Contents

Chapter 1. Introduction

1.1 Overview of Project

A Sport Portal which will allow formal and informal merchants in developing countries to advertise and sell their goods on internet. This would permit rural communities to make their wares available to the rest of the world via the World Wide Web. The objectives of this project are to create a sport portal with content management system which would allow product information to be updated securely using a mobile device. The web portal will have an online interface in the form of a sport portal that will allow users to buy goods from the merchants.

This project will be divided into following separate components:

· The content management system (CMS)

· The Sport website/portal

· The product, merchant and customer database

· The coupons, discount managements

· Reporting of the sales, orders, shipments etc

· The online transaction security system

· The data security system

1.2 Justification of Platform

1.2.1 Admin module

i. **Welcome and Login Page:** The main page, neatly designed page, here login option is provided to login. Also new signup option is provided to for new user to signup

ii. **Product Category Display Page:** This is search screen for selecting product Option must be provide to select the product based on name. Select product based on the category and based on the company, Etc.

iii. **Product Search Result Screen:** This screen shows all the products under the search criteria. If there are no products under these search criteria then system will give the message that there are no products found under this category. Also, option is provided to select the products.

iv. **View Cart Screen:** View cart screen shows all the selected items, here quantity as to be entered. And also, option is provided to deselect the products.

v. **Order Details Screen: -**This screen shows the final products in the view cart. This gives product list, quantity, Total Amount etc., so user will get the clear information that how much is the total amount of this selection.

vi. **Order Confirmation Screen: -**This is the final screen in buying process. Here customer as confirm the order.

vii. **Customer Registration: -**This module deals with the registration of New Customers I.e. the details of the customer like name, email address, password, etc. When Registration successful and customer is allowed to logged in the website. So that he/she can access most of the features available in the website.

viii. **Product Details: -**This module deals with selecting the products to be purchased and adding to the cart.

ix. **Payment Gateway: -**This module deals with Payment Options in this the Link is be given and then when it clicks the PAYMENT GATEWAY (new window will open and pay) and when its successful pay by customers/end user then BACK or HOME option will be available to Reach Welcome Page.

1.2.2 User module

i. **Security: -**The system must automatically log out all customers after a period of inactivity. The System's back-end servers shall only be accessible to authenticated administrators. Sensitive data will be encrypted before being sent over insecure connections like the internet.

ii. **Reliability: -**The reliability of the overall program depends on the reliability of the separate componet. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes.

iii. **Availability: -** The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, are placement page will be shown. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator.

iv. **Maintainability: -**A commercial database is used for maintaining the database and the application server take care of the site. In case of a failure, a re-initialization of the program will be done. Also, the software design is being done with modularity in mind so that maintainability can be done efficiently.

v. **Portability: -**The application is HTML and scripting language based. So, the end user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user is using this system on any OS; either it is Windows or Linux. The system shall run on PC, Laptops, and PDA etc.

CHAPTER 2. SYSTEM ANALYSIS

2.1 Existing system.

i.   Many customers go for purchasing offline so as to examine the product and hold the possession of the product just after the payment for the product. In this Web portal loyalty depends upon the consistent ability to deliver quality, value and satisfaction. Offline shopping has a sense of immediacy.

ii.   You get to possess the item you've purchased the very moment.

iii. If we can search and make a list of items that we would like to try while shopping, before actually going out. This way we can be more confident in our purchase and not missing out something... This can also help us to decide what areas to visit. And plan such an event with friends.

Limitations of Existing System

- Time Consuming
- Shipping Rates
- Refund/Returns Disputes
- Lack of Options
- Cash Back Offers not present
- Bad customer service

Also, there are expenses for traveling from house to shop. More over the shop from

where we would like to buy something may not be open 24*7. In order to overcome these, we have e-commerce solution, i.e. one place where we can get all required sports goods/products **online**.

2.2 Proposed system.

i. To remove all the disadvantages of conventional methods, a system is proposed which is an **online portal for shopping.**

ii. The purpose of online shopping is to save time, save money. Through online shopping one can save his valuable time. One can watch and select things he wants to buy.

iii. Through online shopping we can save our money because prices are less than market prices and we receive our bought things at our home. No need to go anywhere and do shopping.

iv. We can get different varieties of things online and we can choose which one we want.

v. Through Online Shopping**,** a person who wants to buy is a lifestyle e-commerce web application, which retails various fashion and lifestyle products.

vi. This project allows viewing various products available enables registered users to purchase desired products instantly using PayPal payment processor (Instant Pay) and also can place order by using Cash on Delivery (Pay Later) option.

vii. This project provides an easy access to Administrators and Managers to view orders placed using Pay Later and Instant Pay options.

viii. In order to develop an e-commerce website, a number of Technologies must be studied and understood.

2.3 Advantages of System

i. Increase in Sales.

ii. Increase in Customers.

iii. Ability to Open 24/7.

iv. Instant Processing of transactions.

v. Increase business reach.

vi. Use electronic payments refers to paperless monetary transactions.

vii. These business transactions occur either as business-to-business (B2B), business-to-consumer (B2C), consumer-to-consumer (C2C) or consumer-to-business (C2B). The terms e-commerce and e-business are often used interchangeably.

viii. The term **E-Tail** (electronic retailing) is also sometimes used in reference to **transactional processes** for **Sport Portal or Online Shopping**

CHAPTER 3. SYSTEM REQUIREMENT'S

3.1 Hardware requirements

The section of hardware configuration is an important task related to the software development insufficient random-access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

Operating system: - Windows 7 or newer.

Processor: - i3 and above

RAM: - 2GB or above.

HD: - 50 GB or above.

3.2 Software requirements

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system. This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system.

Operating System: Linux, Windows etc.

Development Tool: HTML, CSS, JavaScript.

Data Base: SQLite3.

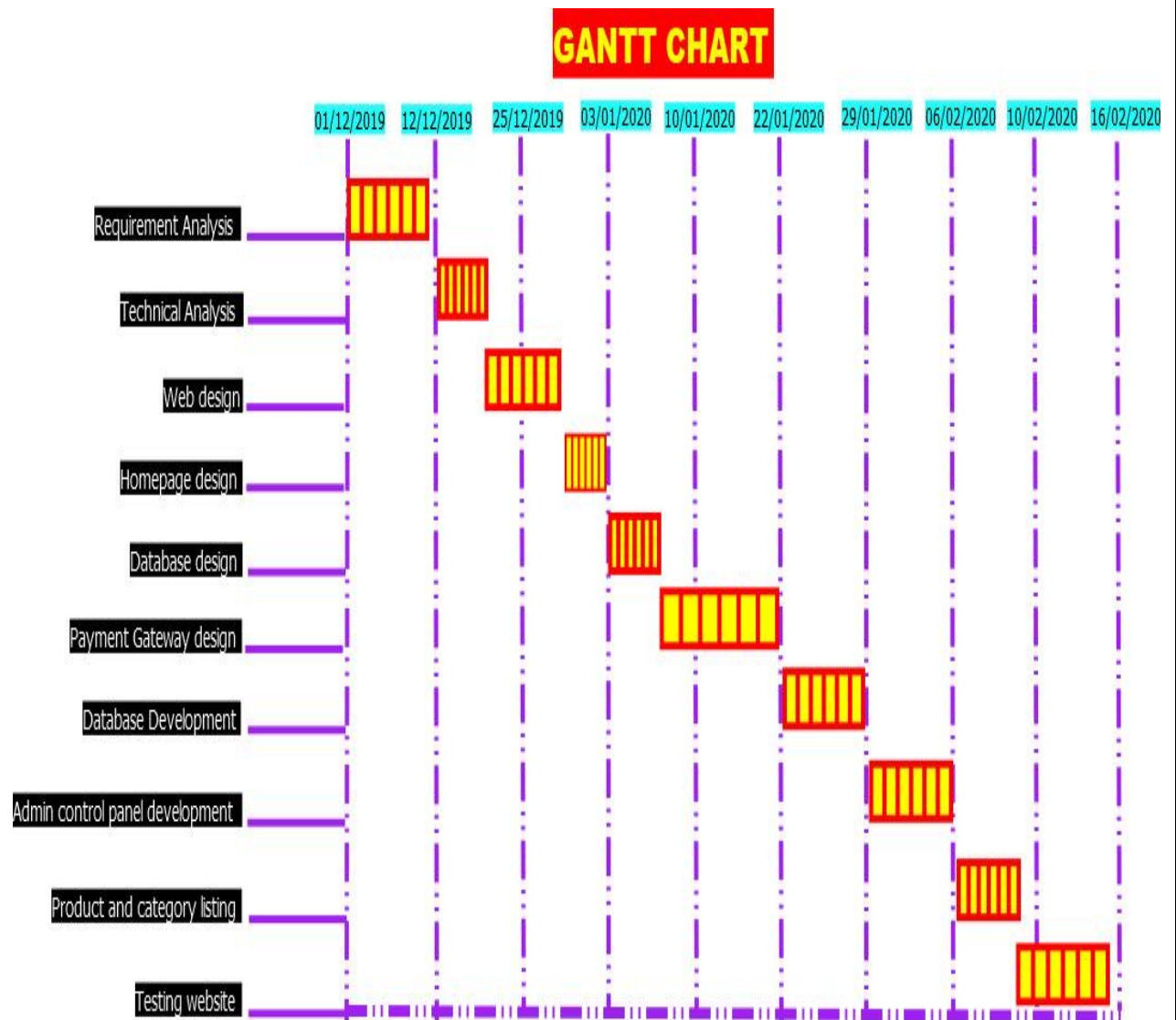Browser: Any

CHAPTER 4. SYSTEM PLANNING (GANTT CHART)

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows you to see at a glance:

 - ➢ What the various activities are
 - ➢ When each activity begins and ends
 - ➢ How long each activity is scheduled to last
 - ➢ Where activities overlap with other activities, and by how much
 - ➢ The start and end date of the whole project

The first Gantt chart was devised in the mid-1890s by Karol Adamiecki, a Polish engineer who ran a steelwork in southern Poland and had become interested in management ideas and techniques. Some 15 years after Adamiecki, Henry Gantt, an American engineer and project management consultant, devised his own version of the chart and it was this that became widely known and popular in western countries.

Consequently, it was Henry Gantt whose name was to become associated with charts of this type. Originally Gantt charts were prepared laboriously by hand; each time a project changed it was necessary to amend or redraw the chart and this limited their usefulness, continual change being a feature of most projects. Nowadays, however, with the advent of computers and project management software, Gantt charts can be created, updated and printed easily.

GANTT CHART

	01/12/2019	12/12/2019	25/12/2019	03/01/2020	10/01/2020	22/01/2020	29/01/2020	06/02/2020	10/02/2020	16/02/2020
Requirement Analysis	▓									
Technical Analysis		▓								
Web design			▓							
Homepage design				▓						
Database design					▓					
Payment Gateway design						▓				
Database Development							▓			
Admin control panel development								▓		
Product and category listing									▓	
Testing website										▓

The steps which carried out according to the Gantt chart are as follows: -

i.   Requirement Analysis phase is started on 01/12/2019 and continued up to 11/12/2019. In this phase analysis of the project is done.

ii. Technical Analysis phase is started on 12/12/2019 and continued up to 23/12/2019. In this phase actual understanding of the contents used for the project is done.

iii. Web Design phase is started on 24/12/2019 and continued up to 31/12/2019. In this phase all the related information is gathered.

iv. Homepage design phase is started on 01/01/2020 and continued up to 03/01/2020. In this phase homepage design is done.

v. Database design phase is started on 04/01/2020 and continued up to 08/01/2019. In this phase form design is done.

vi. Payment Gateway Design phase is started on 09/01/2020 and continued up to 22/01/2020. In this phase coding is done.

vii. Database Development phase is started on 23/01/2020 and continued up to 29/01/2020. In this phase testing is done.

viii. Admin Control Panel Development phase is started on 30/01/2020 and continued up to 06/02/2020. In this phase actual implementation is done.

ix. Product Category listing phase is started on 07/02/2020 and continued up to 09/02/2020. In this phase documentation is done.

x. Testing Website phase is started on 10/02/2020 and continued up to 16/02/2020. In this phase testing is done.

CHAPTER 5. SYSTEM DESIGN APPROACH

5.1 ER Diagram

The Entity Relationship Diagram was originally proposed by Peter in 1976 [Chen 76] as a way to unify the network and relational database views.

ER model is used to model the logical view of the system from data perspective which consist of these components: - i] Entity. ii] Entity type. iii] Entity set.

The connectivity for relations are one to one, one to many, and many to many. There are some notations of ER model that are Entities, Attributes, Relationships, Cardinality and Existence

Components of the ER Diagram

This model is based on three basic concepts:

- Entities
- Attributes
- Relationships

5.1.1 Entity

A real-world thing either living or non-living that is easily recognizable and non recognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world.

An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.
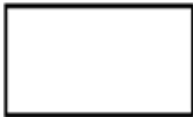
5.1.2 Attributes

It is a single-valued property of either an entity-type or a relationship-type. An attribute is represented by an Ellipse

5.1.3 Relationship

Relationship is nothing but an association among two or more entities.
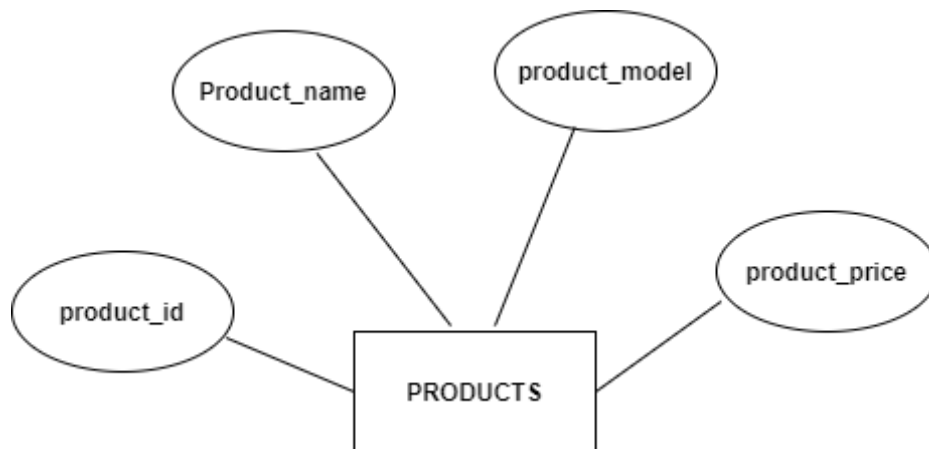
5.1.4 Symbols used

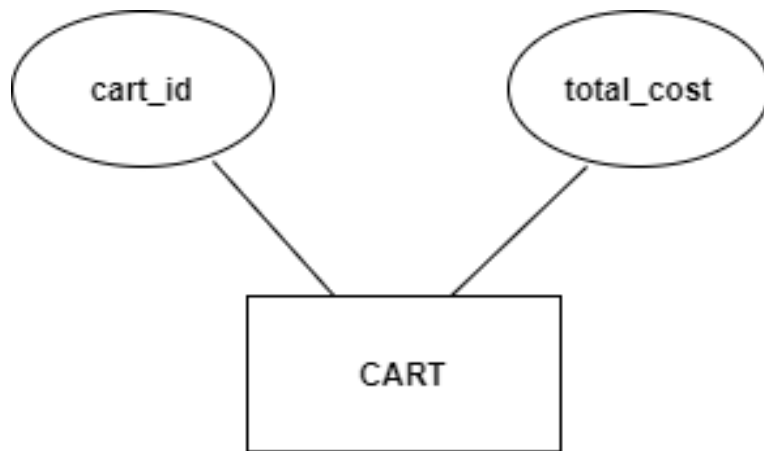| Sr no	Symbol	Name	Description
1		Rectangles	Represents Entity Sets
2		Ellipses	Represents Attributes
3		Diamonds	Represents Relationship sets
4		Lines	Link Attribute to Entity Sets and Entity Sets to Relationship Sets

5.1.5 All Entities with Attributes
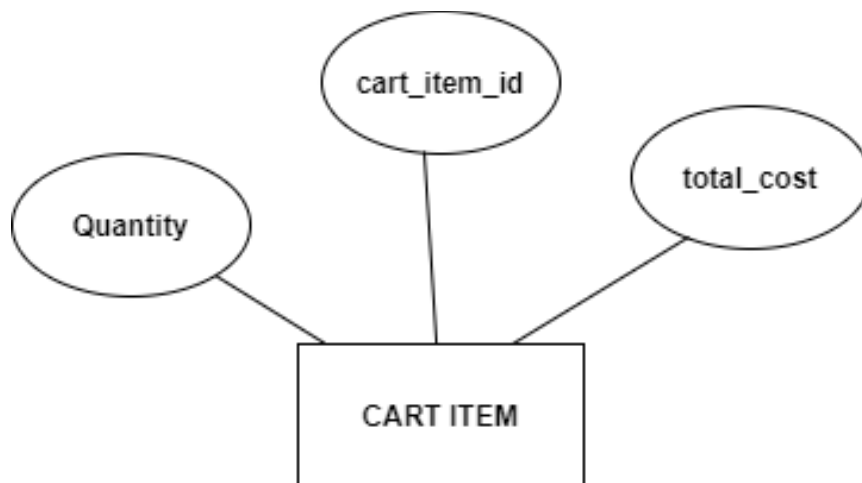
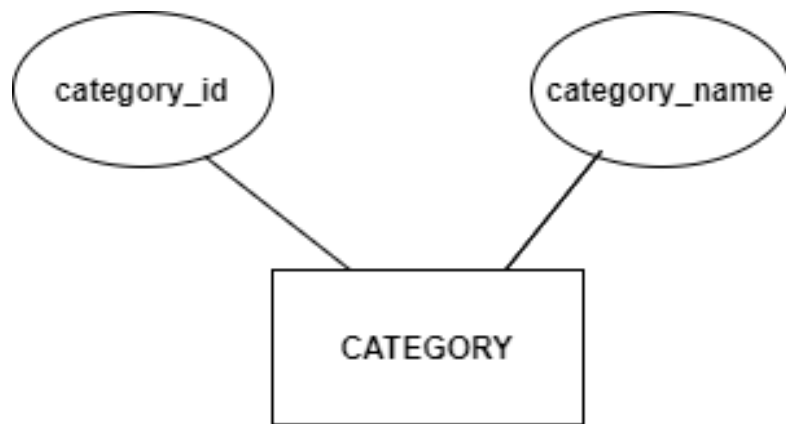### i] E-R Diagram (For Customers)
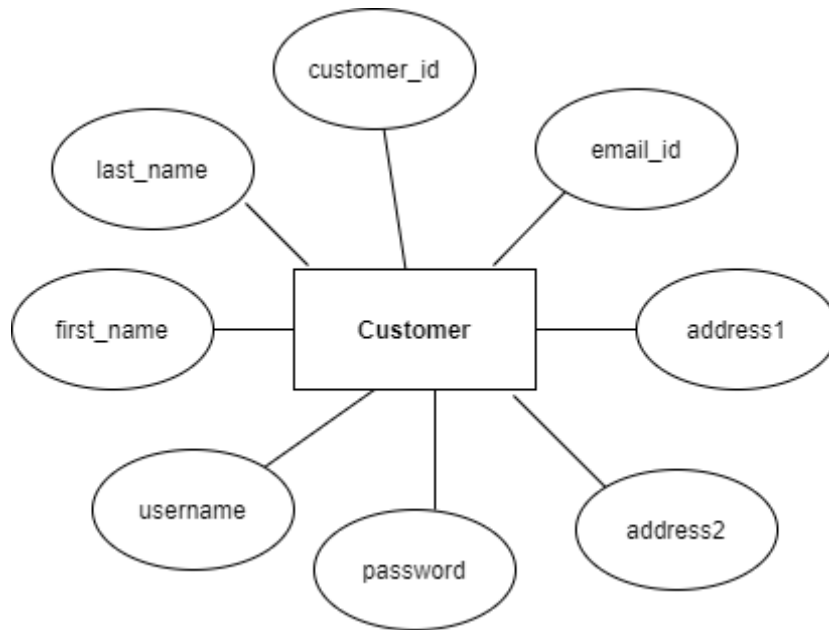
ii]E-R Diagram (For Products)

iii] E-R Diagram (For Cart)

iv] E-R Diagram (For Cart Item)

v]E-R Diagram (For Category)

category_id   category_name

CATEGORY

vi]E-R Diagram (Payment)

payment_id

amount   payment_type

Payment
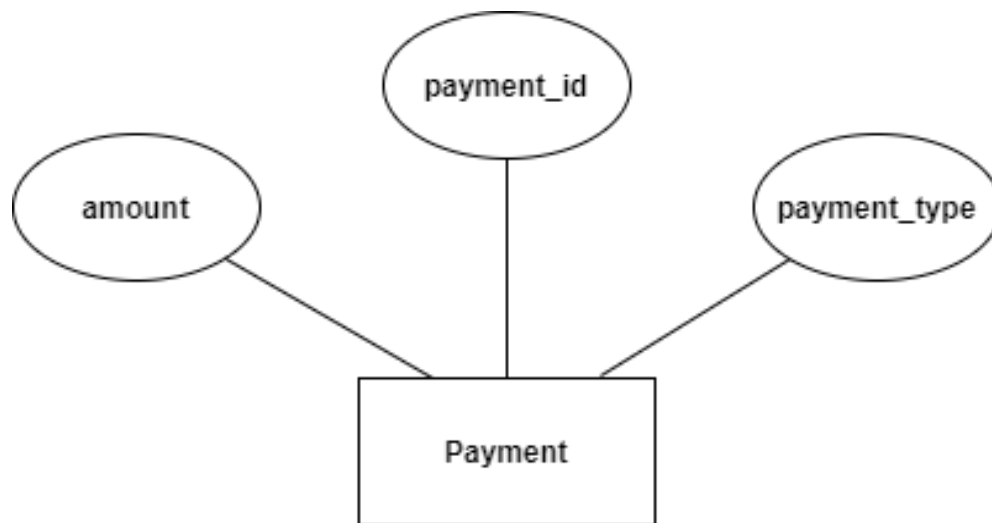
vii]E-R Diagram Sport Portal
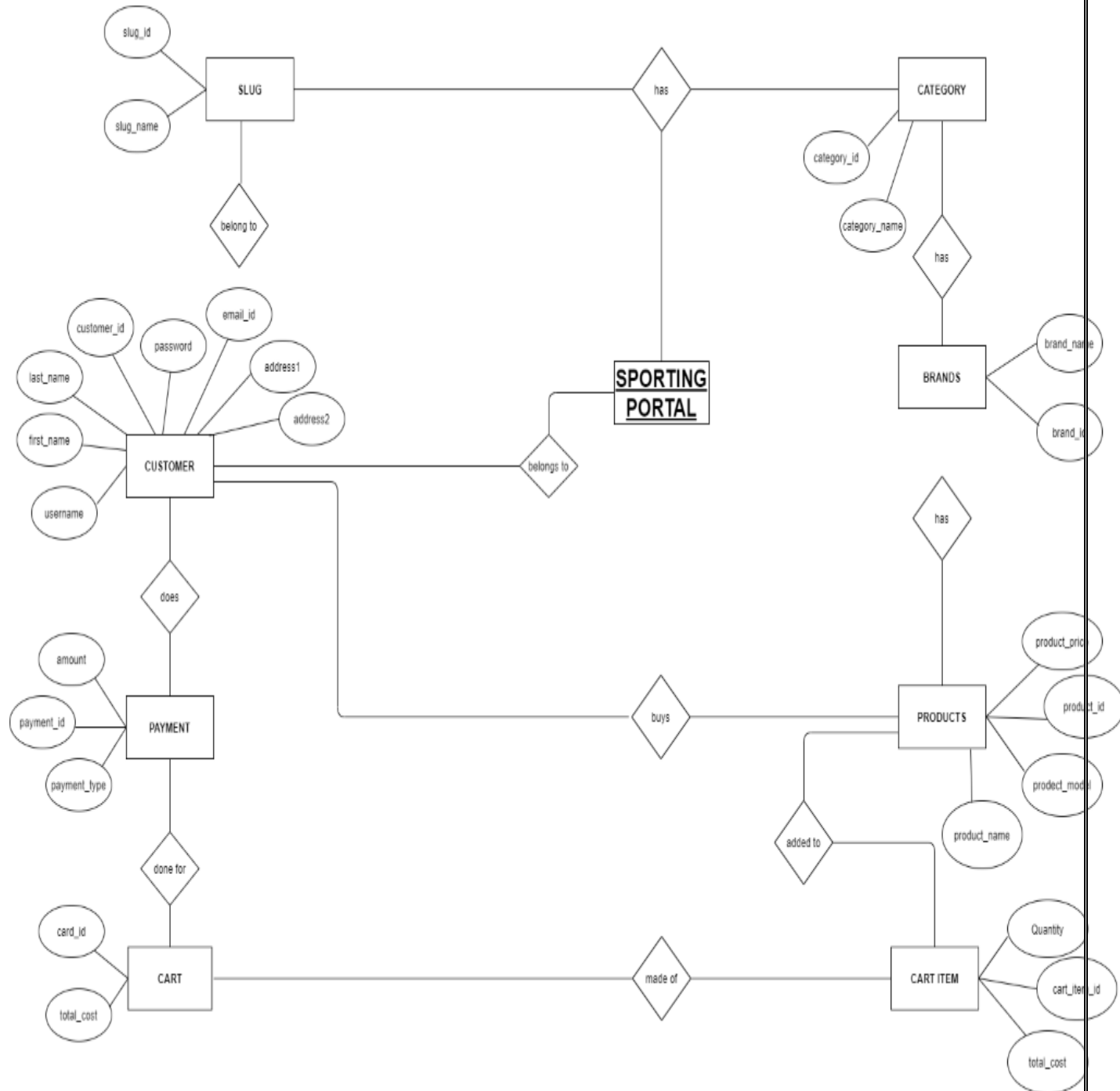
5.2 Activity Diagram

Activity Diagrams are graphical representations of workflow of stepwise activities and actions with support for choice, iteration and concurrency.

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

5.2.1 Purpose of Activity Diagram

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.
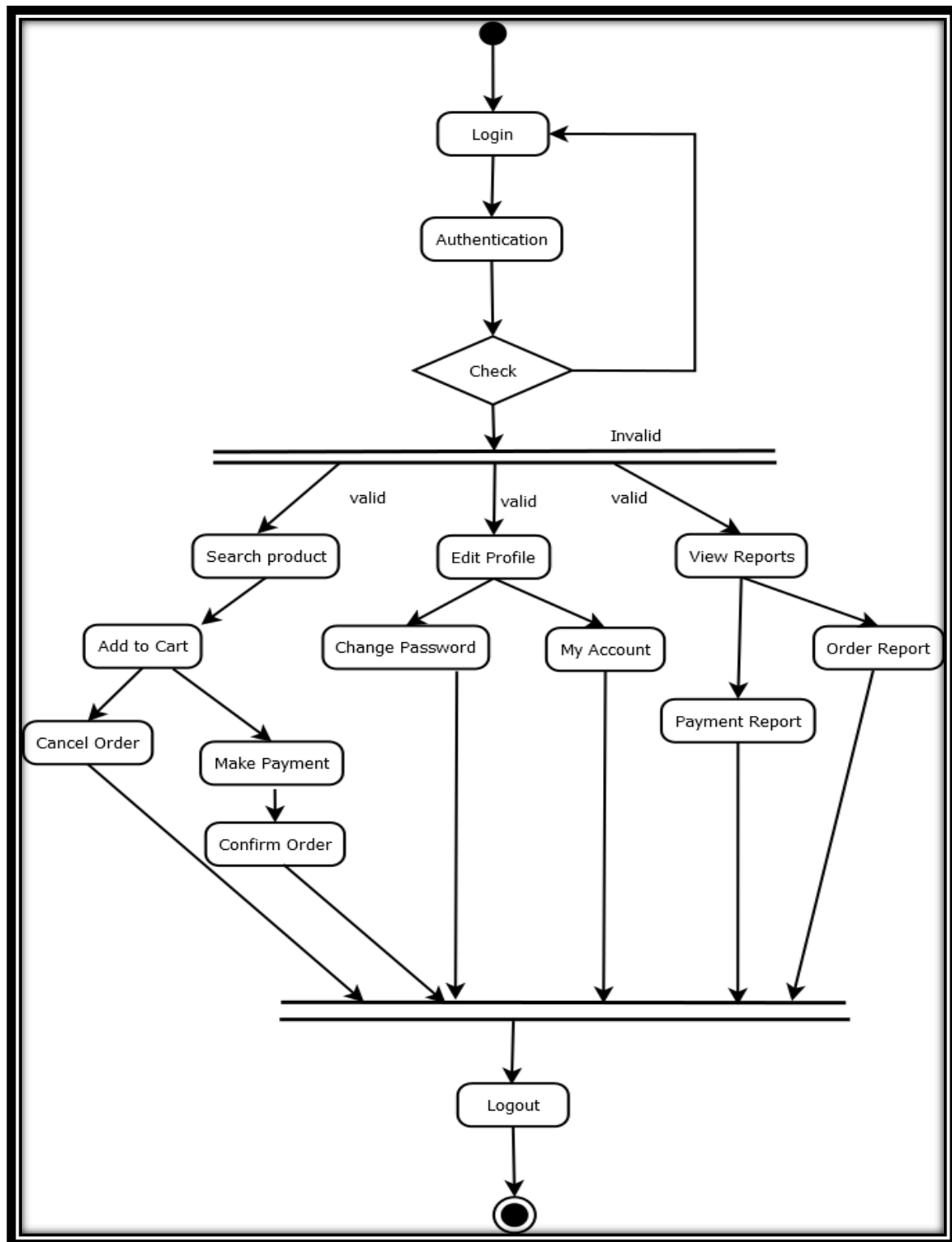
The purpose of an activity diagram can be described as −

- ➢ Draw the activity flow of a system.
- ➢ Describe the sequence from one activity to another.
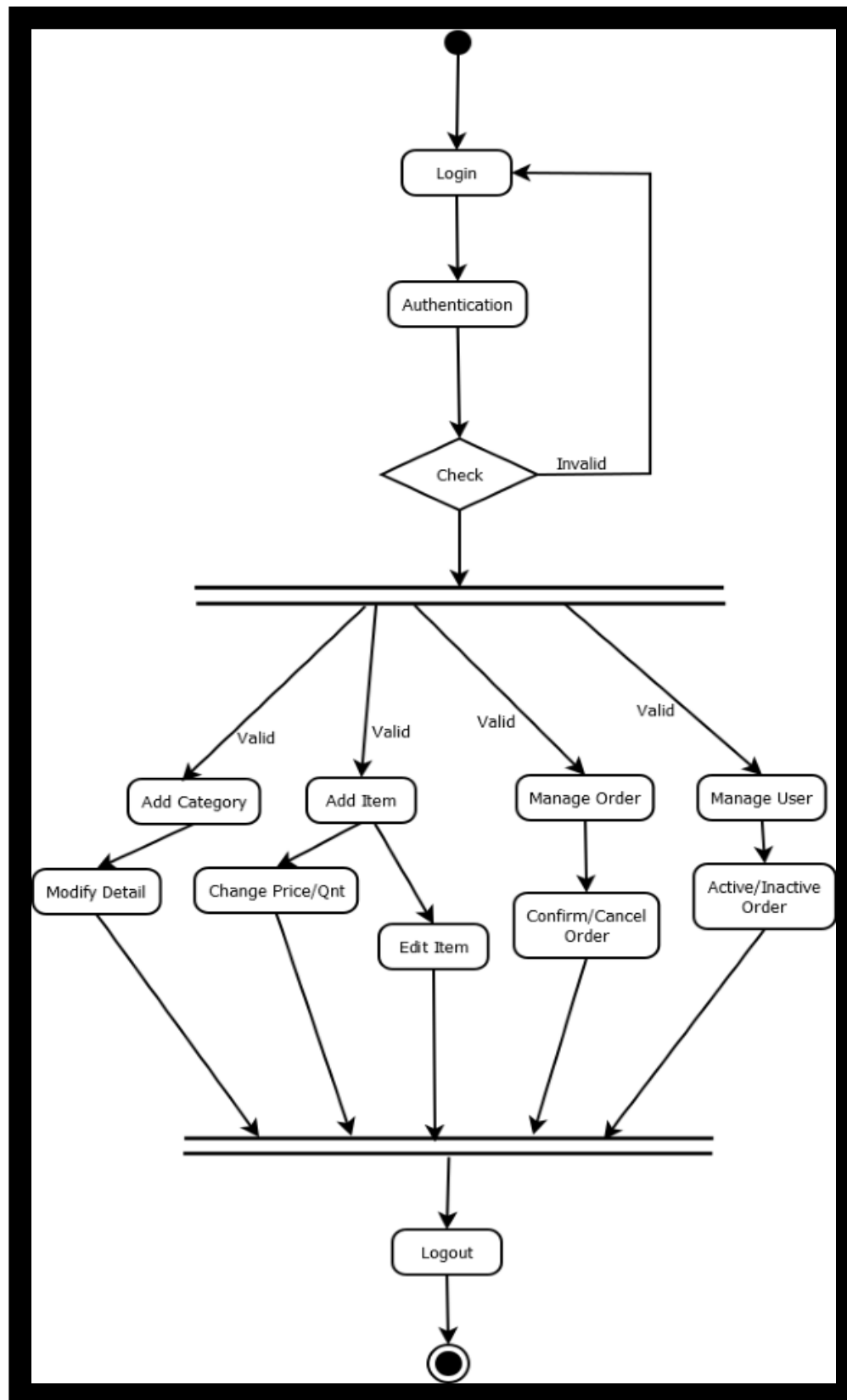- ➢ Describe the parallel, branched and concurrent flow of the system.

5.2.2 Symbols Used in Activity Diagram

Start	●
Stop	◉
Process	⬭
Decision	◇
Link	→
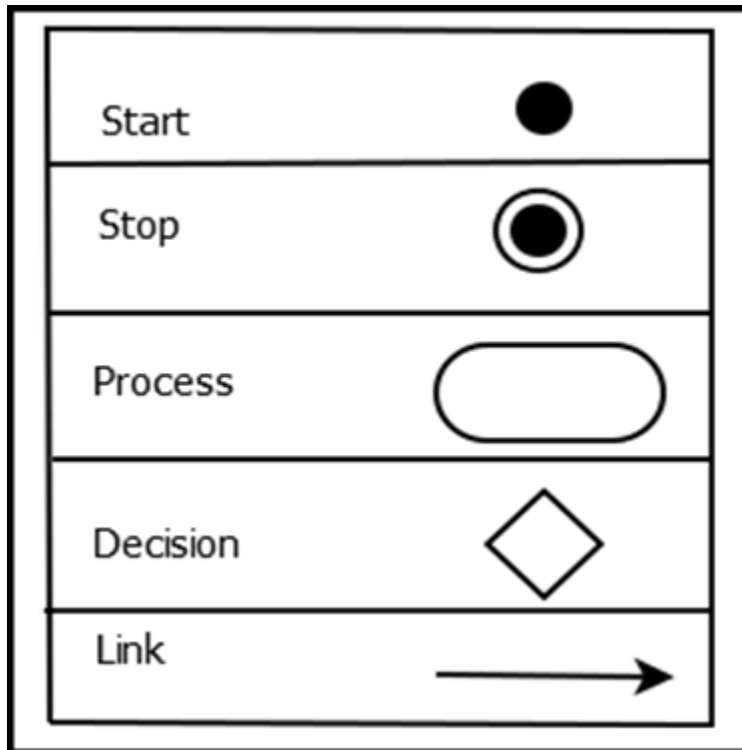
5.2.3 User Activity Diagram

5.2.4 Admin Activity Diagram

5.3 DFD Data Flow Diagram

A DFD is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and process that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation.

It has the purpose of clarifying system requirements and identifying major transformations. So, it is the starting point of the design phase that functionality decomposes the requirements specifications down to the lowest level of detail.

DFD can be considered as an abstraction of the logic of an information-oriented or a process-oriented system flow-chart.
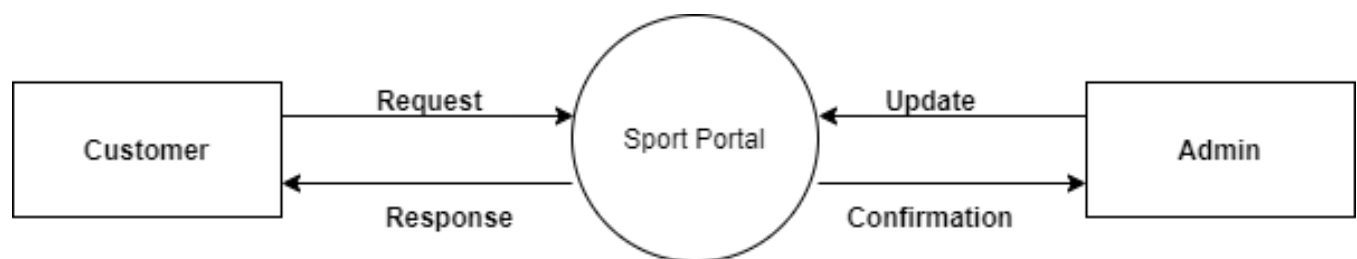
5.3.1 Data Flow Diagram Levels

Data flow diagrams are also categorized by level. Starting with the most basic, level 0, DFDs get increasingly complex as the level increases. As you build your own data flow diagram, you will need to decide which level your diagram will be.
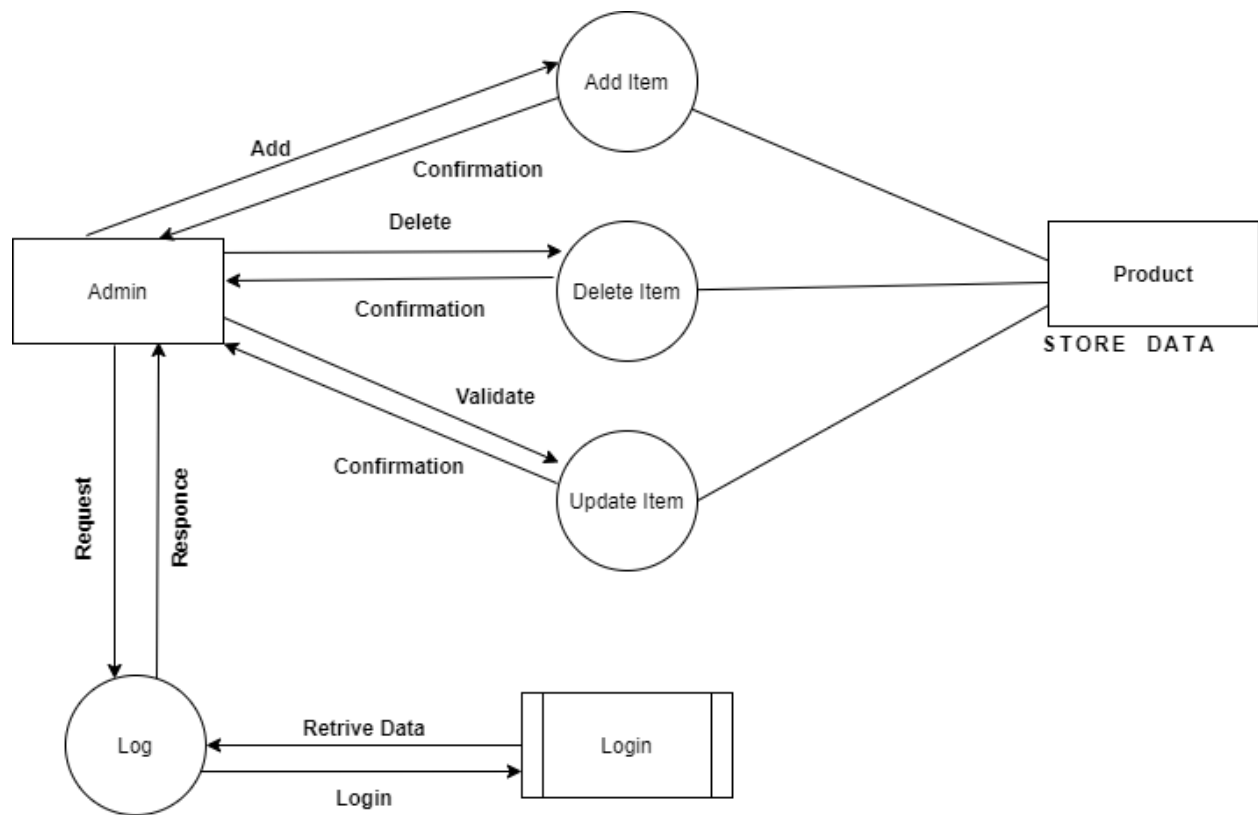
Level 0 DFDs, also known as context diagrams, are the most basic data flow diagrams. They provide a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities.

Level 1 DFDs, are still a general overview, but they go into more detail than a context diagram. In a level 1 data flow diagram, the single process node from the context diagram is broken down into subprocesses. As these processes are added, the diagram will need additional data flows and data stores to link them together.
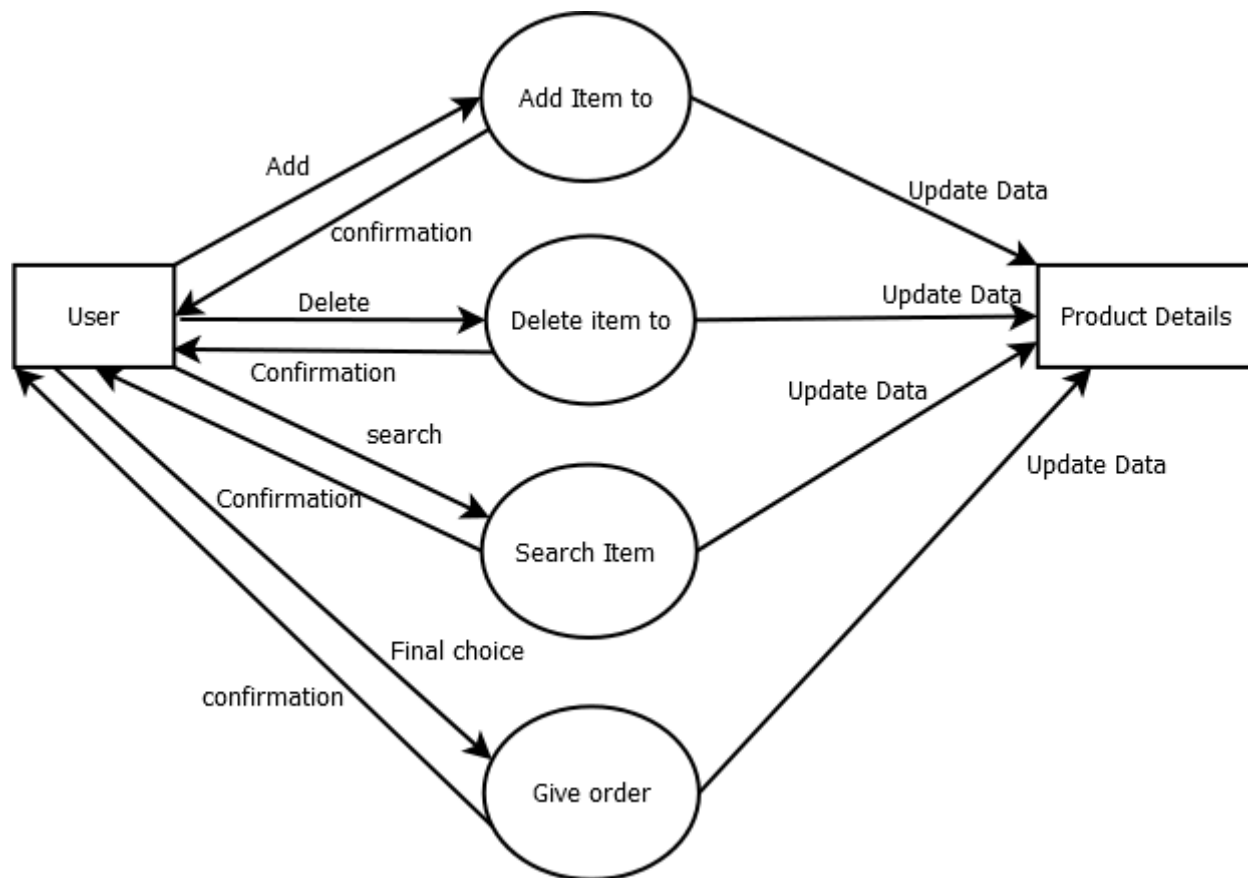
5.3.2 Level 0 DFD

5.3.3Level 1(For Admin)

Add Item

Add

Confirmation

Delete

Admin

Delete Item

Confirmation

Product

STORE DATA

Validate

Confirmation

Update Item

Request

Responce

Log

Retrive Data

Login

Login

5.3.4 Level 1(For User)

User → Add → Add Item to → Update Data → Product Details

Add Item to ← confirmation → User

User → Delete → Delete item to → Update Data → Product Details

Delete item to → Confirmation → User

User → search → Search Item → Update Data → Product Details

Search Item → Confirmation → User

User → Final choice → Give order → Update Data → Product Details

Give order → confirmation → User

5.4 Class Diagram

A description of a group of objects all with similar roles in the system, which consists of:

- ➤ **Structural features** (attributes) define what objects of the class "know". Represent the state of an object of the class. Are descriptions of the structural or static features of a class.
- ➤ **Behavioral features** (operations) define what objects of the class "can do. Define the way in which objects may interact. Operations are descriptions of behavioral or dynamic features of a class
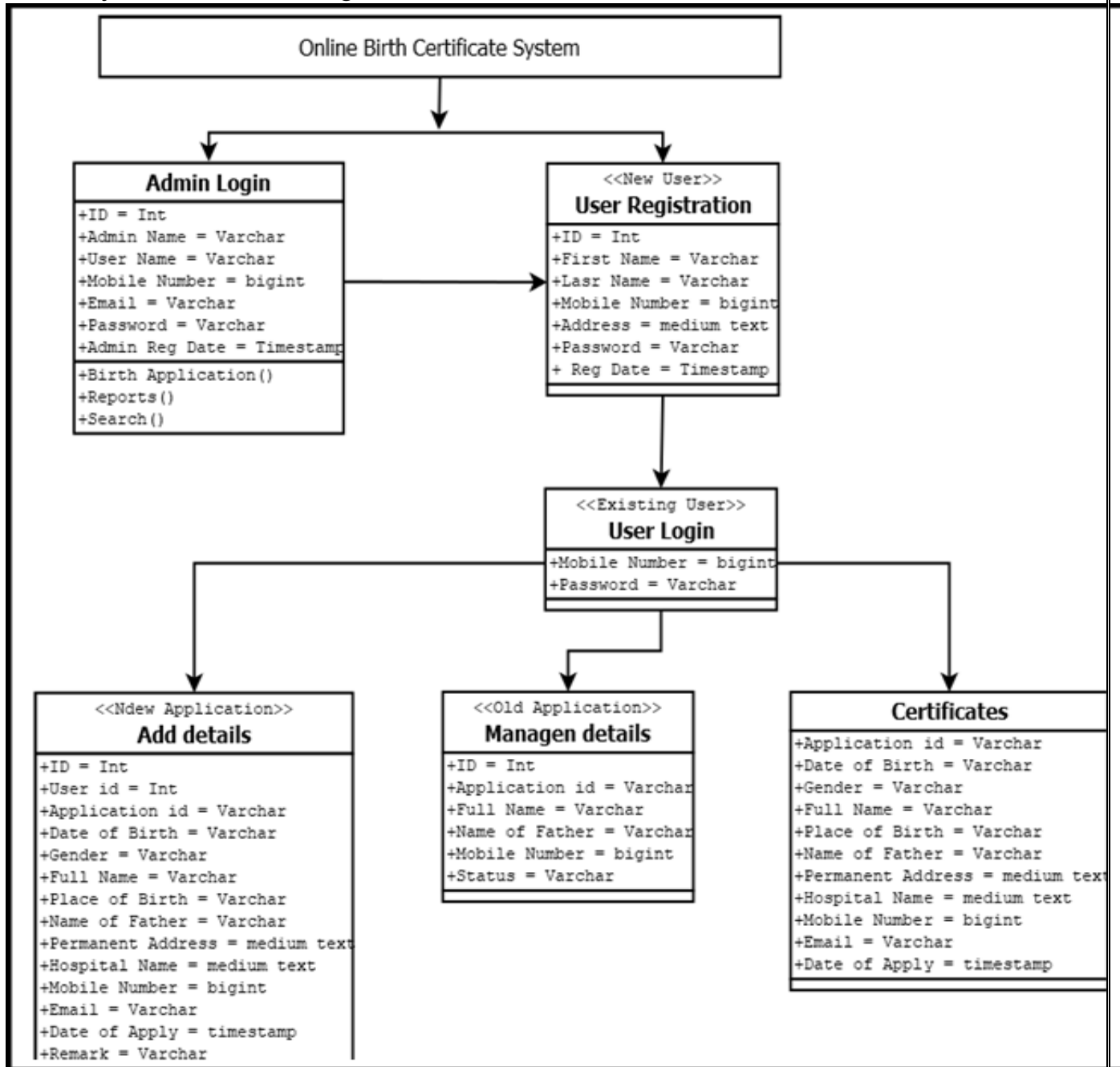
Class Notation

A class notation consists of three parts:

1. **Class Name**: - The name of the class appears in the first partition.
2. **Class Attributes**: - Attributes are shown in the second partition. The attribute type is shown after the colon. Attributes map onto member variables (data members) in code.
3. **Class Operations** (Methods): - Operations are shown in the third partition. They are services the class provides. The return type of a method is shown after the colon at the end of the method signature. The return type of method parameters is shown after the colon following the parameter name. Operations map onto class methods in code.
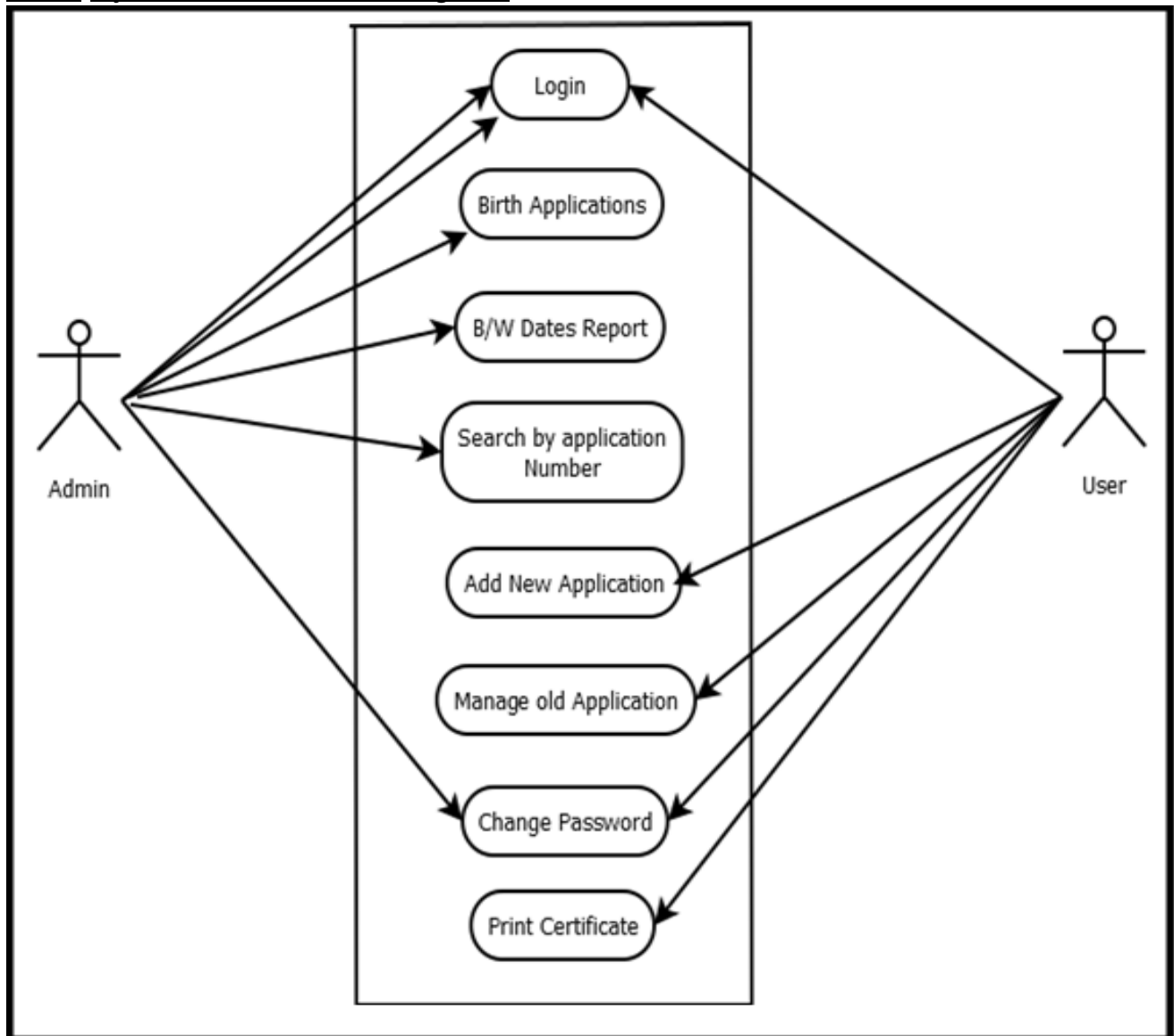
5.4.1 System's Class Diagram

```
                    ┌─────────────────────────────────────────┐
                    │      Online Birth Certificate System     │
                    └─────────────────────────────────────────┘
                                       │
                                       ▼
```

Online Birth Certificate System

Admin Login
+ID = Int
+Admin Name = Varchar
+User Name = Varchar
+Mobile Number = bigint
+Email = Varchar
+Password = Varchar
+Admin Reg Date = Timestamp

+Birth Application()
+Reports()
+Search()

<<New User>>
User Registration
+ID = Int
+First Name = Varchar
+Lasr Name = Varchar
+Mobile Number = bigint
+Address = medium text
+Password = Varchar
+ Reg Date = Timestamp

<<Existing User>>
User Login
+Mobile Number = bigint
+Password = Varchar

<<Ndew Application>>
Add details
+ID = Int
+User id = Int
+Application id = Varchar
+Date of Birth = Varchar
+Gender = Varchar
+Full Name = Varchar
+Place of Birth = Varchar
+Name of Father = Varchar
+Permanent Address = medium text
+Hospital Name = medium text
+Mobile Number = bigint
+Email = Varchar
+Date of Apply = timestamp
+Remark = Varchar

<<Old Application>>
Managen details
+ID = Int
+Application id = Varchar
+Full Name = Varchar
+Name of Father = Varchar
+Mobile Number = bigint
+Status = Varchar

Certificates
+Application id = Varchar
+Date of Birth = Varchar
+Gender = Varchar
+Full Name = Varchar
+Place of Birth = Varchar
+Name of Father = Varchar
+Permanent Address = medium text
+Hospital Name = medium text
+Mobile Number = bigint
+Email = Varchar
+Date of Apply = timestamp

5.5 Use Case Diagram

A system involves a set of use cases and a set of actors. Each use case represents a slice of the functionality the system provides. The UML has a graphical notation for summarizing use cases. A rectangle contains the use case for a system with the actors listed on the outside. The name of the system may be written near a side of the rectangle. A name within an ellipse denote a use case
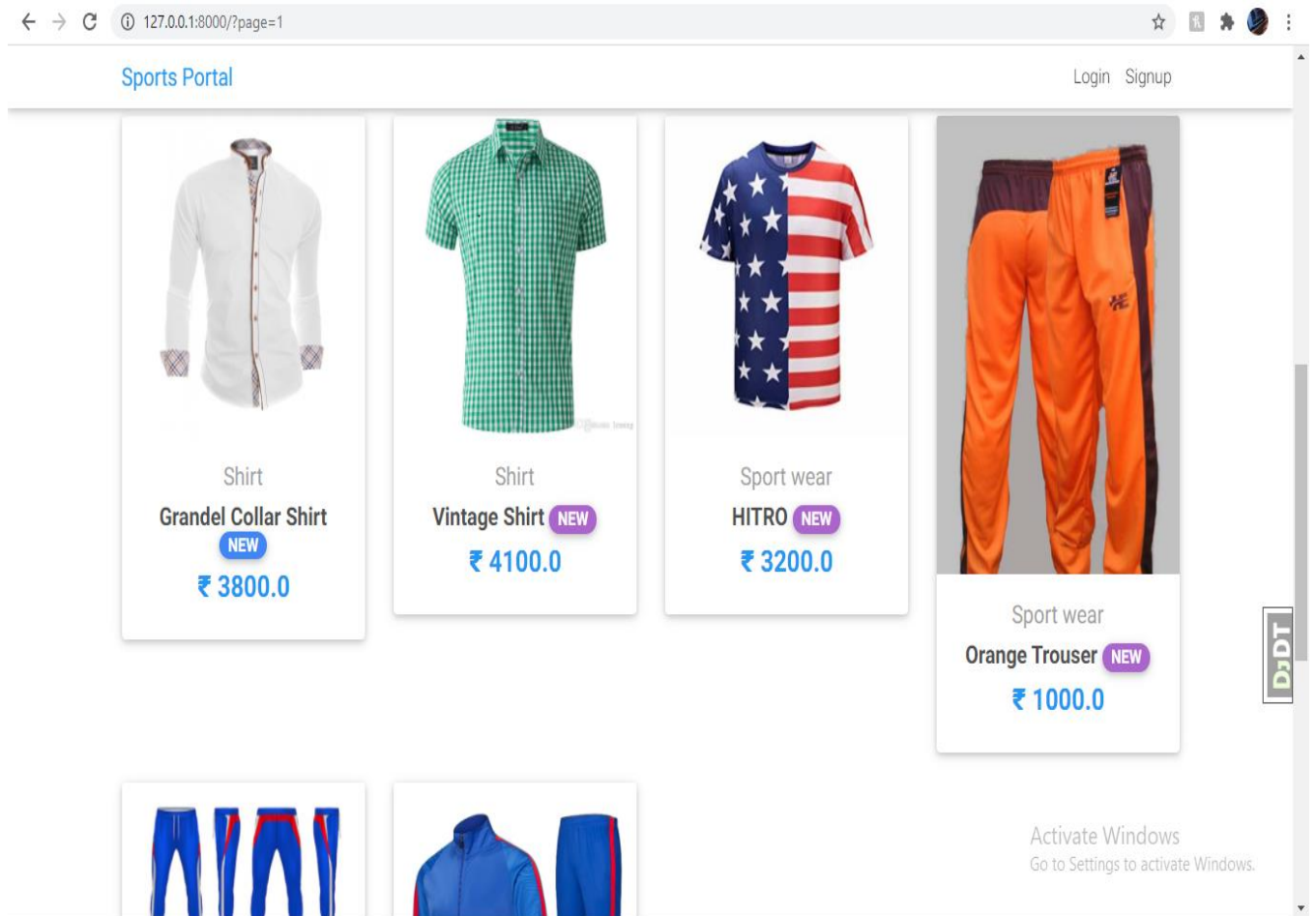
A stick man icon denotes an actor, with the name being placed below or adjacent to the icon. Solid connect use cases to participating actors.

5.5.1 System's Use case Diagram

CHAPTER 6. GUI INTERFACE

i] Home Page

ii] Admin Login

iii] Admin Dashboard

iv] View Users

v] <u>View Ordered Items</u>

vi] <u>User Sign in Page</u>

Sports Portal

Login Signup

Sign In

If you have not created an account yet, then please sign up first.

Username*

> Username

Password*

> Password

☐ Remember Me

FORGOT PASSWORD?   SIGN IN

f 🐦 ▶ ⭘

Activate Windows
Go to Settings to activate Windows.

vii] User Sign up Page

viii] Product information Page

ix] Cart Page

Sports Portal   **1** 🛒 Cart Logout

Order Summary

#	Item title	Price	Quantity	Total Item Price	
1	Outdoor Sport Suit	500.0	− 1 +	₹469.0 Saving ₹31.0	🗑
Order Total				₹469.0	

CONTINUE SHOPPING   **PROCEED TO CHECKOUT**

f 🐦 ▶ 🔘

© 2020 Copyright: Rohit Mokal

Activate Windows
Go to Settings to activate Windows.

x] Checkout Form Page

1 🛒 Cart   Logout

Checkout form

Shipping address

Address
1234 Main St

Address 2 (optional)
Apartment or suite

Country
(select country) ⇅

Zip
Zip code

☐ Billing address is the same as my shipping address
☐ Save as default shipping address
☐ Use default shipping address: 12345

Billing address

Address
1234 Main St

Address 2 (optional)
Apartment or suite

Country
(select country) ⇅

Zip
Zip code

☐ Save as default billing address
☐ Use default billing address: 12345

Payment option

○ Stripe
○ PayPal
○ COD

CONTINUE TO CHECKOUT

Your cart   1

1 x Outdoor Sport Suit   ₹469.0
This is For Any game Practice

Total (INR)   ₹469.0

Promo code   REDEEM

xi] Payment Page

1 🛒 Cart   Logout

Payment

☐ Use default card: **** **** **** 4242 Exp: 4/2024

Credit or debit card

| 💳 Card number | MM / YY CVC |

☐ Save for future purchases

Submit Payment

Your cart   1

1 x Outdoor Sport Suit   ₹469.0
This is For Any game Practice

Total (INR)   ₹469.0

CHAPTER 7. METHODOLOGY USED FOR TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies, a test plan is carried out on each module. The various tests performed in "Network Backup System" are unit testing, integration testing and user acceptance testing.

7.1 Types of Testing

i. Unit Testing: - The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its action as required.

ii. Integration Testing: - Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the Server module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

iii. Functional Testing: - This type of testing ignores the internal part and focuses on the output is as per the requirement.

iv.   System Testing: - The entire software system is tested. The goal is to see if the software meets its requirements.

v.   End to End Testing: - Similar to system testing, involves testing of complete application environment in a situation that mimics real world use, such as interacting with database, using network communication, or interacting with other hardware, application, or system if appropriate.

vi.   Regression Testing: - Testing the application as a whole for the modification in any module or functionality.

vii.   Acceptance Testing: - It is sometimes performed with realist data of the client to demonstrate that the software is working satisfactorily. It focuses on the external behaviour of the system.

viii.   Usability Testing: - User-friendliness check. Application flow is tested, can new user understand the application easily, proper help documented whenever user stuck at any point. Basically, system navigation is checked in this testing.

ix.   Comparison Testing: - Comparison of product strength and weakness with previous version or other similar products.

x.   Validation Testing: - Software validation will be achieved through a series of black box testing.

7.3 Test Cases done in the Project

i] Password Verification.

It checks the validity of the password which has been entered by the user. If a correct password is entered then only it will accept it otherwise, if a wrong password is entered then an error message saying "Invalid details" will pop up.

ii] Username Verification.

This checks that a valid username is being entered. If wrong username is entered than the message "Invalid username" will appear.

iii] Admin's Username Verification.

If the admin who wants to login in enters the wrong username the same "Invalid details" pop up will appear on the screen.

iv] Admin's Password Verification.

If the admin who wants to login in enters the wrong password then "Invalid details" pop up will appear on the screen.

v] LOGIN FOR USER

Serial No	Description	Expected Result	Actual Result	Result
1.	This page contains 2 fields user name and password and a login button to submit the information. User is entering correct information.	User home page should open after successful login.	Respective user home page is opening after successful login by user.	Passed
2.	If either user name or password is filled incorrect or left blank.	An error messages should be displayed and user should be asked fill the information again.	When wrong information is entered by user then an error message is displayed.	Passed

vi] USER RESISTRATION PAGE

Serial No	Description	Expected Result	Actual Result	Result
1.	User registration page 1 consist of detail information about User and a submit button to submit the information .Here user is entering correct information	After submitting information User registration page 2 should be displayed	After submitting information User registration page 2 is displayed.	Passed
2.	If the information entered by user in incorrect or left somewhere blank.	An error message should be displayed and ask the user to fill the information again.	An error message is occurred if the information is incorrect or left blank.	Passed

vii] Actions   Expected Result

Correct Password ⟶ Validation clear

Wrong Password ⟶ Display an error prompt

Correct Username ⟶ Validation clear

Wrong Username ⟶ Display an error prompt

Correct Admin Username ⟶ Validation Clear

Wrong Admin Username ⟶ Display an error prompt

Correct Admin Password ⟶ Validation clear

Wrong Admin Password ⟶ Display an error prompt

CHAPTER 8. CONCLUSION

The package is designed in such a way that modifications and updates can made in future. The system provides user friendly interface for better performance and faster as compared to existing system.

It gives appropriate access to authorized user's depending on permission. Updating information on applications verification process can be made more advance in future. The system has lots of scope for future changes if needed.

The Sport Portal is designed for the applicant's time consumption and fastening the process of providing Sport Products to the appropriate User/Customers.

CHAPTER 9. FUTURE ENHANCEMENT

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

- Payment Method can be Updated in future for user Friendly option.
- Categorized products can be done.
- More advance technology for making the verification strong can be implemented.

CHAPTER 10. REFERENCES

➢ Whitten, Bentley and Dittman 2004. System Analysis and Design Methods (5th edition) McGraw-Hill-New York.

➢ https://wikipedia.org/wiki/html

➢ https://wikipedia.org/wiki/css

➢ https://www.djangoproject.com/start/

➢ https://docs.djangoproject.com/en/3.1/ref/contrib/postgres/

➢ https://stripe.com/docs/payments/checkout

➢ https://en.wikipedia.org/wiki/Python_(programming_language)

➢ https://www.tutorialspoint.com/python/index.htm