A PROJECT ON

Fashion 24 x 7
(Online Clothes Shopping Website)

SUBMITTED FOR THE COURSE
POST GRADUATE DIPLOMA IN ADVANCED COMPUTING (PG – DAC)

INSTITUTE OF ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT

Akurdi, Pune - 411044

SUBMITTED BY :

Somesh Dattatraya Dahotre
Hanumant Rangnath Bhite

UNDER THE GUIDANCE OF :

Mrs. Sonali Mogal
Faculty Member : IACSD, Akurdi, Pune

SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)
FOR
Fashion 24 x 7 (Online Clothes Shopping Website)

Authors:-

1. Mr. Hanumant Rangnath Bhite (230341220046)
2. Mr. Somesh Dattatraya Dahotre (230341220191)

Customer:-

Mr. XYZ

Guidance By:-

Mrs. Sonali Mogal

1.1 PURPOSE

Defining and describing the functions and specifications of the **Fashion 24 x 7 (Online Clothes Shopping Website)** is the primary goal of this Software Requirements Specification (SRS). This Software Requirements Specification describes the uses and required functionalities provided by the application.

The basic terminologies used here in this document are defined in the section of acronyms with the purpose of avoiding misconceptions & ambiguity. This document is useful to developers, testers, UI designers and product owner for referencing the functionalities throughout the development of the web application.

1.2 SCOPE

Fashion 24 x 7 (Online Clothes Shopping Website) will allow any user to create an account to become a customer. The application will allow users to browse, search, select, and add clothes products to a cart. Then, after selecting products in their cart, they can pay for the order items as per the order details. The above specified web application also allows an admin to manage the stock with full create, retrieve, update and delete (CRUD) functionality.

The application can by deployed by the owner provided with the specified requirements of server, virtual space & database versions. Also, it can be accessed by the normal users simply with the help of an updated Google Chrome Browser.

1.3 DEFINITIONS / ACRONYMS / ABBREVIATIONS

Fashion 24 x 7 (Online Clothes Shopping Website) involves following abbreviations:-

Sr. No.	Abbreviation	Definition
1	SRS	Software Requirement Specification
2	CRUD	Create Read Update Delete
3	POJO	Plain Old Java Object
4	DB	Database
5	MYSQL	My Structured Query Language
6	E-R	Entity Relationship

1.4 REFERENCES FOR REQUIREMENT ANALYSIS & DESIGN

Following references are considered throughout the development of Fashion 24 x 7 (Online Clothes Shopping Website) :-

- Google for problem solving
- https://www.tutorialspoint.com/java/
- http://www.javatpoint.com/java-tutorial
- https://docs.oracle.com/javase/tutorial/
- Effective Java By Joshua Bloch
- http://www.tutorialspoint.com/mysql/

1.5 OVERVIEW OF THE PROJECT

Nowadays, the excitement of the customers for clothes shopping is getting crushed by crowdie malls, the long lines involved in the manual process of payments & what not. This is why many customers are opting to enjoy shopping from home instead of shopping at malls, shops etc.

This document includes Software Requirements Specification which is built to describe the agreement between the customer and the developer regarding the specification of the software requested for 'Online **Clothes Shopping Website** '. Its primary purpose is to provide a clear and descriptive '**Statement of User Requirements**' that can be used as a reference in further development of the software system. This document is broken into a number of sections used to logically separate the whole content for the ease of reference. This Software Requirements Specification aims

to describe the Assumptions, Constraints, Scope of Software to be developed, Functional Requirements, Non-Functional Requirements, various diagrams used while software development related to software described throughout the rest of the document.

This SRS describes, in clear terms, the software's primary uses and required functionality needed to general customer. This project of developing a hazel – free online portal for clothes shopping is providing separate portals for both customers & managers. Also, this online website aims towards reserving the rights to add & update products only to managers.

2. REQUIREMENTS

2.1. FUNCTIONAL REQUIREMENTS

Fashion 24 x 7 (Online Clothes Shopping Website) will provide the following functionalities:-

- List Of Functions :-

Sr. No.	Function ID	Name Of The Function
1	F - 01	Collection of Clothes (as a product)
2	F - 02	User's Records
3	F - 03	List Of Available Products
4	F - 04	Log In – Log Out
5	F - 05	Shopping Cart
6	F - 06	Sorting Criteria
7	F – 07	Reviews & Ratings
8	F - 08	Admin Role
9	F - 09	Checkout
10	F - 10	Account Recovery
11	F - 11	Concurrency (Multi Threading)

- Functional Description :-

i. Function ID : F – 01

a. Purposes : i. To maintain data associated with the inventory,
: ii. A product (clothes) has a title, brand name, size, quantities as per sizes, availability, category, description, price etc.
: iii. The inventory also keeps track of the stock / quantity of each cloth product.

b. Entities Involved : Product (POJO)
c. Organizational Unit :
d. Frequency of Use : High

ii. Function ID : F – 02

a. Purposes : i. To maintain records for many users,
: ii. An user can be either a member or non-

member,
: iii. An user has a username (unique across all users), email address, password (with proper validations), and address (including detailed address, city, pin code, country etc.) , contact number etc.
: iv. Anyone may sign up for a customer account.

b. Entities Involved : User (POJO)
c. Organizational Unit :
d. Frequency of Use : High

iii. Function ID : F – 03

a. Purposes : i. To show a listing of available products,
: ii. All the clothes products are to be displayed in proper order by title.
: iii. Each cloth will list the following – Title / Brand / Price / Availability

b. Entities Involved : Product (POJO)
c. Organizational Unit :
d. Frequency of Use : High

iv. Function ID : F – 04

a. Purposes : i. To allow users and managers to log in and out of the system,
: ii. Users (Including both user and the manager) will be logged out if inactive for 30 minutes.

b. Entities Involved : User, Manager (POJOs)
c. Organizational Unit :
d. Frequency of Use : High

v. Function ID : F – 05

a. Purposes : i. Any registered user is able to add one or more clothes products to the shopping cart,
: ii. The shopping cart does not need to allow multiple copies of any clothes product.

b. Entities Involved : Cart, Cart-item (POJOs)
c. Organizational Unit :
d. Frequency of Use : Medium

vi. Function ID : F – 06

a. Purposes : i. The website must provide consistent sorting options including following criteria - Sort By Price / View New Products / Sort By Discounts / Sort By Category
b. Entities Involved : Product, Category (POJOs)
c. Organizational Unit :
d. Frequency of Use : Medium

vii. Function ID : F – 07

a. Purposes : i. This web application must give options to all the registered users to give reviews & ratings for all the products
b. Entities Involved : Feedback (POJO)
c. Organizational Unit :
d. Frequency of Use : Medium

viii. Function ID : F – 08

a. Purposes : i. Registered Manager should be able to view and add new products,
: ii. Also, they should be able to view and update user's details.
b. Entities Involved : Manager (POJO)
c. Organizational Unit :
d. Frequency of Use : High

ix. Function ID : F – 09

a. Purposes : i. Checkout is only available to logged-in users / managers,
: ii. When a user enters incorrect credentials, will be given a chance to log in.
b. Entities Involved : User, Manager (POJOs)
c. Organizational Unit :
d. Frequency of Use : Medium

x. Function ID : F – 10

a. Purposes : i. This online website should give facility to recover account in case of ' Forgot Password Scenario '.
b. Entities Involved : User, Manager (POJOs)
c. Organizational Unit :

d. Frequency of Use : Low

xi. Function ID : F – 11

a. Purposes : i. The online clothes shopping system would need to be highly concurrent. There will be multiple ordering requests for clothes shopping at particular point in time. The service should handle this type of load gracefully and fairly.

b. Entities Involved : - - -

c. Organizational Unit :

d. Frequency of Use : - - -

2.2. NON - FUNCTIONAL REQUIREMENTS

Fashion 24 x 7 (Online Clothes Shopping Website) will provide the following non – functional requirements:-

i. Interface :-

~ Fashion 24 x 7 (Online Clothes Shopping Website) must provide user interactive interface in order to attract more users.
~ The application should use best available attractive colour shade combinations.

ii. Performance:-

~ Number of Concurrent Users: - The application must handle maximum number of requests.

iii. Security:-

~ The online clothes shopping website must provide maximum level of security regarding data.
~ The data of the users, product details, valuable feedbacks, login credentials must be protected in order to maintain high customer satisfaction.
~ The application must provide separation via Authorization & Authentication.

iv. Availability:-

~ Fashion 24 x 7 (Online Clothes Shopping Website) must be available 24 X 7 i.e. throughout the day & night, so that users can enjoy shopping all the time.

v. Reliability:-

~ The specified application must be reliable, especially at the time of weekend, festival days, year endings etc.
~ The application must be reliable in the perspective of login / payment failures also.

vi. Safety:-

~ The online application must be saved against session fixations / SQL injection etc. malicious attacks.
~ The whole software must use firewall configurations in order to safeguard the application.

vii. Maintainability:-

~ The Fashion 24 x 7 (Online Clothes Shopping Website) should be able to maintain with as less efforts & changes as possible.

viii. Portability:-

~ The specified application must provide portability in order to change components of architecture in case of emergencies.
~ It should hazel – free facility to replace the databases to enhance the efficiency in needed in future. Like replacement from MYSQL to Oracle or MYSQL to MongoDB.

ix. Accessibility:-

~ The online website must be accessible via desktops, laptops, smart devices including mobile phones, tablets etc.
~ The UI – UX must not hamper in case of above options. It should remain uniform throughout all the devices.

x. Durability:-

~ The overall application should be durable, especially in the terms of data, product availability, and uniform performance over time.

xi. Other Requirements:-

- **Hardware: -** The application is expected to function on Dell G3 – 15 with 1100 MHz Pre Processor Equivalent Or Above, 4 GB RAM, 512 GB HDD.

- **Software: -** The Fashion 24 x 7 (Online Clothes Shopping Website) shall work on Microsoft Windows operating systems family (MS Windows XP & Above). It configures to work with MYSQL database. This System works on Apache Tomcat server. It uses browser Google Chrome Browser.

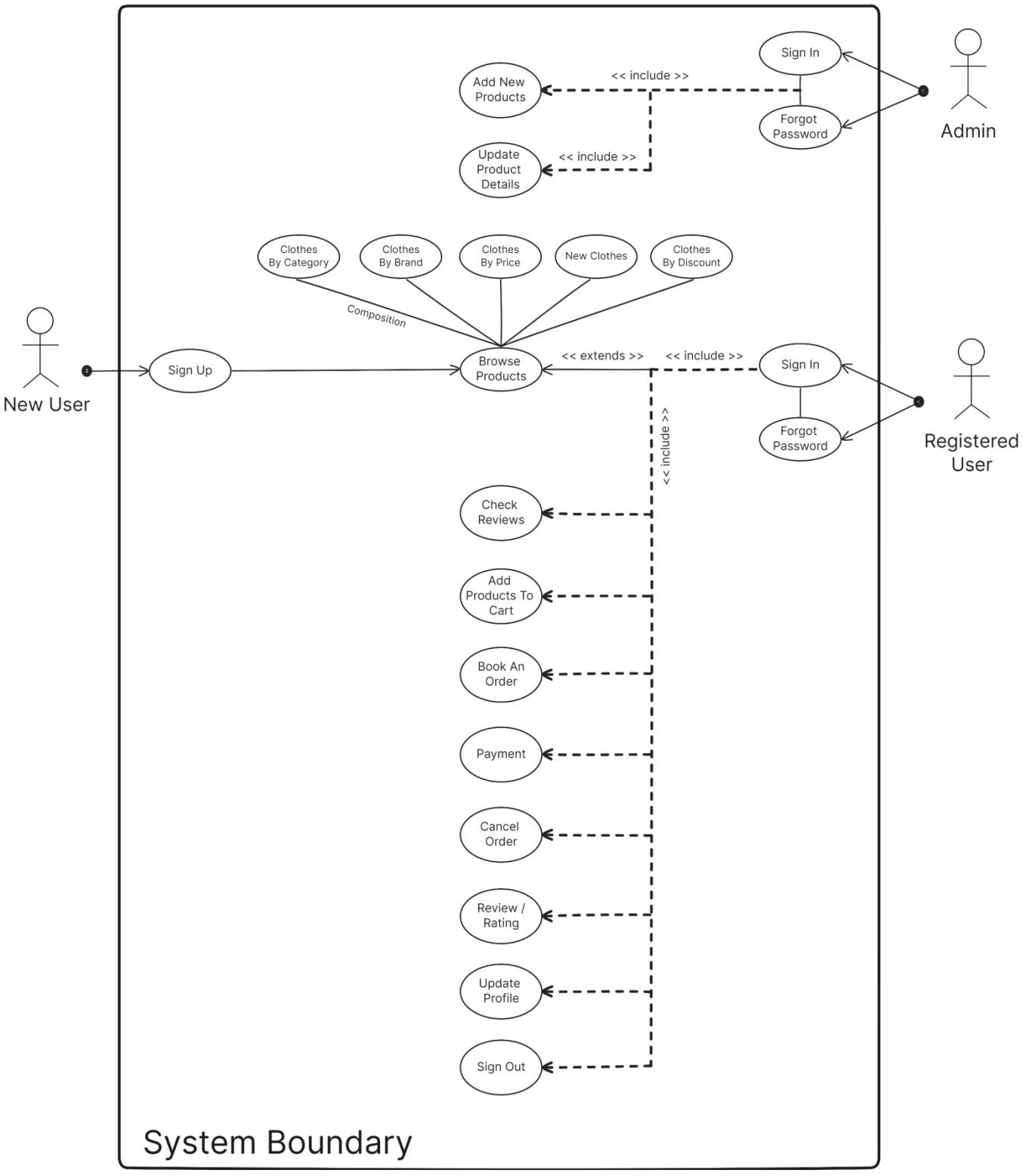
3. DIAGRAMS

3.1. USE CASE DIAGRAM:-

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system / subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

Here are all the basic terms used in the Use Case Diagram:-

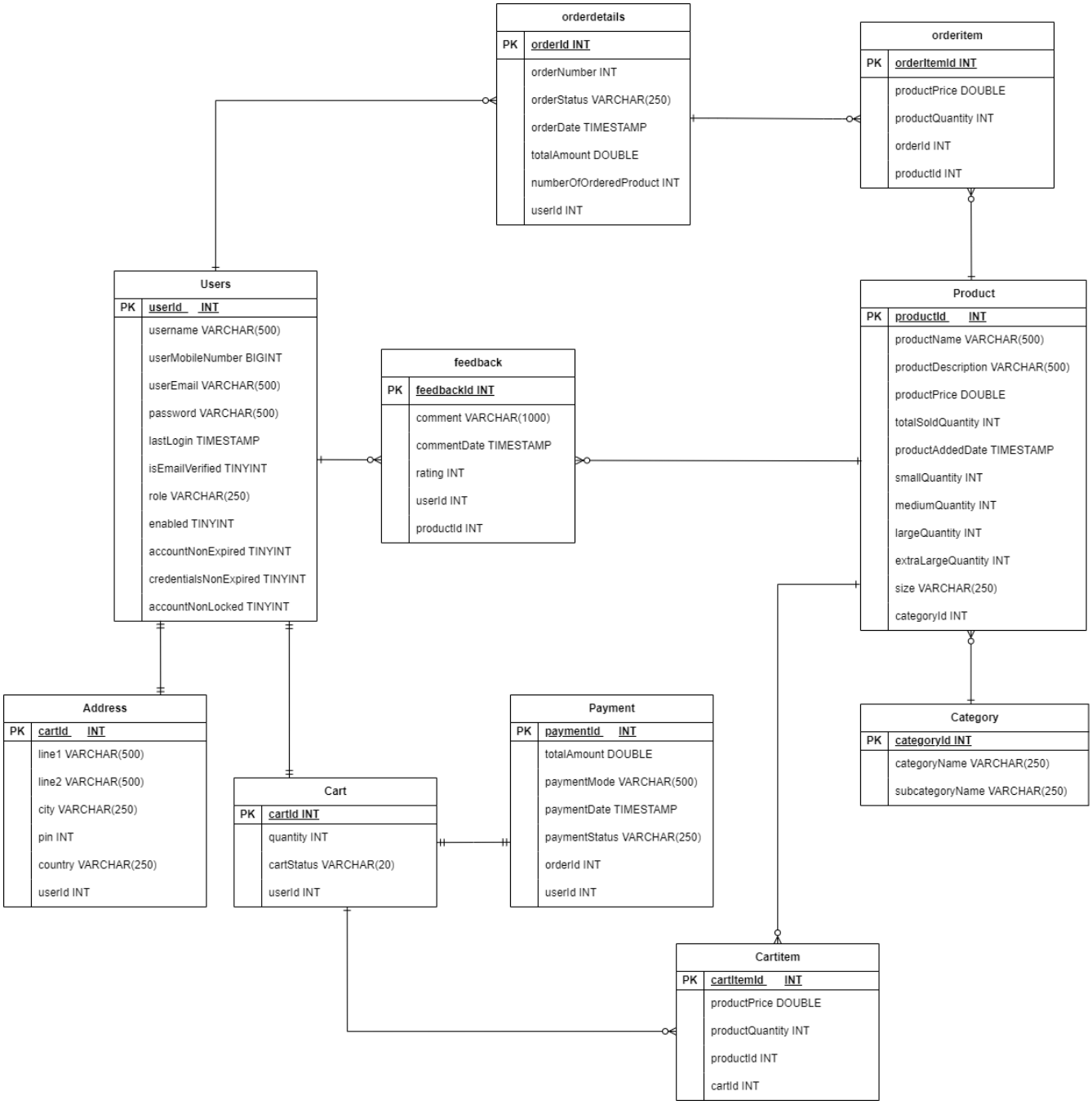
1. Use cases: Horizontally shaped ovals that represent the different uses that a user might have.
2. Actors: Stick figures that represent the people actually employing the use cases.
3. Associations: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases. In this, Usually two keywords are used to denote the tight coupling & loose coupling i.e. include & extends respectively.
3. System boundary boxes: A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system.


Sign In
Add New Products
<< include >>
Forgot Password
Admin
Update Product Details
<< include >>
Clothes By Category
Clothes By Brand
Clothes By Price
New Clothes
Clothes By Discount
Composition
New User
Sign Up
Browse Products
<< extends >>
<< include >>
Sign In
Forgot Password
Registered User
<< include >>
Check Reviews
Add Products To Cart
Book An Order
Payment
Cancel Order
Review / Rating
Update Profile
Sign Out
System Boundary

3.2. E-R DIAGRAM:-

Entity Relationship Diagram is used to define the data elements and relationship for a specified application. It develops a conceptual design for the database. It also develops a very simple and easy to design view of the data.

In Entity Relationship Diagram, the data is represented by using various components including entities, attributes, relationships (One To Many / Many To Many etc.)

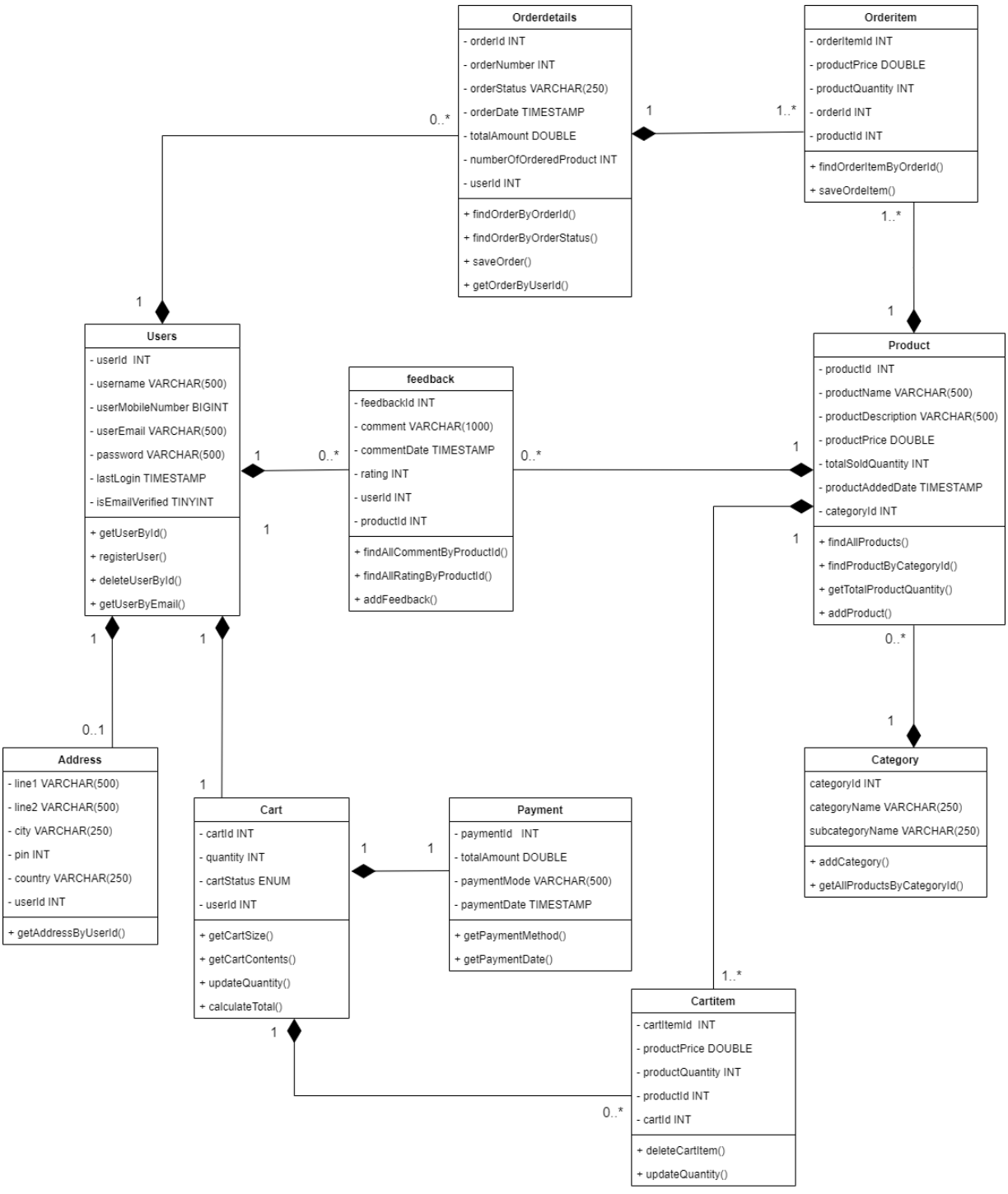


3.3. CLASS DIAGRAM:-

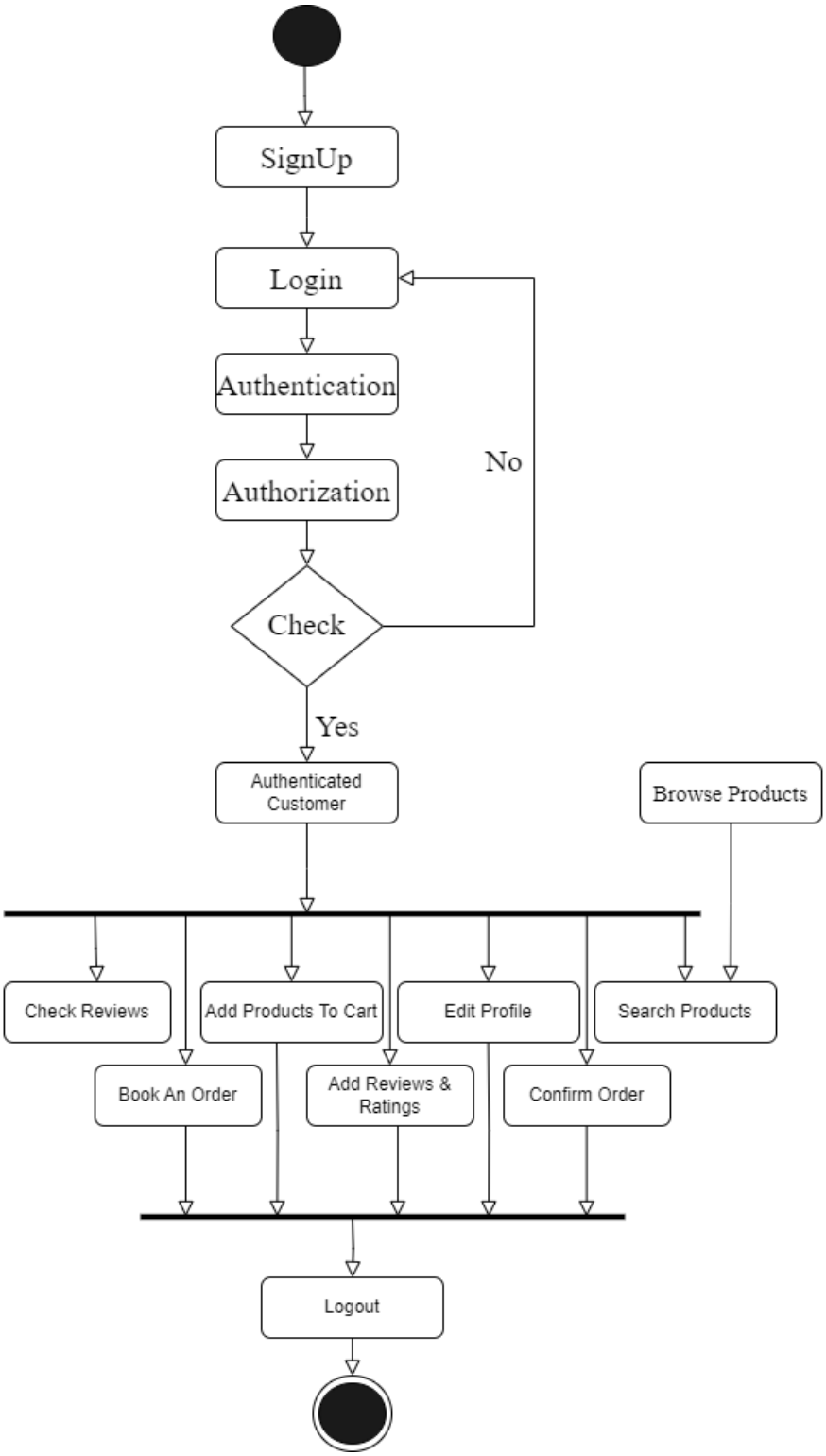
Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram. The purpose of the class diagram can be summarized as:-

A. Analysis and design of the static view of an application.
B. Describe responsibilities of a system.
C. Base for component and deployment diagrams.
D. Forward and reverse engineering.

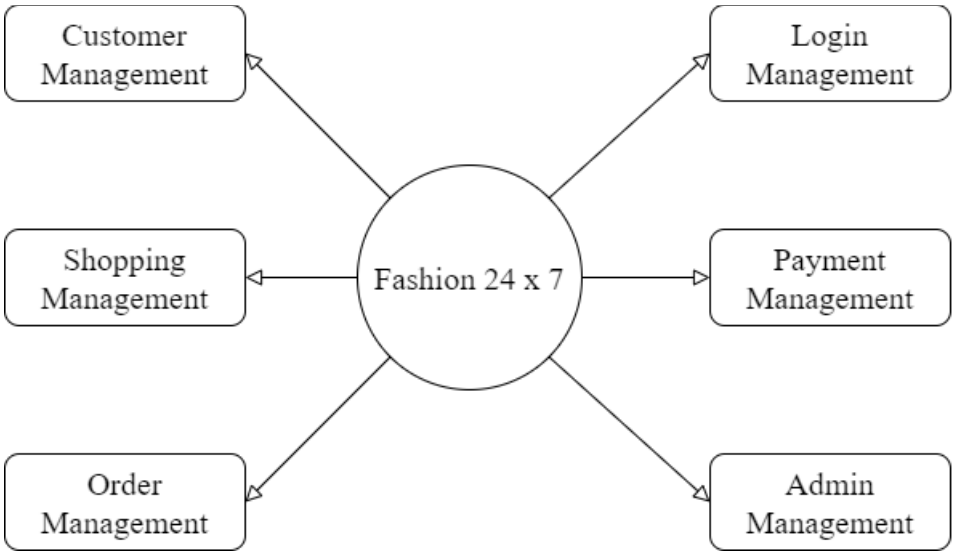

Orderdetails
- orderId INT
- orderNumber INT
- orderStatus VARCHAR(250)
- orderDate TIMESTAMP
- totalAmount DOUBLE
- numberOfOrderedProduct INT
- userId INT
+ findOrderByOrderId()
+ findOrderByOrderStatus()
+ saveOrder()
+ getOrderByUserId()
Orderitem
- orderItemId INT
- productPrice DOUBLE
- productQuantity INT
- orderId INT
- productId INT
+ findOrderItemByOrderId()
+ saveOrdeItem()
Users
- userId INT
- username VARCHAR(500)
- userMobileNumber BIGINT
- userEmail VARCHAR(500)
- password VARCHAR(500)
- lastLogin TIMESTAMP
- isEmailVerified TINYINT
+ getUserById()
+ registerUser()
+ deleteUserById()
+ getUserByEmail()
feedback
- feedbackId INT
- comment VARCHAR(1000)
- commentDate TIMESTAMP
- rating INT
- userId INT
- productId INT
+ findAllCommentByProductId()
+ findAllRatingByProductId()
+ addFeedback()
Product
- productId INT
- productName VARCHAR(500)
- productDescription VARCHAR(500)
- productPrice DOUBLE
- totalSoldQuantity INT
- productAddedDate TIMESTAMP
- categoryId INT
+ findAllProducts()
+ findProductByCategoryId()
+ getTotalProductQuantity()
+ addProduct()
Address
- line1 VARCHAR(500)
- line2 VARCHAR(500)
- city VARCHAR(250)
- pin INT
- country VARCHAR(250)
- userId INT
+ getAddressByUserId()
Cart
- cartId INT
- quantity INT
- cartStatus ENUM
- userId INT
+ getCartSize()
+ getCartContents()
+ updateQuantity()
+ calculateTotal()
Payment
- paymentId INT
- totalAmount DOUBLE
- paymentMode VARCHAR(500)
- paymentDate TIMESTAMP
+ getPaymentMethod()
+ getPaymentDate()
Category
categoryId INT
categoryName VARCHAR(250)
subcategoryName VARCHAR(250)
+ addCategory()
+ getAllProductsByCategoryId()
Cartitem
- cartItemId INT
- productPrice DOUBLE
- productQuantity INT
- productId INT
- cartId INT
+ deleteCartItem()
+ updateQuantity()

3.4. ACTIVITY DIAGRAM:-

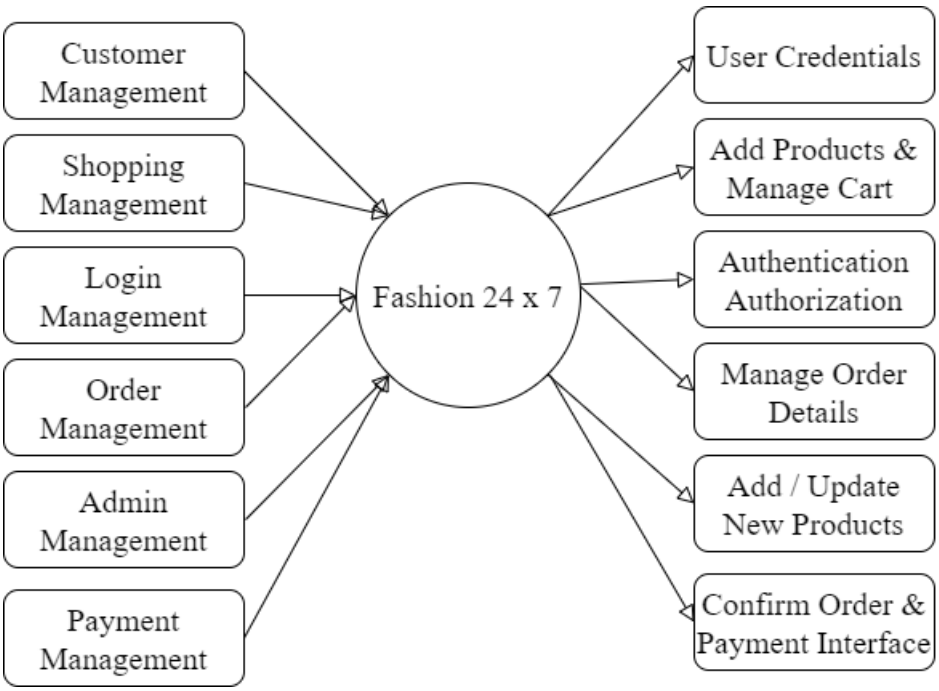


3.5. DATA FLOW DIAGRAMS:-

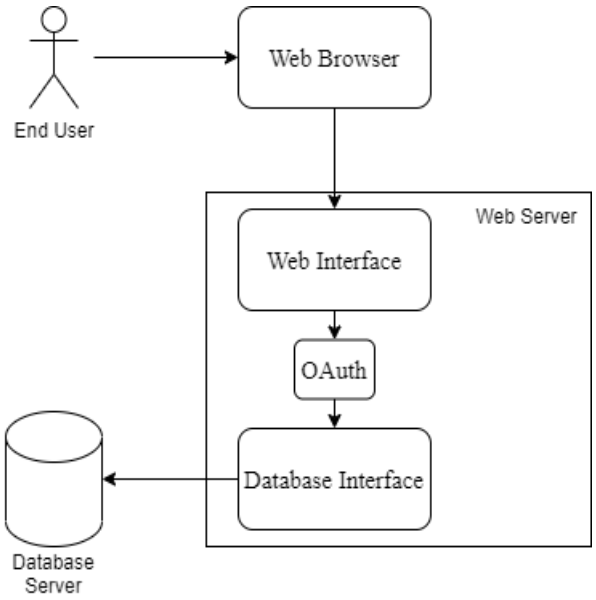
3.5.1. ZERO LEVEL DATA FLOW DIAGRAM:-



3.5.2. FIRST LEVEL DATA FLOW DIAGRAM:-



3.5.DEPLOYMENT DIAGRAM:-



4. TABLE GENERATION

Fashion 24 x 7 (Online Clothes Shopping Website) generate following tables in the database:-

```
root>desc users;
+----------------------+--------------+------+-----+-------------------+-------------------+
| Field                | Type         | Null | Key | Default           | Extra             |
+----------------------+--------------+------+-----+-------------------+-------------------+
| userId               | int          | NO   | PRI | NULL              | auto_increment    |
| username             | varchar(500) | YES  |     | NULL              |                   |
| userMobileNumber     | bigint       | YES  | UNI | NULL              |                   |
| gender               | varchar(45)  | YES  |     | NULL              |                   |
| userEmail            | varchar(500) | YES  | UNI | NULL              |                   |
| password             | varchar(500) | YES  |     | NULL              |                   |
| registeredAt         | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| lastLogin            | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| isEmailVerified      | tinyint      | YES  |     | 0                 |                   |
| role                 | varchar(250) | YES  |     | NULL              |                   |
| enabled              | tinyint      | YES  |     | 1                 |                   |
| accountNonExpired    | tinyint      | YES  |     | 1                 |                   |
| credentialsNonExpired | tinyint     | YES  |     | 1                 |                   |
| accountNonLocked     | tinyint      | YES  |     | 1                 |                   |
+----------------------+--------------+------+-----+-------------------+-------------------+
14 rows in set (0.04 sec)
```

```
root>desc product;
+--------------------+---------------+------+-----+-------------------+-------------------+
| Field              | Type          | Null | Key | Default           | Extra             |
+--------------------+---------------+------+-----+-------------------+-------------------+
| productId          | int           | NO   | PRI | NULL              | auto_increment    |
| productName        | varchar(500)  | YES  |     | NULL              |                   |
| productDescription | varchar(1000) | YES  |     | NULL              |                   |
| productPrice       | double        | YES  |     | NULL              |                   |
| productAddedDate   | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| imageUrl           | varchar(500)  | YES  |     | NULL              |                   |
| categoryId         | int           | YES  | MUL | NULL              |                   |
+--------------------+---------------+------+-----+-------------------+-------------------+
7 rows in set (0.01 sec)
```

```
root>desc address;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| addressId | int          | NO   | PRI | NULL    | auto_increment |
| line1     | varchar(500) | YES  |     | NULL    |                |
| line2     | varchar(500) | YES  |     | NULL    |                |
| city      | varchar(250) | YES  |     | NULL    |                |
| state     | varchar(90)  | YES  |     | NULL    |                |
| pin       | int          | YES  |     | NULL    |                |
| country   | varchar(250) | YES  |     | NULL    |                |
| userId    | int          | YES  | MUL | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

```
root>desc cart;
+---------------+------------+------+-----+-------------------+-------------------+
| Field         | Type       | Null | Key | Default           | Extra             |
+---------------+------------+------+-----+-------------------+-------------------+
| cartId        | int        | NO   | PRI | NULL              | auto_increment    |
| totalQuantity | int        | YES  |     | NULL              |                   |
| totalCartPrice| double     | YES  |     | NULL              |                   |
| cartStatus    | varchar(45)| YES  |     | NULL              |                   |
| creationTime  | timestamp  | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| userId        | int        | YES  | MUL | NULL              |                   |
+---------------+------------+------+-----+-------------------+-------------------+
6 rows in set (0.00 sec)
```

```
root>desc cartitem;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| cartItemId   | int         | NO   | PRI | NULL    | auto_increment |
| unitQuantity | int         | YES  |     | NULL    |                |
| unitPrice    | double      | YES  |     | NULL    |                |
| totalPrice   | double      | YES  |     | NULL    |                |
| productId    | int         | YES  | MUL | NULL    |                |
| cartId       | int         | YES  | MUL | NULL    |                |
| sizeName     | varchar(45) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

```
root>desc category;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| categoryId   | int          | NO   | PRI | NULL    | auto_increment |
| categoryName | varchar(250) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

```
root>desc feedback;
+-------------+---------------+------+-----+-------------------+-------------------+
| Field       | Type          | Null | Key | Default           | Extra             |
+-------------+---------------+------+-----+-------------------+-------------------+
| feedbackId  | int           | NO   | PRI | NULL              | auto_increment    |
| comment     | varchar(1000) | YES  |     | NULL              |                   |
| commentDate | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| rating      | int           | YES  |     | NULL              |                   |
| userId      | int           | YES  | MUL | NULL              |                   |
| productId   | int           | YES  | MUL | NULL              |                   |
+-------------+---------------+------+-----+-------------------+-------------------+
6 rows in set (0.00 sec)
```

```
root>desc orderdetails;
+-----------------------+---------------+------+-----+-------------------+-------------------+
| Field                 | Type          | Null | Key | Default           | Extra             |
+-----------------------+---------------+------+-----+-------------------+-------------------+
| orderId               | int           | NO   | PRI | NULL              | auto_increment    |
| orderNumber           | int           | YES  | UNI | NULL              |                   |
| orderStatus           | varchar(250)  | YES  |     | NULL              |                   |
| orderDate             | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| totalAmount           | decimal(10,2) | YES  |     | NULL              |                   |
| numberOfOrderedProduct | int           | YES  |     | NULL              |                   |
| userId                | int           | YES  | MUL | NULL              |                   |
+-----------------------+---------------+------+-----+-------------------+-------------------+
7 rows in set (0.00 sec)
```

```
root>desc orderitem;
+-----------------+--------+------+-----+---------+----------------+
| Field           | Type   | Null | Key | Default | Extra          |
+-----------------+--------+------+-----+---------+----------------+
| orderItemId     | int    | NO   | PRI | NULL    | auto_increment |
| productPrice    | double | YES  |     | NULL    |                |
| productQuantity | int    | YES  |     | NULL    |                |
| orderId         | int    | YES  | MUL | NULL    |                |
| productId       | int    | YES  | MUL | NULL    |                |
+-----------------+--------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
root>desc payment;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| paymentId     | int          | NO   | PRI | NULL    | auto_increment |
| totalAmount   | double       | YES  |     | NULL    |                |
| paymentMode   | varchar(500) | YES  |     | NULL    |                |
| paymentDate   | timestamp    | YES  |     | NULL    |                |
| paymentStatus | varchar(250) | YES  |     | NULL    |                |
| orderId       | int          | YES  | UNI | NULL    |                |
| userId        | int          | YES  | MUL | NULL    |                |
| cartId        | int          | YES  | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

```
root>desc oauth_client_details;
+-------------------------+--------------+------+-----+---------+-------+
| Field                   | Type         | Null | Key | Default | Extra |
+-------------------------+--------------+------+-----+---------+-------+
| client_id               | varchar(255) | NO   | PRI | NULL    |       |
| resource_ids            | varchar(255) | YES  |     | NULL    |       |
| client_secret           | varchar(255) | YES  |     | NULL    |       |
| scope                   | varchar(255) | YES  |     | NULL    |       |
| authorized_grant_types  | varchar(255) | YES  |     | NULL    |       |
| web_server_redirect_uri | varchar(255) | YES  |     | NULL    |       |
| authorities             | varchar(255) | YES  |     | NULL    |       |
| access_token_validity   | int          | YES  |     | NULL    |       |
| refresh_token_validity  | int          | YES  |     | NULL    |       |
| additional_information  | varchar(4096)| YES  |     | NULL    |       |
| autoapprove             | varchar(255) | YES  |     | NULL    |       |
+-------------------------+--------------+------+-----+---------+-------+
11 rows in set (0.00 sec)
```

```
root>desc oauth_client_token;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| token_id           | varchar(255) | YES  |     | NULL    |       |
| token              | mediumblob   | YES  |     | NULL    |       |
| authentication_id  | varchar(255) | NO   | PRI | NULL    |       |
| user_name          | varchar(255) | YES  |     | NULL    |       |
| client_id          | varchar(255) | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
root>desc oauth_code;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| code           | varchar(255) | YES  |     | NULL    |       |
| authentication | mediumblob   | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

```
root>desc oauth_refresh_token;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| token_id       | varchar(256) | YES  |     | NULL    |       |
| token          | mediumblob   | YES  |     | NULL    |       |
| authentication | mediumblob   | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
root>desc oauth_access_token;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| token_id           | varchar(255) | YES  |     | NULL    |       |
| token              | mediumblob   | YES  |     | NULL    |       |
| authentication_id  | varchar(255) | NO   | PRI | NULL    |       |
| user_name          | varchar(255) | YES  |     | NULL    |       |
| client_id          | varchar(255) | YES  |     | NULL    |       |
| authentication     | mediumblob   | YES  |     | NULL    |       |
| refresh_token      | varchar(255) | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
root>desc oauth_approvals;
+---------------+-------------+------+-----+-------------------+-----------------------------------------------+
| Field         | Type        | Null | Key | Default           | Extra                                         |
+---------------+-------------+------+-----+-------------------+-----------------------------------------------+
| userId        | varchar(255) | YES  |     | NULL              |                                               |
| clientId      | varchar(255) | YES  |     | NULL              |                                               |
| scope         | varchar(255) | YES  |     | NULL              |                                               |
| status        | varchar(10)  | YES  |     | NULL              |                                               |
| expiresAt     | timestamp    | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
| lastModifiedAt | timestamp   | NO   |     | NULL              |                                               |
+---------------+-------------+------+-----+-------------------+-----------------------------------------------+
6 rows in set (0.00 sec)
```

```
root>desc oauth_approvals;
+---------------+-------------+------+-----+-------------------+-----------------------------------------------+
| Field         | Type        | Null | Key | Default           | Extra                                         |
+---------------+-------------+------+-----+-------------------+-----------------------------------------------+
| userId        | varchar(255) | YES  |     | NULL              |                                               |
| clientId      | varchar(255) | YES  |     | NULL              |                                               |
| scope         | varchar(255) | YES  |     | NULL              |                                               |
| status        | varchar(10)  | YES  |     | NULL              |                                               |
| expiresAt     | timestamp    | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED on update CURRENT_TIMESTAMP |
| lastModifiedAt | timestamp   | NO   |     | NULL              |                                               |
+---------------+-------------+------+-----+-------------------+-----------------------------------------------+
6 rows in set (0.00 sec)
```

CLOSURE REPORT

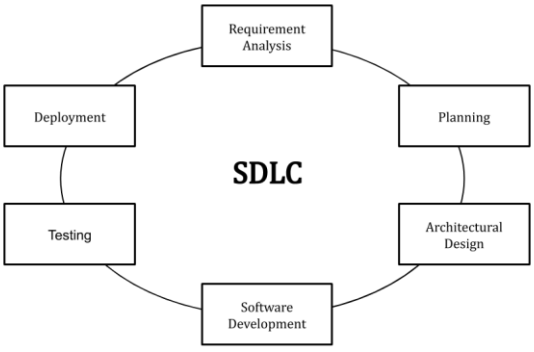
1. GENERAL INFORMATION

The general information about the Fashion 24 x 7 (Online Clothes Shopping Website) is as follows:-

1.1. PRODUCTIVITY:-

- As per the requirements gathered from the customer during initial phase of the software development, the functionalities like CRUD operations for online clothes shopping scenario has been added successfully. That is a registered customer can add, remove clothes product into & from the cart.
- Also, admin can add new products as well as update existing products. This functionality of the proposed website can be used effectively to apply discounts on the products.
- The new i.e. unregistered customers may browse clothes products even without official registration. This functionality is useful to achieve non functional requirement of the website, which is to provide hassle-free use of the website as far as customers are concerned.
- This proposed website for online clothes shopping can be accessed by any latest version of Google Chrome through smart devices as well as personal computers, laptops.
- Spring Security concept is used effectively to secure data by including 'Authentication, Authorization '.

1.2. PROCESS USED & DEVIATION:-

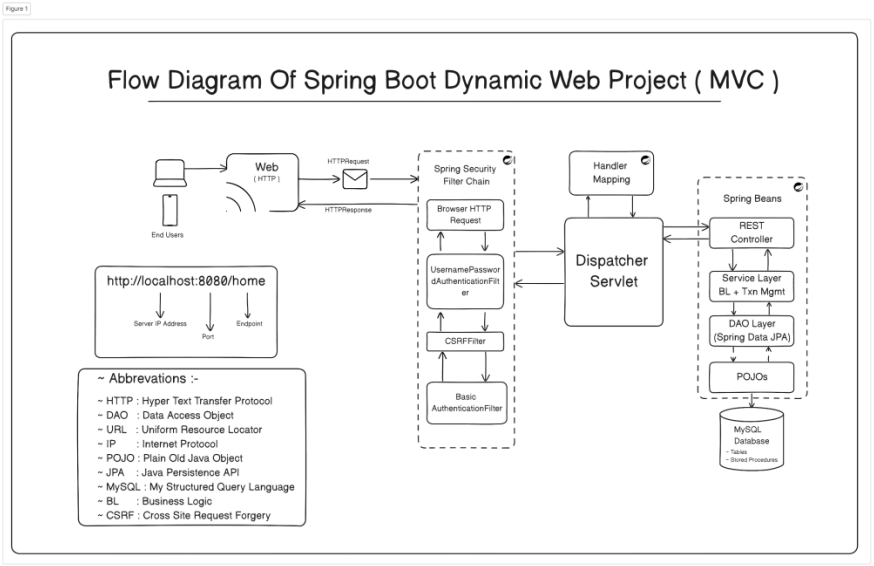


Fashion 24 x 7 (Online Clothes Shopping Website) has been successfully implemented the SDLC Lifecycle in which it went through different phases, which are described as follows:-

1. Requirement Analysis: - Basic requirements were gathered from the

customer and analyzed in a proper way so as to set aim, objectives, and time constraints of the proposed project.

2. Planning:- To meet the specified requirements given by the customer, planning phase is conducted in which the various technologies were taken into consideration for the online clothes shopping website project.
3. Architectural Design: - In this phase, basic flow of the whole project was finalized. Also, included the necessary security measures in the architectural design of the project. The basic lifecycle of a request sent by the client from browser to the response generated by the server is as follows:-



4. Software Development: - In this phase of software development, actual code was written in selected programming languages and finalized frameworks. Fashion 24 x 7 (Online Clothes Shopping Website) has been successfully implemented by using Spring Boot Dynamic Web Project (MVC) and ReactJS as a front end of the project.
5. Testing: - Agile Methodology is used for increasing efficiency of the proposed project. In which, testing is done in parallel with software development. Also, the concept of 'V-Model' is used here effectively, which involves Verification conducted by the developer itself so as to mitigate the defects at earlier stages of the application development.
6. Deployment: - The completed project has been deployed on the EC2 Instance hosted by Amazon Web Services (AWS).

- During the whole process of execution, as some of the requirements were added by the customer in later phases of the Software Development Lifecycle. These changes were effectively considered and revised the structure of the proposed project by adding necessary functionalities. These additions improvised the SRS as per the specified new added requirements.

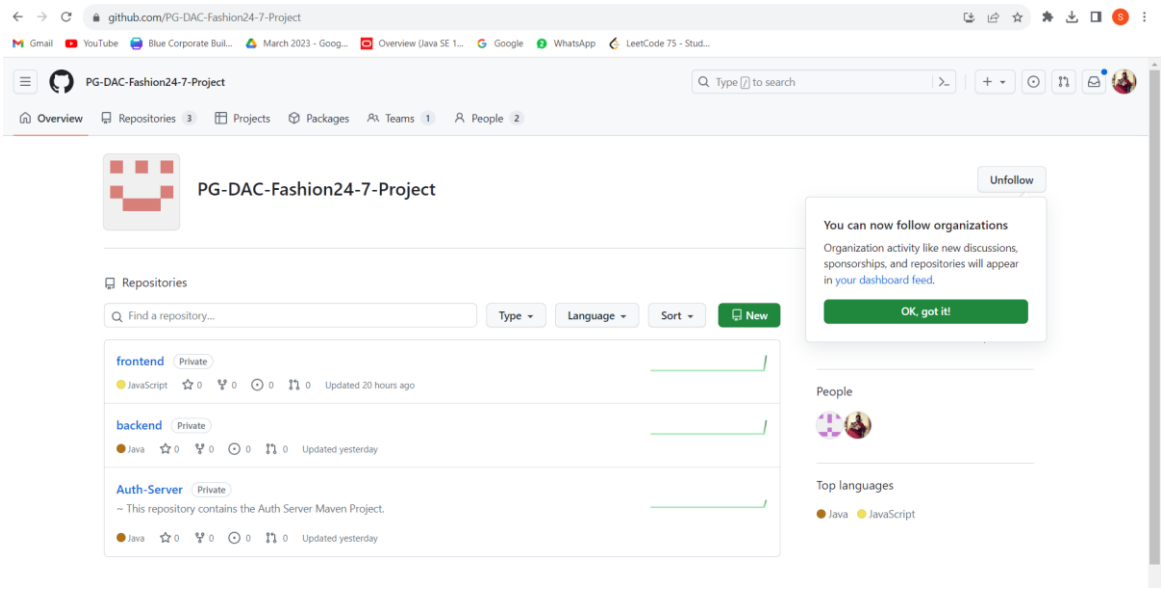
1.3. ESTIMATED & ACTUAL START – END DATES OF THE PROJECT:-

- Estimated Dates:-
  - Start of the Software Development: - 1st August, 2023
  - End of the Software Development: - 25st August, 2023
- Actual Dates:-
  - Start of the Software Development: - 11th August, 2023
  - End of the Software Development: - 31st August, 2023

1.4. TOOLS USED:-

Following are the tools used effectively during the software development of Fashion 24 x 7 (Online Clothes Shopping Website):-

- GitHub: - This tool is used to control versions of the application throughout the development phase. In which separate repositories were created so as to maintain modularity throughout the project.



- Spring Tool Suite (STS):- This tool contains the necessary JVM to compile the written code throughout the Software Development Phase of SDLC. Basic

Debugger also used to detect errors in codes by the developer. Version: 3.9.18 RELEASE, Build Id: 2021091440905, Platform: Eclipse 2021-09 (4.21.0) is used in the project.

- Visual Studio Code: - Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE. This tool provided the necessary GUI for developing front end code in ReactJS.

2. RISK MANAGEMENT

2.1. RISK IDENTIFIED AT START OF THE PROJECT:-

Following are the some of the risks encountered at the start of the software development of Fashion 24 x 7 (Online Clothes Shopping Website):-

- Hosting Platform: - At the start of the proposed project, the platform on which the website is going to be hosted was not finalized. The selection of the hosting platform depends on many parameters which will be available at the completion phase of the online clothes shopping website project. Hence, the decision of selection of the hosting platform was finalized at the later stage of the software development.
- Changes in the database selection: - As the development team decided to use MySQL as a database for this project at start of the project. But as the requirements from the customer were frequently changing, development team was in dilemma to switch database to MongoDB, as it gives hassle-free image handling utility.

2.2. RISK ENCOUNTERED DURING THE PROJECT:-

Fashion 24 x 7 (Online Clothes Shopping Website) encountered following risks during the execution of the project:-

- Security of Data:- As Spring Boot Framework does not support using plain text passwords in the application and forces to encrypt all the passwords used for login to ensure more security of the user credentials. Hence, the development team added the Spring Security Filter Chain to ensure required security of the data and password encoder is used to encrypt the passwords.

2.3. NOTES ON RISK MITIGATION:-

'Risk Mitigation' in software development refers to the process of identifying potentials risks that could impact the success of a software project and taking proactive measures to reduce their likelihood or impact. Following are the basic steps are used to handle risks in software development:-

- Risk Identification
- Risk Assessment
- Risk Mitigation Planning
- Risk Monitoring
- Risk Response Execution
- Documentation

In case of our proposed project of online clothes shopping website, risk of security of data mitigated by using 'DAO Base JWT', in which the encrypted user credentials were persisted in database, MySQL in this case.

3. SIZE

3.1. ESTIMATED AND ACTUAL SIZE (IN KLOC):-

'KLOC' (thousands of lines of code) is a traditional measure of how large a computer program is or how long or how many people it will take to write it. The code measured is usually source code. The KLOC metric is often used when evaluating an application's total number of lines of code (LOC) -- or source lines of code (SLOC).

To find KLOC value, divide total number of lines of code by 1000.

- Estimated Size:-
  - Total Lines of Code = 25000
  - KLOC Value = Total Lines of Code / 1000
    = 25000 / 1000 = 25.00

- Actual Size:-
  - Total Lines of Code = 26156
  - KLOC Value = Total Lines of Code / 1000
    = 26156 / 1000 = 26.15

4. DEFECTS

4.1. SDLC Stage Wise Defects:-

Defects (also known as bugs or issues) can occur at various stages of the Software Development Life Cycle (SDLC). Each stage of the SDLC involves different activities, and defects can arise during any of these activities. Here's a breakdown of the common SDLC stages and the types of defects that can be found at each stage:

- Requirements Gathering and Analysis: Requirements Defects: Incomplete, ambiguous, conflicting, or incorrect requirements can lead to defects down the line.
- System Design: Design Defects: Inadequate or flawed design can result in defects during implementation or testing.
- Implementation (Coding): Coding Defects: Errors introduced during coding, such as syntax errors, logic errors, or incorrect algorithm implementations.
- Security Defects: Vulnerabilities due to improper handling of inputs, lack of validation, or other security-related issues.
- Deployment: Configuration Defects: Issues with the deployment environment's configuration that affect the software's behavior.
- Maintenance and Support: Maintenance Defects: Defects that are identified after the software has been released and is in production use.

4.2. Distribution of Defects:-

Defects classification as major or minor can vary based on project specific criteria and severity levels defined by the development team. Here, in this project, the defects are distributed as follows:-

- Major Defects :
  - Requirements Defects
  - System Design
  - Coding Defects
  - Security Defects
  - Configuration Defects
  - Maintenance Defects
- Minor Defects :
  - Performance Defects