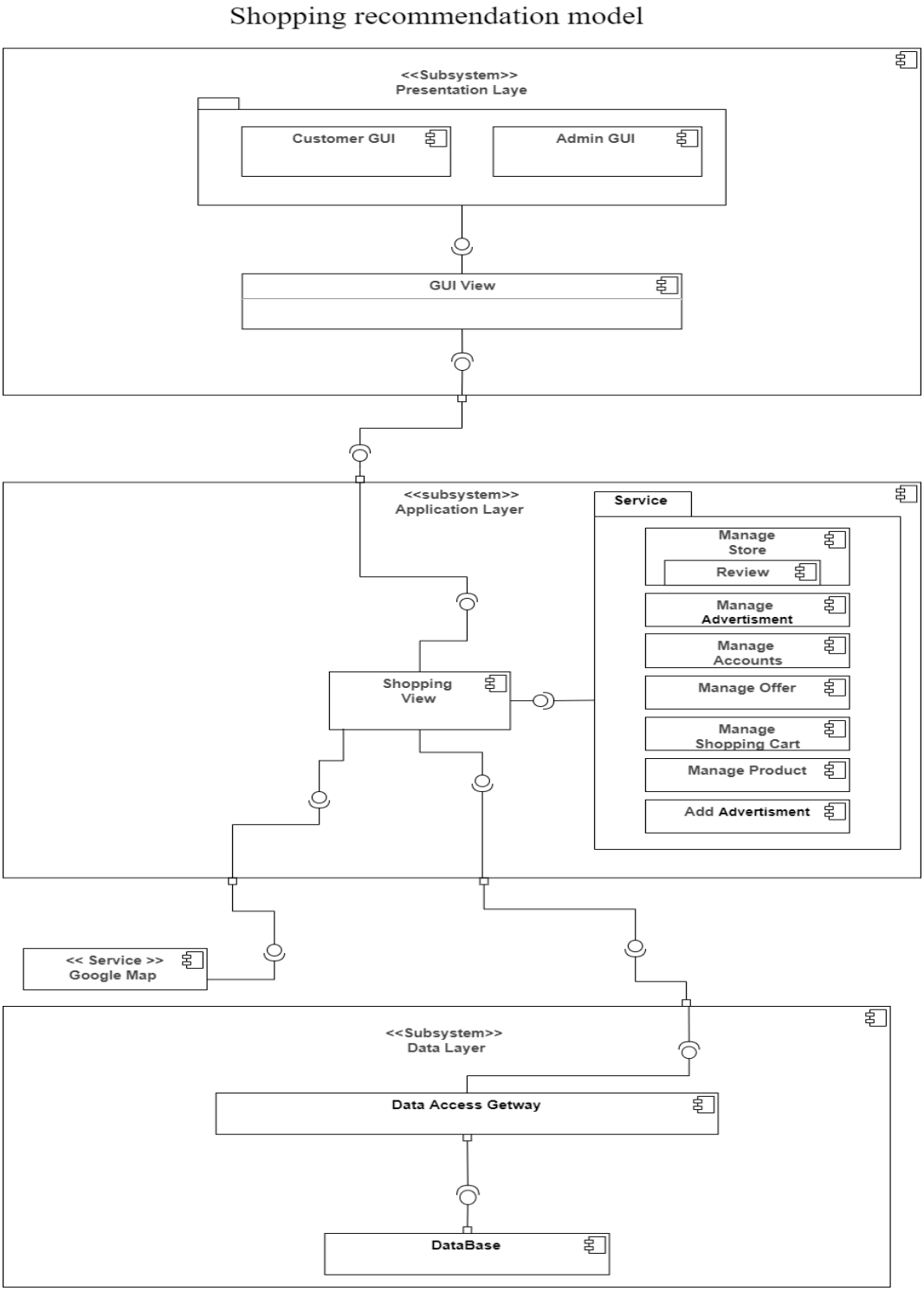


Figure 4-2 System Logical view (shopping recommendation model diagram)

4.2.2 PROCESS VIEW

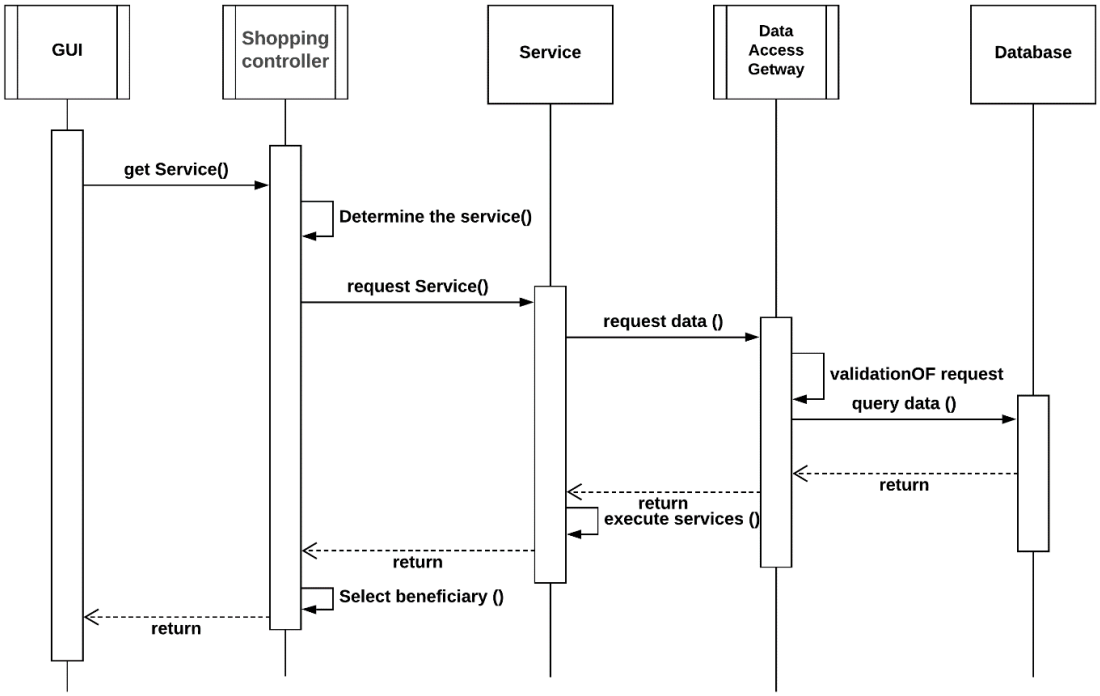


Figure 4-2 System Process view (Sequence diagram)

4.2.3 PHYSICAL VIEW

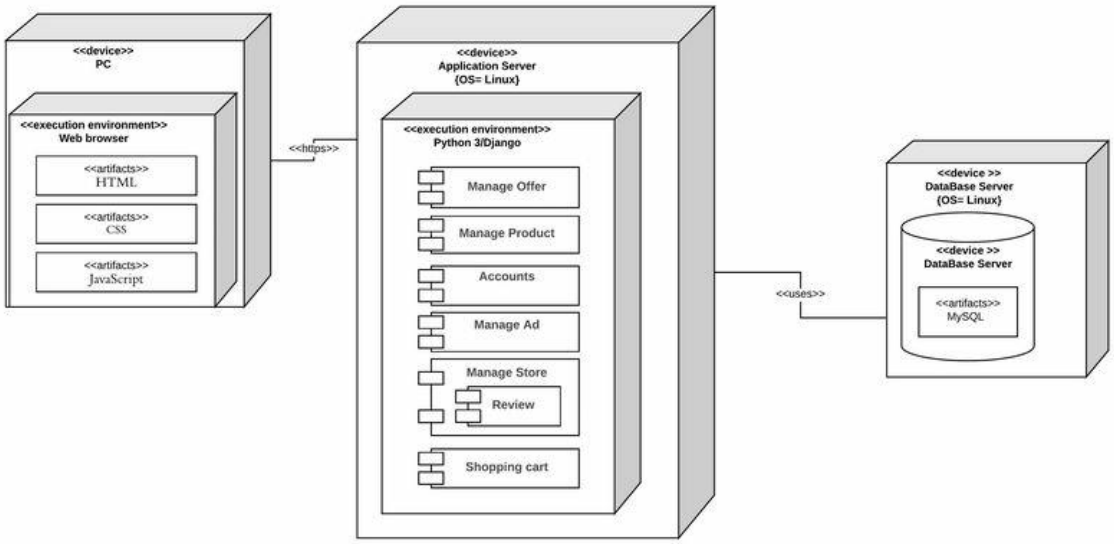
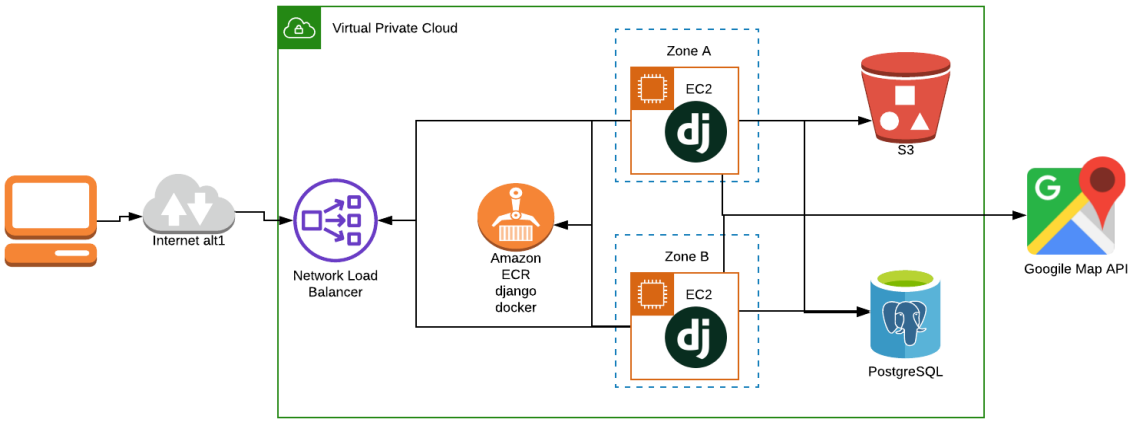


Figure 4-3 System Physical view (Deployment diagram)



4.2.4 DETAILS OF EACH COMPONENT IN A SEPARATE SECTION.

Presentation Tier

GUI customer: This component is responsible for the Services Manager graphical user interface provided by the system, because it identifies and manages the process of sending and receiving information, and identifies the user and the beneficiary, and manages some of the authorities

GUI Store Admin: component provides Admin services when logging in as the system administrator, as it can take advantage of authorized system services by communicating with the component that controls the graphical user interface

GUI View: It is responsible for the front-view layer in the user interface system as it is provided for sender and receiver identification, recipient and sender validation, data processing, and transfer to the application layer via its API

Application Tier

Model: This component provides results about the best options and possibilities for creating a shopping plan for a group of different products that match the customer's wishes and is responsible for research .

Manage Advertisement: This component is responsible for providing notification services and managing the advertising process in terms of ad location, ad time, ad period, and ad serving mechanism, as well as identifying customers interested in a specific product.

Manage Store: Representing the store and the services it provides from the products and offers offered in the market

Review: It is a process based on the evaluation of products and stores by expressing the user's comments and rate in the description box

Manage Accounts: This component is allowing the account owner to accounts by edit, add and delete his information and some operations related to account management

Shopping View: This component is responsible for application to business logic that drives the basic capabilities of the application to determine the type of service, verify the validity of the recipient and the sender, process data, send them to the database, and take advantage of the services provided from the outside the system

Data Tier

Database: is information that is organized into tables and stored where many operations can be performed from modification to search to extraction to deletion.

Data Access Gateway: is responsible for data level services, database / data storage management system, data access layer and communication validation. The data is accessed via the application layer via API calls.

4.3 Software design

4.3.1 WHITE BOX UML SEQUENCE DIAGRAM

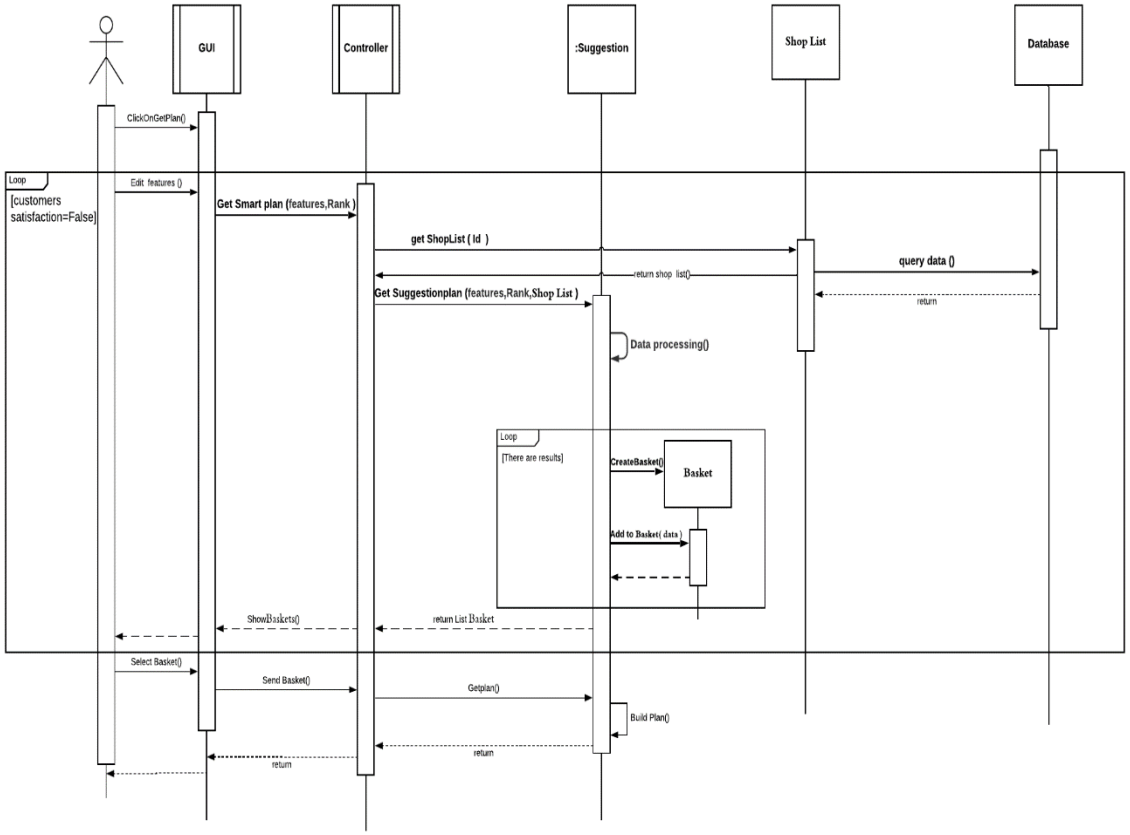


Figure 4-4 white box Sequence Diagram (Organize shopping list)

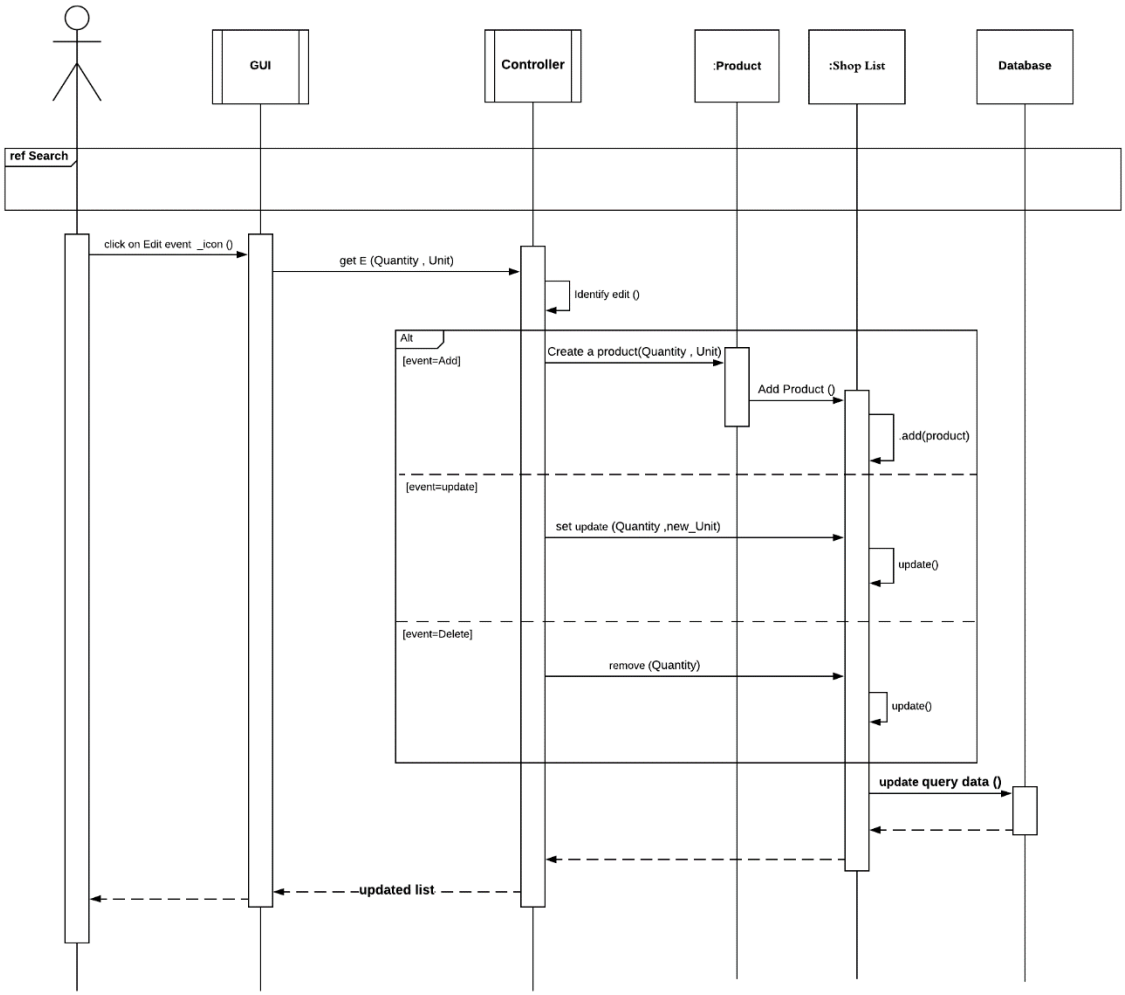
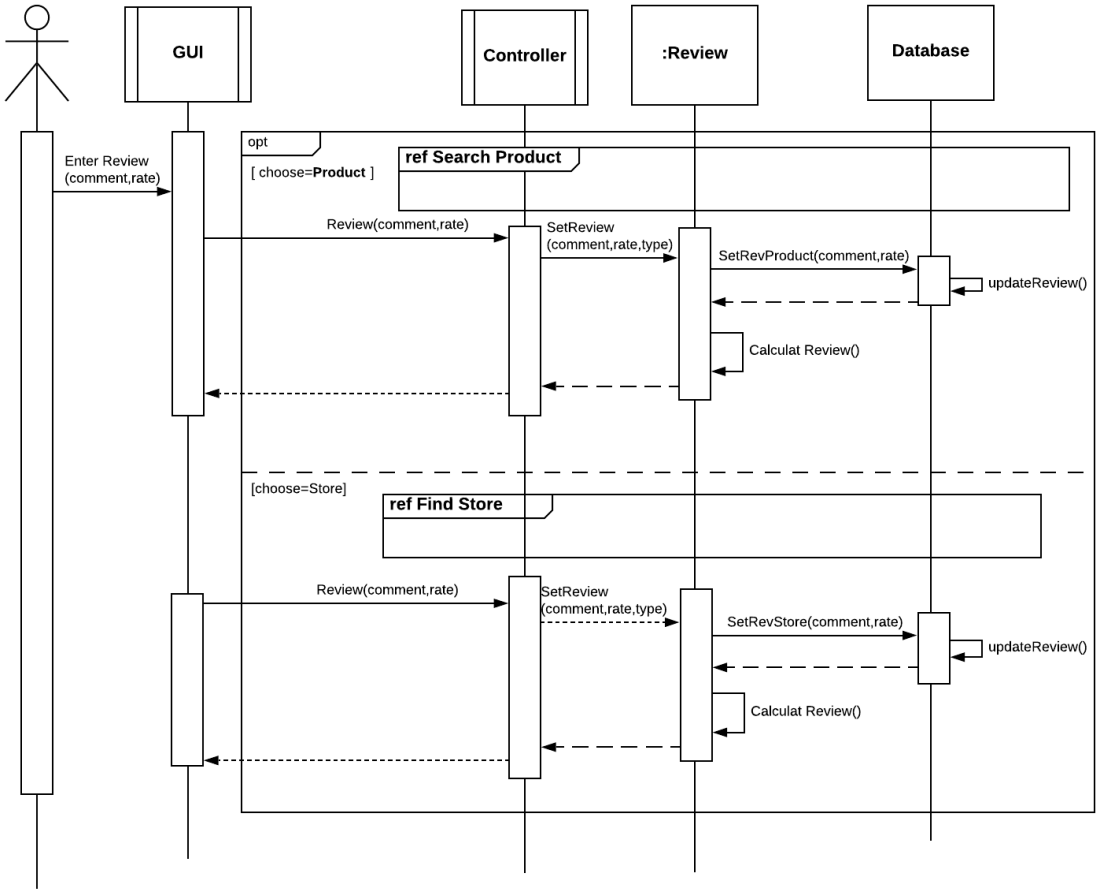
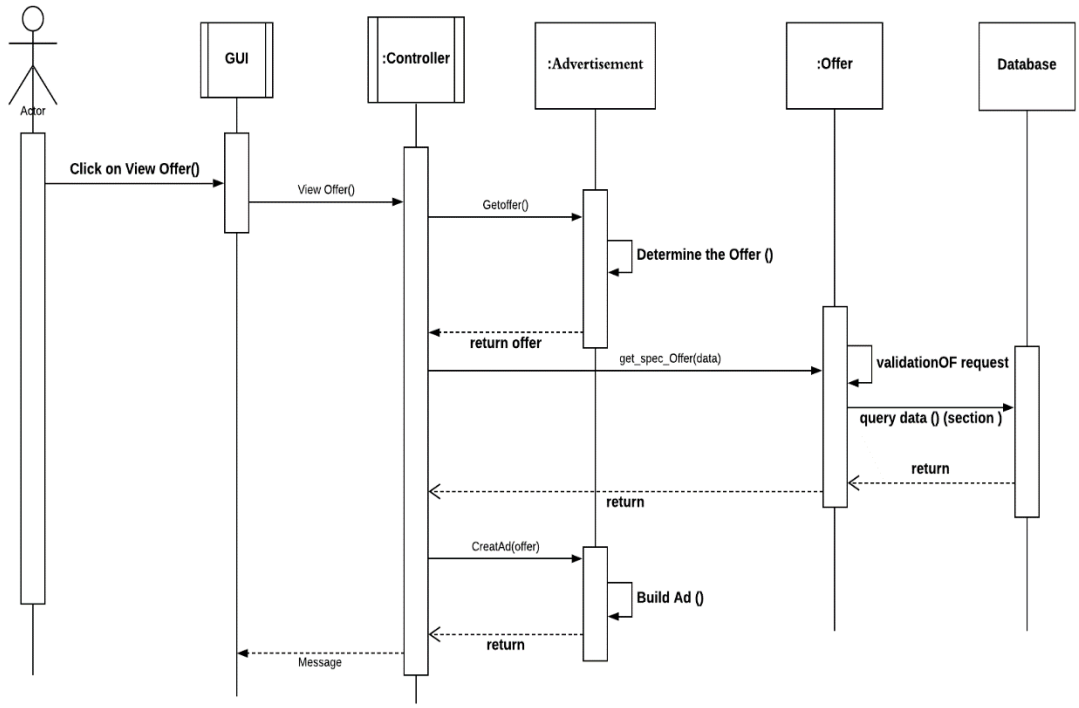


Figure 4-5white box sequence Diagram (Edit shopping list)



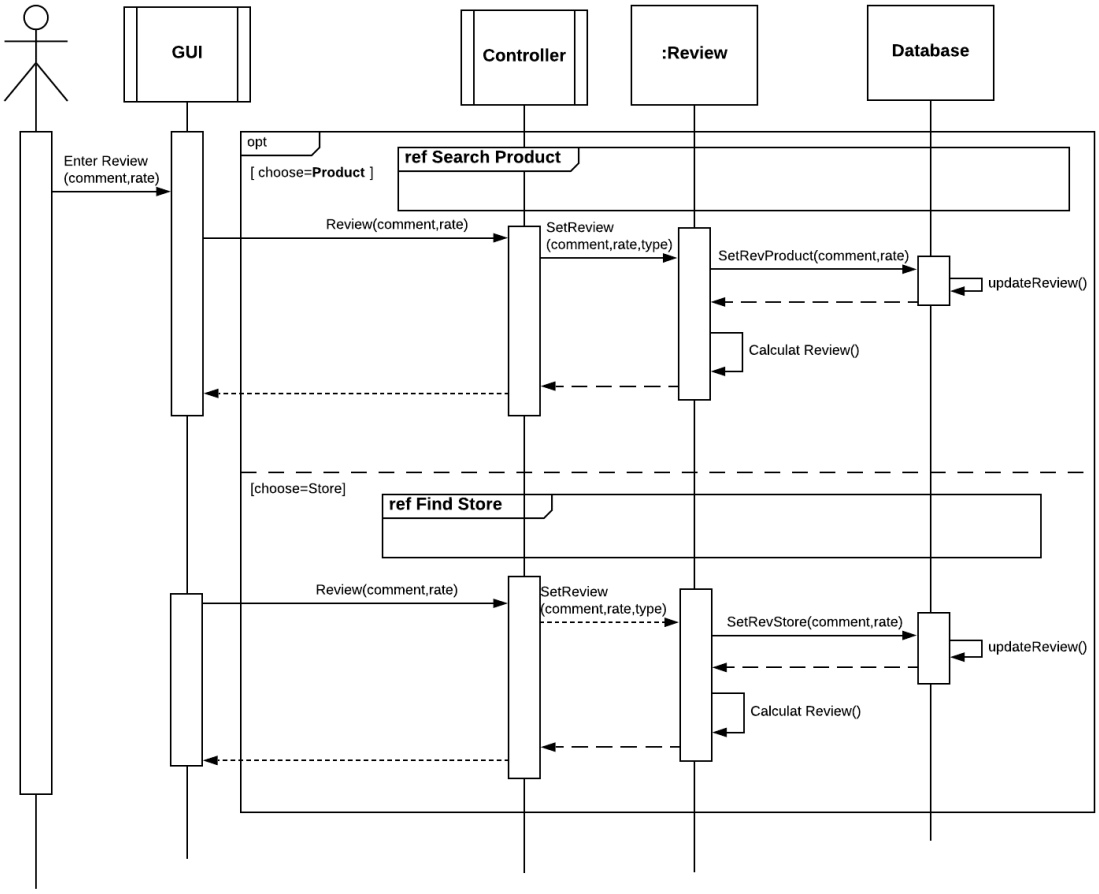
white box Sequence Diagram (Provide Review)

Figure 4-6 white box Sequence Diagram (Provide Review)



White Box sequence Diagram (view offer)

Figure 4-7 White Box sequence Diagram (view offer)



white box Sequence Diagram (Provide Review)

Figure 4-8 white box Sequence Diagram (Provide Review)

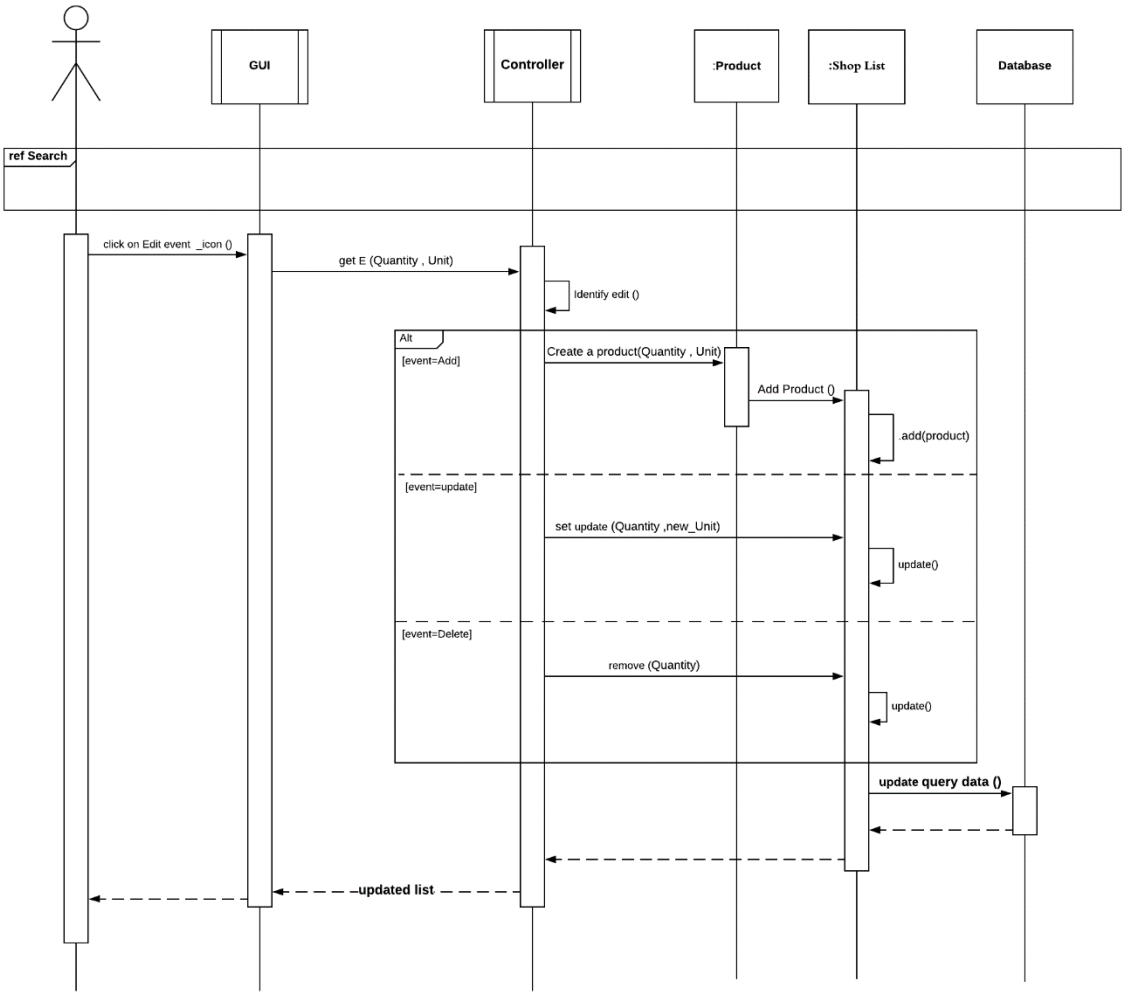


Figure 4-9White Box sequence Diagram (view shopping Cart & share)

Figure 4-10 white box sequence Diagram (View Reward)

4.3.2 CLASS DIAGRAM

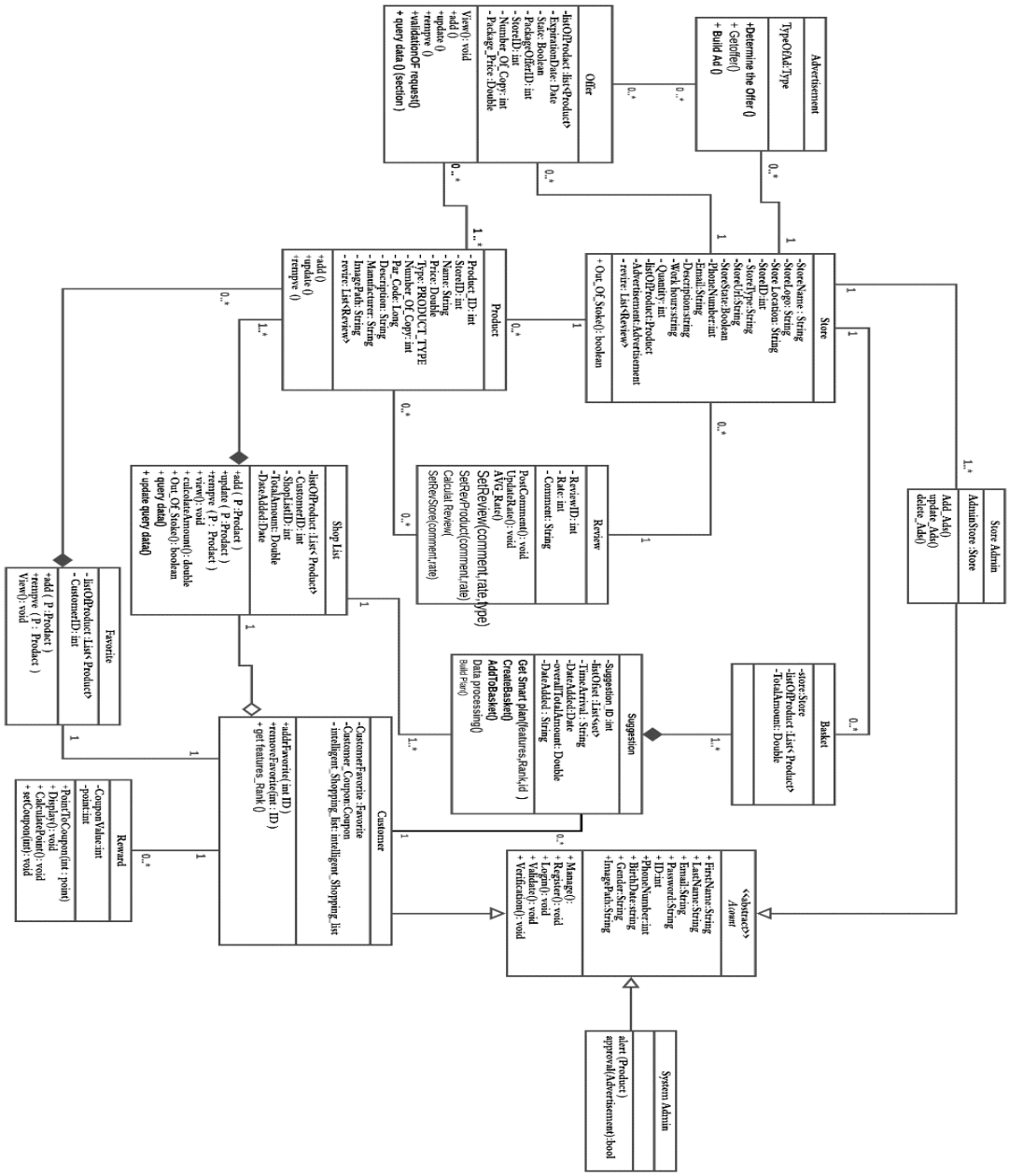


Figure 4-11 Class diagram

4.3.3 ER DIAGRAM

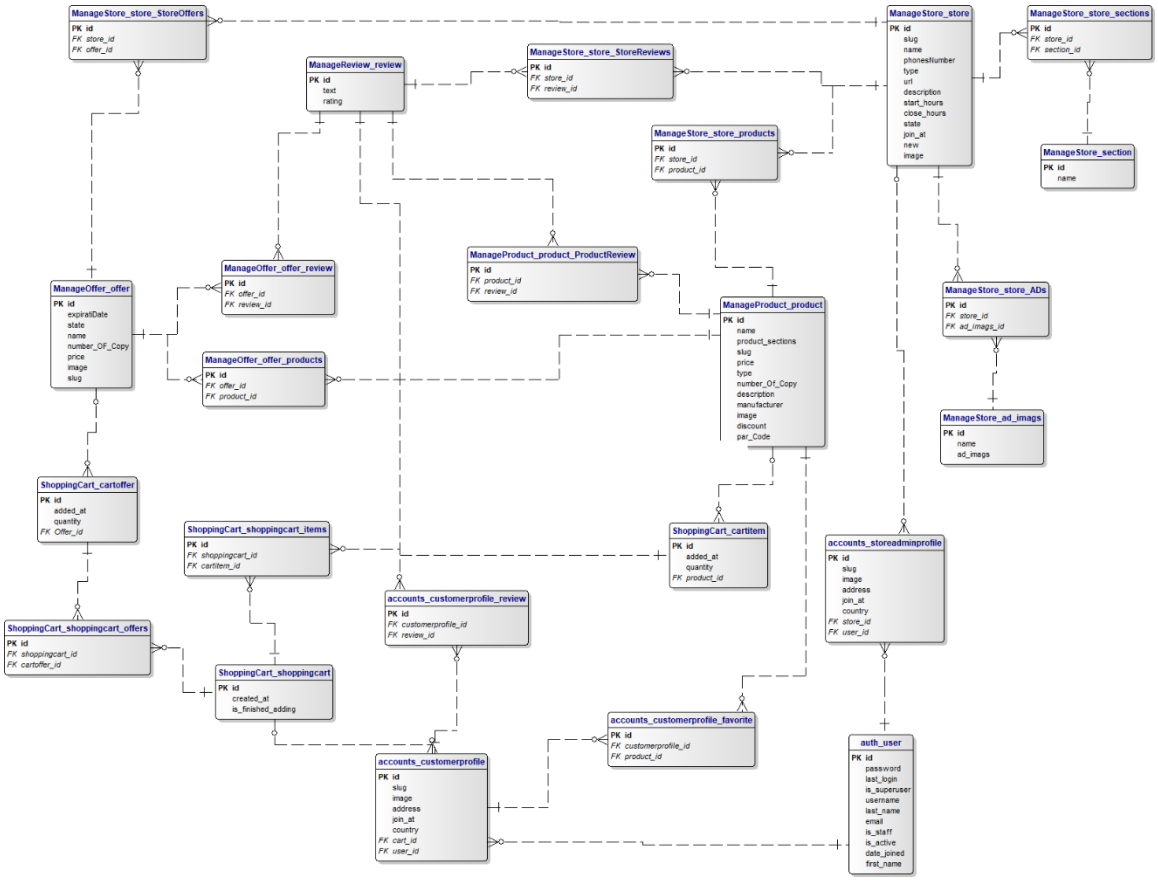


Figure 4-12 ER diagram

4.3.4 ACTIVITY DIAGRAM

4.4 User interface design (prototype)

1. Login.
2. Sign up.
3. Manage product for store admin actor.
4. Manage offer for store admin actor.

We connect our part of implementation to online database server, and using picas (2.5.0) library to control the image that store admin will upload to our database.

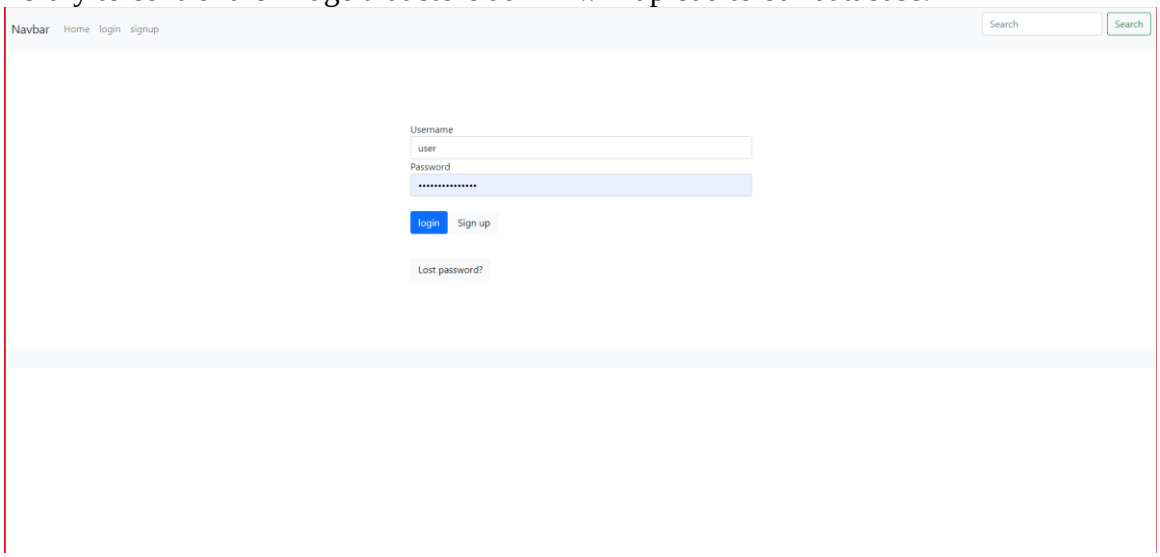


Figure 13login screen

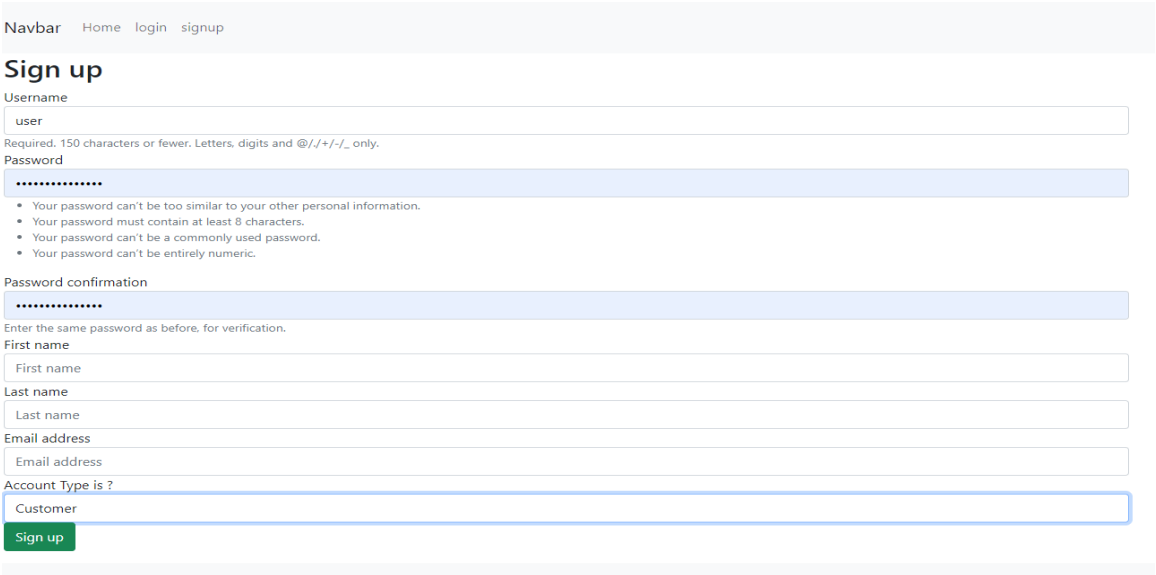


Figure 14 Sign up

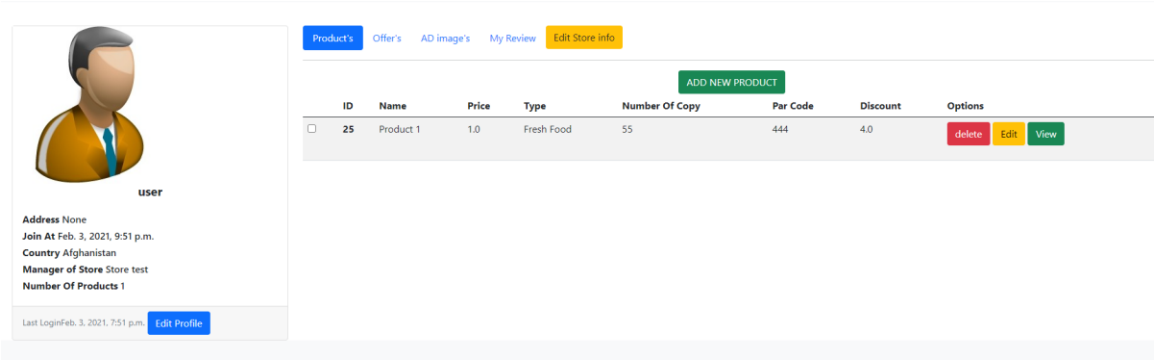


Figure 15 Manage product for store admin actor.

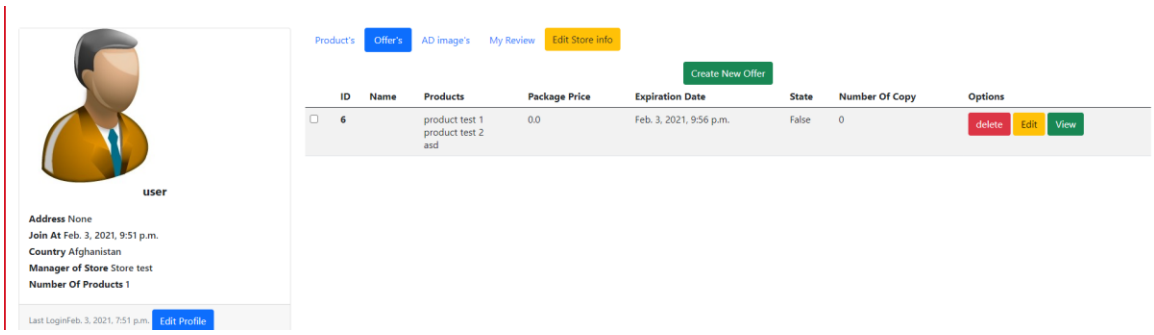


Figure 16 Manage offer for store admin actor

CHAPTER 5 : Implementation Plan

5.1 Description of Implementation

Table 5-1 Description of Implementation

Tisk	Names member	Start Date	Duration (D)
Project Management Service	Feras &Jiyan	7/15/2020	20
User Management Service	Ahmad	7/15/2020	20
Frameworks Services	Karam	7/15/2020	20
Creating user guide and demo	The full group	8/6/2020	10
System integration	Ahmad &feras	8/17/2020	25
Data-set Services	Jiyan	8/17/2020	34
Deployment of the webserver	Karam	9/2/2020	10
Reviews Services	Feras	9/12/2020	10
Deployment of the database server	Jiyan &Feras	9/23/2020	12
Client-Side	Jiyan &Feras	10/6/2020	16
Deployment of the application server	Jiyan &Feras	10/22/2020	16
Integration testing	Karam & Ahmad	11/9/2020	14
Unit testing	Jiyan &Feras	11/9/2020	14
System testing	The full group	11/25/2020	14
Accepting testing	The full group	12/11/2020	14
System delivery	The full group	12/18/2020	7

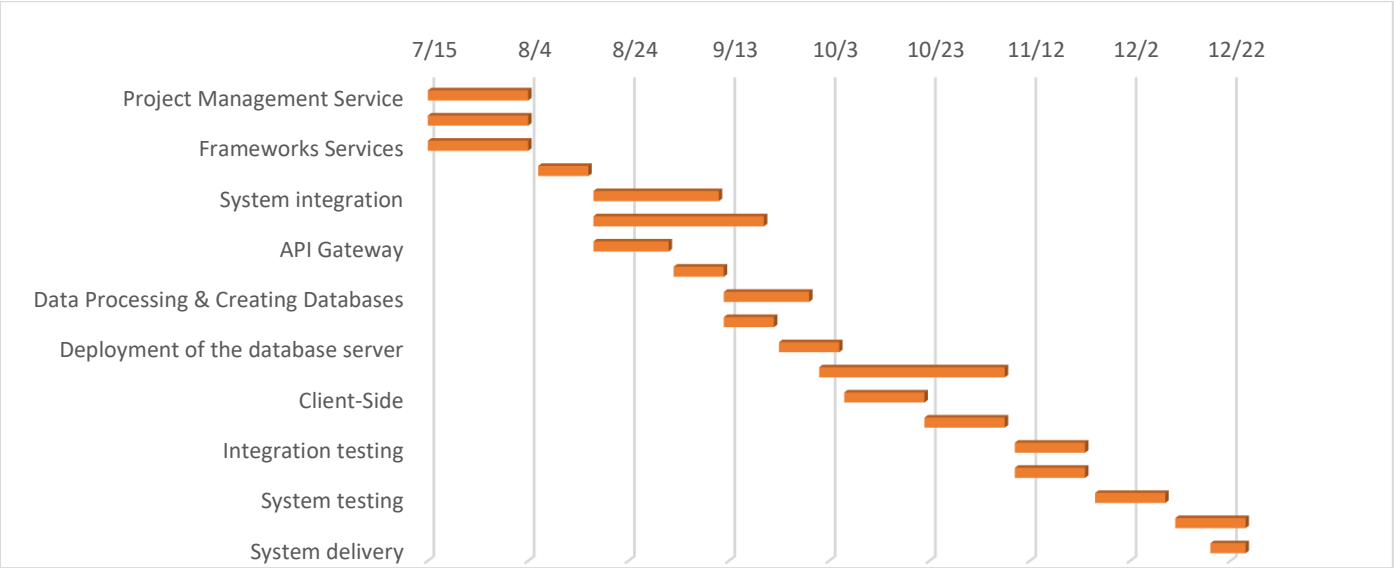


Figure 5-1Gantt Chart of Description of Implementation

5.2 Programming language and technology

language and technology Programming languages that will be used in :

Client-Side:

1) HTML
2) CSS
3) JavaScript

Database

1) MySQL database

Data Processing:

1) Python 3
2) Django

Server Side:

1) ubuntu

5.3 part of implementation

In part of implementation we decided to provide 4 use case:

1. Login.
2. Register.
3. Manage product for store admin actor.
4. Manage offer for store admin actor.

We connect our part of implementation to online database server, and using picas (2.5.0) library to control the image that store admin will upload to our database.

CHAPTER 6 : Testing Plan

The software testing process is one of the fundamental processes in software development, where every successful software product is tested in one way or another. However, the testing process often has to run on limited resources in terms of time or money. To compensate for the lack of resources, the testing process can be modified to comply with the restrictions set by the operational ecosystem; In fact, studies have concluded that adequate testing can be achieved with a low amount of resources, even as low as 15% of the resources required. On the other hand, it is also reasonable to say that software testing can become costly and wasted if done without any prior planning. A comprehensive set of test cases including all scenarios and possible outcomes cannot be performed when software complexity begins to rise. There is room for development of the testing process, only if it is to guide test practices toward better efficiency and effectiveness.

First, we start with component testing and it is the lowest test level according to V-Model.

In the fifth model, each verification stage has a corresponding stage in the verification stage.

During the design phase of the unit. This test is performed to eliminate errors at the symbol or unit level. The unit is the smallest entity that can exist independently. Unit testing verifies that the smallest entity can operate correctly.

6.1 Black-box

Table 2 Customer GUI Testing

condition	Valid partition	Invalid partition	Valid boundaries	invalid boundaries
Username	4-32 char	<4 char	4 char	2 char
	Valid char	>32 char	32 char	33 char
Password	8-32 digit	<8 digit	8 digits	7 digits
		>32 digit	32 digit	33 digit
Email	12 – 40	<12 char	12 char	11 char
	Valid char	>40 char	40 char	41 char
Phone Number	10-digit integer number	<10 digit	10 digits	9 digits
	Start with 079, 078 or 077	>10 digit		11 digits

Table 3 Store Admin GUI Testing

condition	Valid partition	Invalid partition	Valid boundaries	invalid boundaries
Username	4-32 char	<4 char	4 char	2 char
	Valid char	>32 char	32 char	33 char
Password	8-32 digit	<8 digit	8 digits	7 digits
		>32 digit	32 digit	33 digit
Email	12 – 40	<12 char	12 char	11 char
	Valid char	>40 char	40 char	41 char
Phone Number	10-digit integer number	<10 digit	10 digits	9 digits
	Start with 079, 078 or 077	>10 digit		11 digits
Address	Lees than 50 char	>50 char	<50char	>50 char
Country	Lees than 50 char	>50 char	<50char	>50 char

Table 4 Offer GUI Testing

condition	Valid partition	Invalid partition	Valid boundaries	invalid boundaries
Name	Lees than 50 char	>50 char	<50char	>50 char
Number Of Copy	Positive Integer	<0	>0	Negative Integer
Price	Positive Integer	<0	>0	Negative Integer

Table 5 Product GUI Testing

condition	Valid partition	Invalid partition	Valid boundaries	invalid boundaries
Name	Lees than 50 char	>50 char	<50char	>50 char
Number Of Copy	Positive Integer	<0	>0	Negative Integer
Price	Positive Integer	<0	>0	Negative Integer
Product Section	Lees than 50 char	>50 char	<50char	>50 char
Par Code	Lees than 50 char	>50 char	<50char	>50 char
Description	Lees than 500 char	>500 char	<500char	>500 char
Manufacturer	Lees than 50 char	>50 char	<50char	>50 char
Discount	Positive Integer	<0	>0	Negative Integer

Table 6 Product GUI Testing

condition	Valid partition	Invalid partition	Valid boundaries	invalid boundaries
Name	Lees than 50 char	>50 char	<50char	>50 char
Phone Number	10-digit integer number	<10 digit	10 digits	9 digits
Type	Lees than 50 char	>50 char	<50char	>50 char
Description	Lees than 500 char	>500 char	<500char	>500 char

6.1.1 *Account Testing component.*

Table 6-7 Decision Table of Account Testing component.

	R1	R2	R3	R4	R5
Valid Login	**T**	**T**	**T**	**T**	**F**
search Account	**T**	**T**	**F**	**F**	**-**
View Account	**T**	**F**	**T**	**F**	**-**
Successful login	**T**	**T**	**T**	**T**	**F**
Add Account	**F**	**T**	**X**	**T**	**F**
Modify Account	**T**	**F**	**X**	**F**	**F**

6.1.2 *Manage Advertisement Testing Component.*

Table 6-8 Decision Table of Manage Advertisement Testing Component.

	R1	R2	R3	R4
Valid Login	**T**	**T**	**F**	**F**
Click Advertisement	**T**	**F**	**T**	**F**
Successful login	**T**	**T**	**F**	**F**
Add Advertisement	**-**	**T**	**F**	**F**
Delete Advertisement	**-**	**T**	**F**	**F**
View Advertisement	**T**	**-**	**T**	**F**

5. *Manage Offer GUI Testing Component*

Table 6-9 Decision Table of Manage Offer GUI Testing Component

	R1	R2	R3	R4
Valid Login	**T**	**T**	**F**	**F**
Click On Offer	**T**	**F**	**T**	**F**
Successful login	**T**	**T**	**F**	**F**
View Offer	**T**	**F**	**T**	**F**
Add Offer	**T**	**T**	**F**	**F**
Edit Offer	**T**	**T**	**F**	**F**
Delete Offer	**T**	**T**	**F**	**F**

6.1.3 *Manage Product GUI Testing Component*

Table 6-10 Decision Table of Manage Product GUI Testing Component

	R1	R2	R3	R4
Valid Login	**T**	**T**	**F**	**F**
Click On Product	**T**	**F**	**T**	**F**
Successful login	**T**	**T**	**F**	**F**
View Product	**T**	**F**	**T**	**F**
Add Product	**T**	**T**	**F**	**F**
Edit Product	**T**	**T**	**F**	**F**
Delete Product	**T**	**T**	**F**	**F**

6.2 White-box

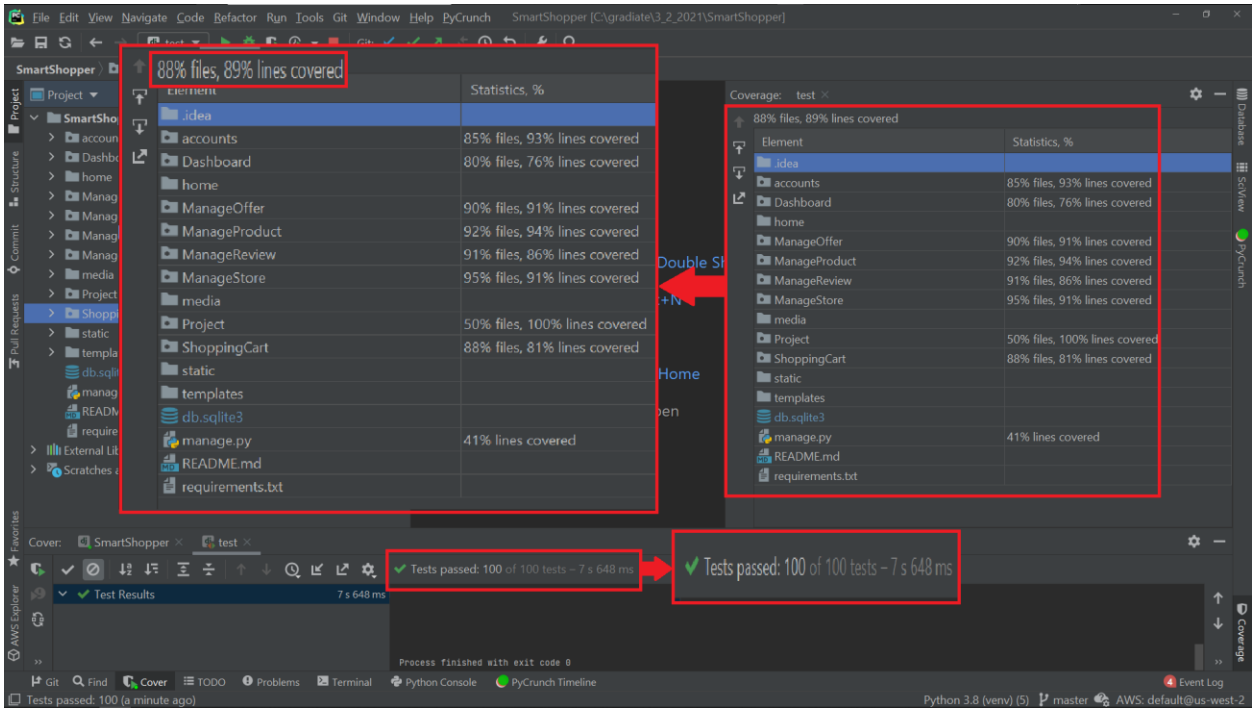
In white testing we use (Django test) and (unit test) tool to test:

1. Code coverage.
2. Line coverage.

Unit Testing:

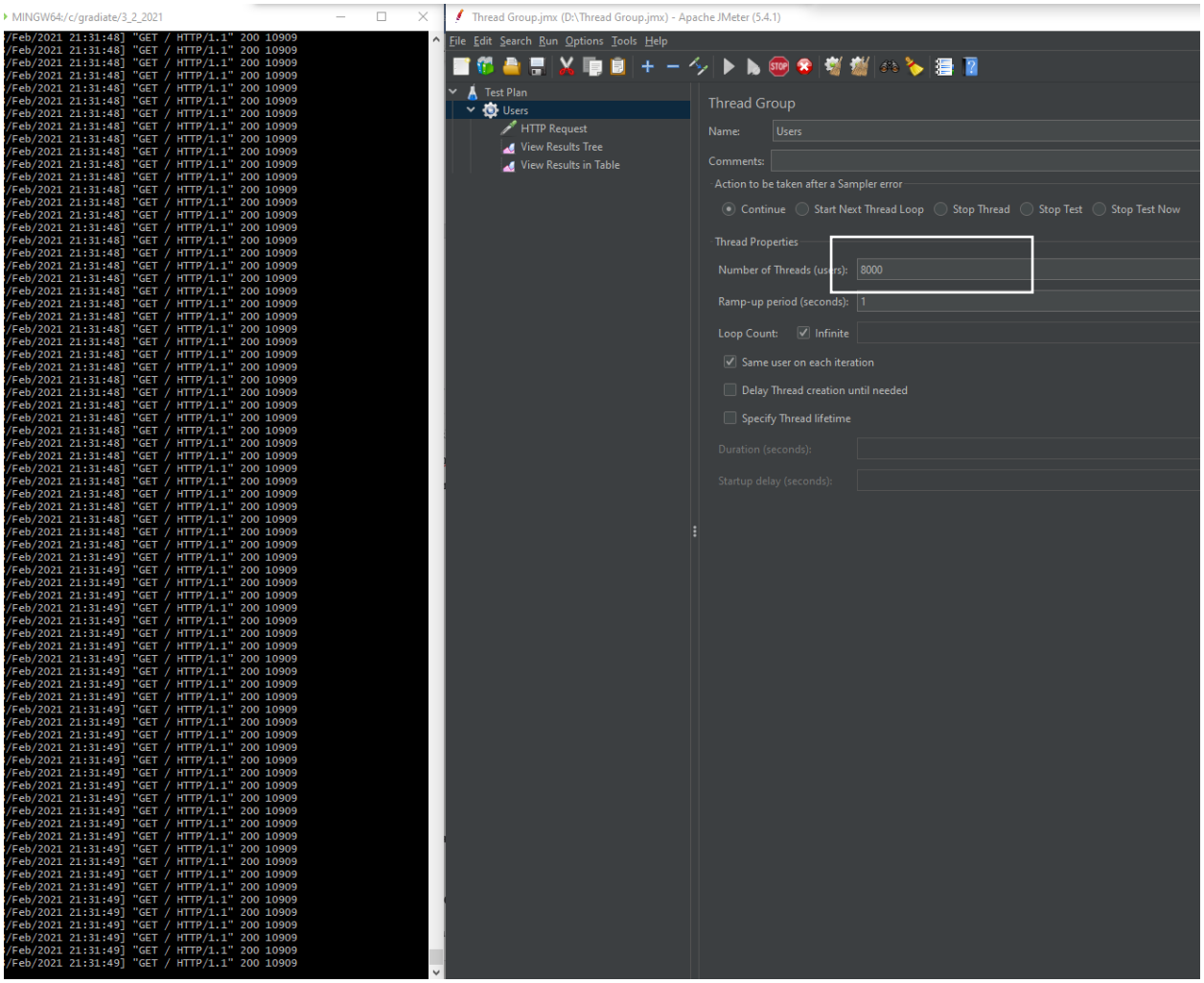
We are use testing library from (django.test and unittest), first we test every url and every variable that should pass to view, after done from urls we start test the view if return the correct data from models and send it to the template, then start test models by check the constraint for each field and each function in class model by pass valid data and in valid data to models, finally test forms that that the user fill the data in to check if the constraint don't broken.

We are use plugin testing tool like (PyCrunch - Live Testing) to automatically runs impacted tests on code change, shows coverage inline with my code and see which lines are hit by each tests to make sour we don't messing any view, url, models or forms, and to make sour we don't have unreachable codes.



6.2Testing automation

1. The automation tools that have been used to control the execution of tests and the comparison of actual outcomes with predicted outcomes.
    1. The JMeter was used to check the bearing capacity of the application.
    2. (django.test ,unittest, PyCrunch - Live Testing) was used to obtain the maximum possible branch test, statement coverage, resolution coverage, and path test.



6.3 Integration Testing Plan

Incremental Integration Plan for Presentation Layer:

Baseline 0: Admin GUI.

Baseline 1: Customer GUI + Admin GUI.

Incremental Integration Plan for Application Layer:

Baseline 0: Model.

Baseline 1: Manage Accounts + Model.

Baseline 2: Manage Store + Manage Accounts + Model.

Baseline 3: Manage Product + Manage Store + Manage Accounts + Model.

Baseline 4 : Manage Offer + Manage Product + Manage Store + Manage Accounts + Model.

Baseline 5: Manage Review + Manage Offer + Manage Product + Manage Store + Manage Accounts + Model.

Baseline 6: Dashboard + Manage Review + Manage Offer + Manage Product + Manage Store + Manage Accounts + Model.

Baseline 7: Shopping Cart + Dashboard + Manage Review + Manage Offer + Manage Product + Manage Store + Manage Accounts + Model.

Incremental Integration Plan for All Layer:

Baseline 0: Data Layer.

Baseline 1: Application Layer + Data Layer.

Baseline 2: Presentation Layer + Application Layer + Data Layer.

CHAPTER 7 : Conclusion and Results

The idea of this system revolves around buying products through the Internet to get rid of the problems users face in the traditional system from wasting time and effort and buying products of lower quality than they expect. We aim through the system to work to provide users with the best products and the best offers available in the market by specifying the user's priorities, conditions, distance, stores, and conditions that must be taken to determine the best products appropriate for him.

Benefit: Solving some problems in some systems and providing features that were not present in those systems and through research we found that some systems did not provide complete comfort to the user, in our system we provided all the features that provide comfort and ease and how to deal with them and with the current difficult conditions, our system is huh The solution

CHAPTER 8 : References

[1] Google shopper,
from : http://www.google.com/smartShopper,
Accessed June 2020

[2]Groupon,
from : http://www.groupon.com,
Accessed June 2020

[3] PriceGrabber
from : http://www.pricegrabber.com/
Accessed June 2020

[4] Target
from : https://www.target.com/
Accessed June 2020

[5] 3-Tier Architecture: A Complete Overview, from : https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/ Accessed June 2020