FASTO COURIER MANAGEMENT SYSTEM

Project Report Submitted By

STEPHY GRACE OOMMEN

Reg. No: AJC17MCA-I054

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**FASTO COURIER MANAGEMENT SYSTEM"** is the bonafide work of **STEPHY GRACE OOMMEN (Reg.No:AJC17MCA-I054)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Ms. Gloriya Mathew
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev.Fr.Dr.Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **"FASTO COURIER MANAGEMENT SYSTEM"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date: **STEPHY GRACE OOMMEN**

KANJIRAPPALLY **Reg.No:AJC17MCA-I054**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude and respect towards our Head of the Department **Rev.Fr.Dr.Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinators **Rev.Fr.Dr.Rubin Thottupurathu Jose** and **Ms**. **Meera Rose Mathew** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**,** **Ms. Gloriya Mathew** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

STEPHY GRACE OOMMEN

ABSTRACT

FASTO Courier Management System mainly focus on the courier service process that's performed in online mode. The existing courier delivery services use legacy systems and technologies. Due to which the travel time of the courier delays and people have a sub-standard experience. People are looking for a solution that provides online courier pick-up and drop facility with standard costs based on the dimensions and the traveling distance across all varieties of goods and services. The Proposed system aims at courier pick-up and drop facility by keeping track of details about day-to-day courier senders, receivers, courier men, hub details using courier services. The basic work flow goes in the following manner: Login section will have login facility for the admin who operates the system and online tracking of consignment and shipping details for domestic shipping. While taking orders, it will take the details of sender and receiver. During billing process, system generates a Tracking ID for their products. Through this Tracking ID customers will able to track their couriers from any location using internet. The system sends a confirmation email to the users email id. The Proposed system will provide status of the product after placing orders and brings transparency to the system and help to save money.

CONTENT

SL.NO	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	10
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	PHP	11
3.3.2	MYSQL	12
4	SYSTEM DESIGN	14
4.1	INTRODUCTION	15
4.2	UML DIAGRAM	15
4.2.1	USE CASE DIAGRAM	16
4.2.2	SEQUENCE DIAGRAM	19
4.2.3	COMPONENT DIAGRAM	21
4.2.4	ACTIVITY DIAGRAM	22
4.2.5	CLASS DIAGRAM	24

4.2.6	COLLABORATION DIAGRAM	25
4.2.7	STATE CHART DIAGRAM	27
4.2.8	DEPLOYMENT DIAGRAM	28
4.3	USER INTERFACE DESIGN USING FIGMA	29
4.4	DATA BASE DESIGN	32
5	SYSTEM TESTING	40
5.1	INTRODUCTION	41
5.2	TEST PLAN	42
5.2.1	UNIT TESTING	42
5.2.2	SELENIUM AUTOMATION TESTING	43
5.2.2.1	TEST CASE FOR LOGIN	44
5.2.2.2	TEST CASE FOR UPDATE PASSWORD	47
5.2.3	INTEGRATION TESTING	49
5.2.4	VALIDATION TESTING	49
5.2.5	USER ACCEPTANCE TESTING	49
6	IMPLEMENTATION	50
6.1	INTRODUCTION	51
6.2	IMPLEMENTATION PROCEDURES	51
6.2.1	USER TRAINING	52
6.2.2	TRAINING ON THE APPLICATION SOFTWARE	52
6.2.3	SYSTEM MAINTENANCE	52
7	CONCLUSION AND FUTURE SCOPE	53
7.1	CONCLUSION	54
7.2	FUTURE SCOPE	54
8	BIBLIOGRAPHY	55
9	APPENDIX	57
9.1	SAMPLE CODE	58
9.2	SCREENSHOTS	69

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

“FASTO COURIER MANAGEMENT SYSTEM” is a web application which is meant to help the customers to send their courier easily. The existing courier delivery services use legacy systems and technologies. Due to which the travel time of the courier delays and people have a sub-standard experience. People are looking for a solution that provides online courier pick-up and drop facility with standard costs based on the dimensions and the traveling distance across all varieties of goods and services. The Proposed system aims at courier pick-up and drop facility by keeping track of details about day-to-day courier senders, receivers, courier men, hub details using courier services. The basic work flow goes in the following manner: Login section will have login facility for the admin who operates the system and online tracking of consignment and shipping details for domestic shipping. While taking orders, it will take the details of sender and receiver. During billing process, system generates a Tracking ID for their products. Through this Tracking ID customers will able to track their couriers from any location using internet. The Proposed system will provide status of the product after placing orders and brings transparency to the system and help to save money.

1.2 PROJECT SPECIFICATION

In Proposed system, all the work is automated, and there is no wastage of time. In order to send a courier then the person can just login into the panel and send a request to the courier company and then the courier team can themselves contact you. In proposed Courier Management System, we need not waste time in any procedure and or in following any step, and all the work is done so easily without taking much time.

The system includes 4 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add courier categories, staff etc. Admin can view all the registered customers staff and delivery boy, can able to approve or reject users and also can able to view all registered customer details and courier list.

2. Customer Module

Customer can request courier pickup and do secure online payment. Customer can also add feedbacks and complaints regarding courier service.

3. Staff Module

Office staff can update courier status of customer order. So that, the Staff can study the plan details and requirements of the customer and they can update complaint status of customers. Staff can also apply leave online.

4. Courier boy Module

Delivery boy can login to the panel and view and update all couriers to deliver. Delivery boy can also apply leave online through their panel.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Existing Courier Service has no automation. All the work is done manually and has many flaws such as, in existing system the sender needs to first send the courier by going to the shop for sending the courier and then after some days the courier will be send when the courier man comes and receives the courier. Existing system takes a lot of time. Also the time taken is very much for sending even one courier. Thus existing system needs to be changed.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient in managing courier details including less transparency, no standardized packages.
- Courier often delayed with no progress visibility.
- Unskilled labour force and unmonitored quality of material and not all technology based.
- Human effort is needed.
- Customers do not get a chance to identify the quality of materials used

2.4 PROPOSED SYSTEM

In Proposed Courier Service, all the work is automated, and there is no wastage of time. In order to send a courier then the person can just login into the panel and send a request to the courier company and then the courier team can themselves contact you. In proposed Courier Management System, we need not waste time in any procedure and or in following any step, and all the work is done so easily without taking much time.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- **Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

- **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

- **Better service: -**

The system will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- Will the costs conduct a full system investigation?
- Will the cost of the hardware and software?
- Will the benefits in the form of reduced costs or fewer costly errors?

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, FASTO Courier Management System was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

FASTO Courier Management System, GUI is simple so that users can easily use it.

FASTO Courier Management System is simple enoughso that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel core i3
RAM	-	4 GB
Hard disk	-	1 TB

3.2.2 Software Specification

Front End	-	HTML, CSS
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:Hypertext Preprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**
  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, youneed a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**
  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL92" refers to the standard released in 1992, "SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open-Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams.

All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1 : Use case diagram for FASTO Courier Management System (Customer and Admin)

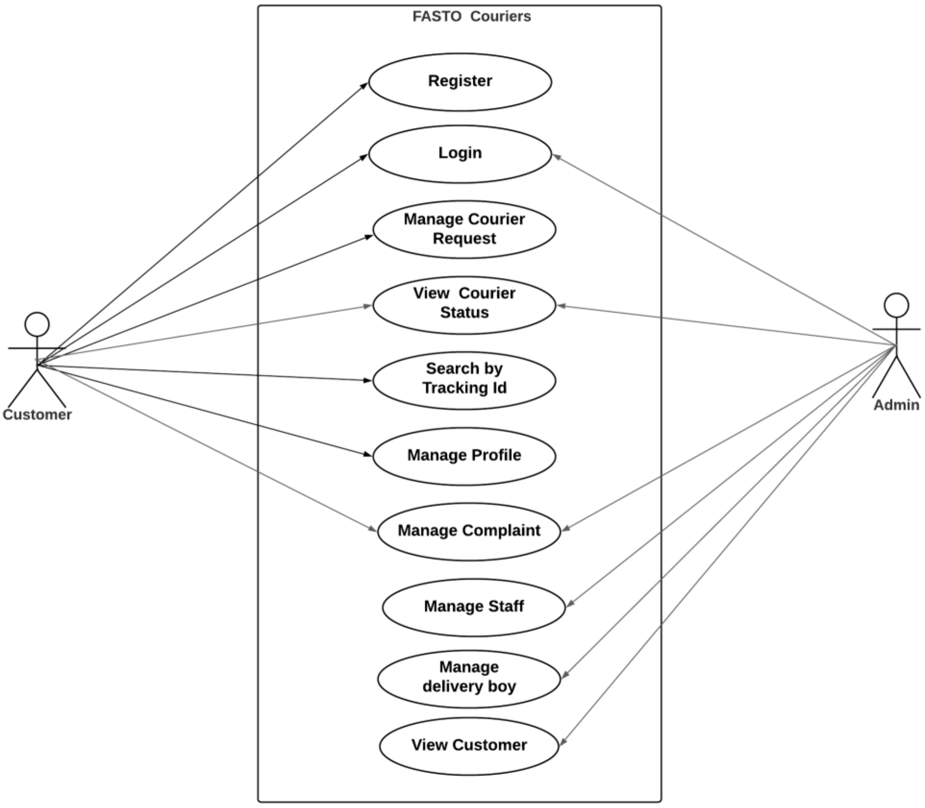
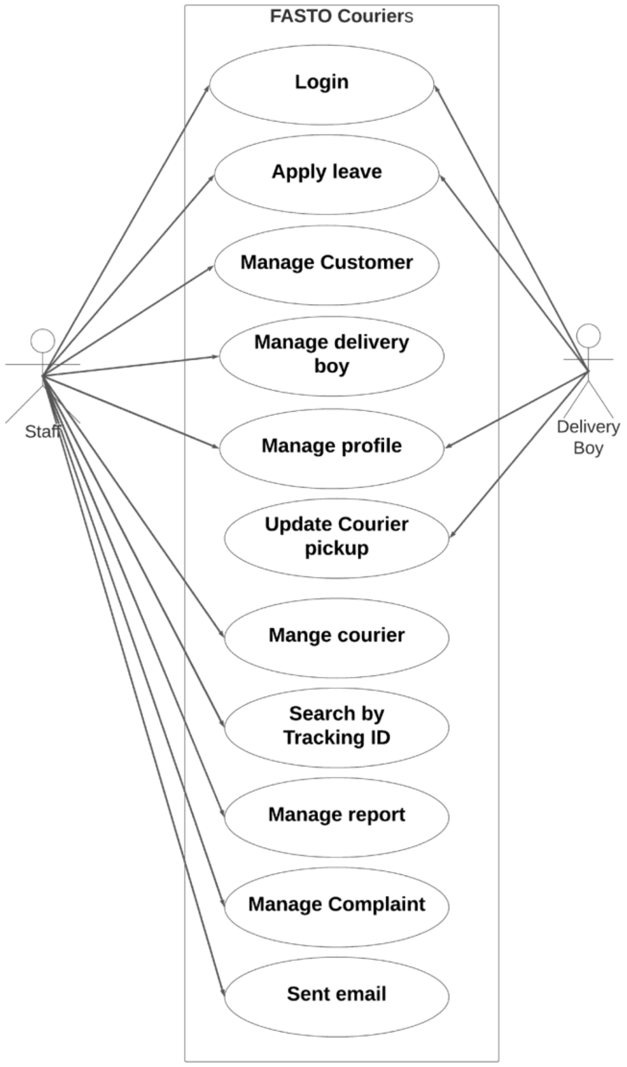


Fig 2 : Use case diagram for FASTO Courier Management System (Staff and Delivery Boy)



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

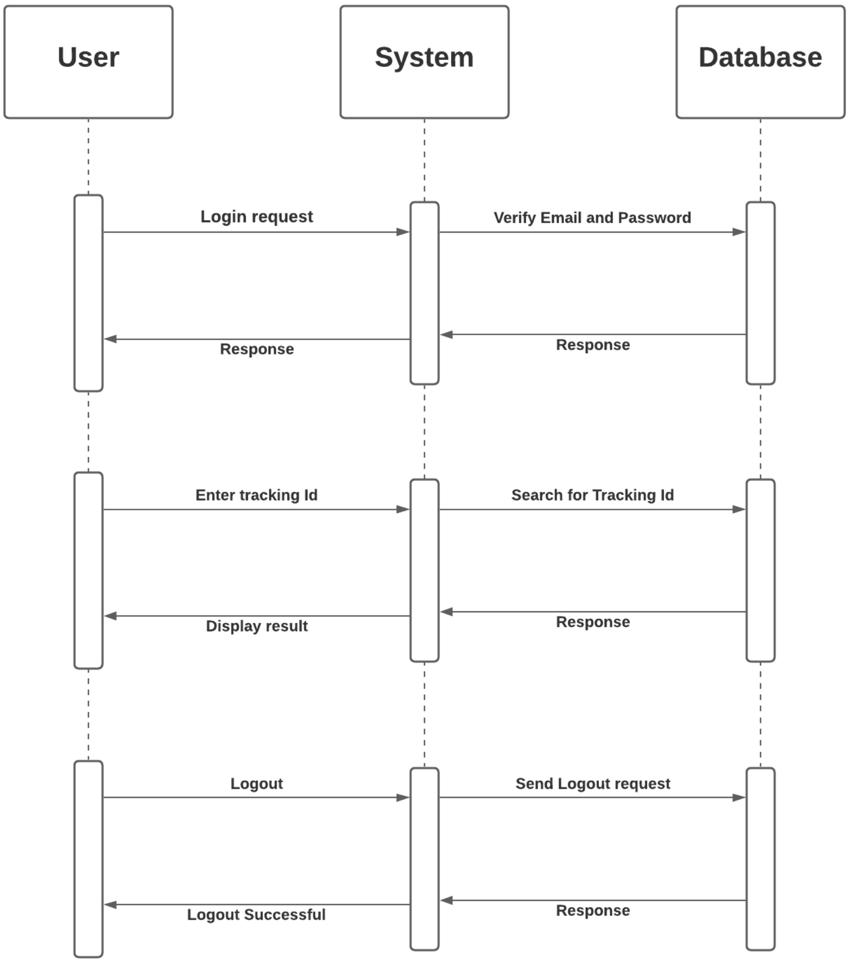
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message

iv. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Fig 3 : Sequence diagram for FASTO Courier Management System



4.2.3 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose of Component Diagrams-

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

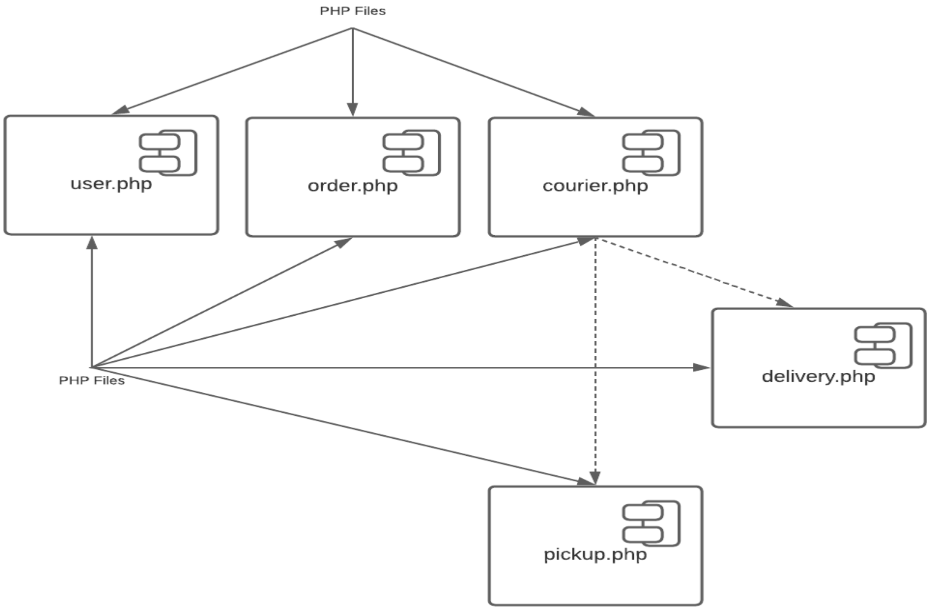
Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

The purpose of the component diagram can be summarized as −

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.

Fig 4: Component diagram for FASTO Courier Management System



4.2.4 ACTIVITY DIAGRAM

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

Purpose of Activity Diagrams-

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior 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.

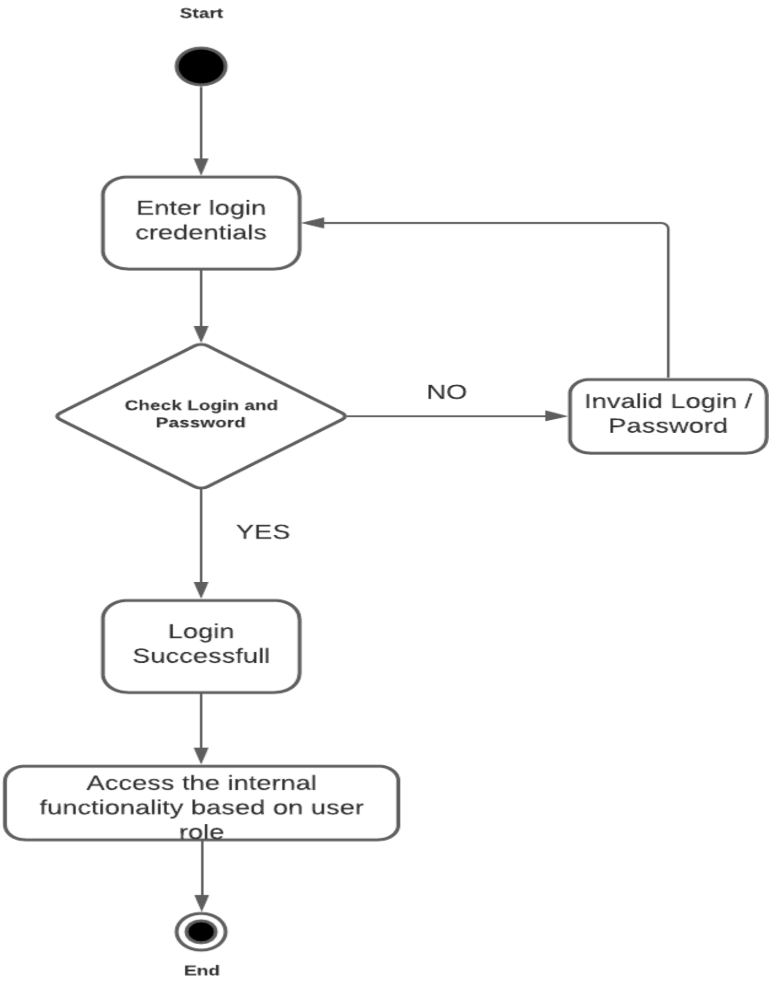
It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart.

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.

Fig 5: Activity diagram for FASTO Courier Management System



4.2.5 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.

Purpose of Class Diagrams

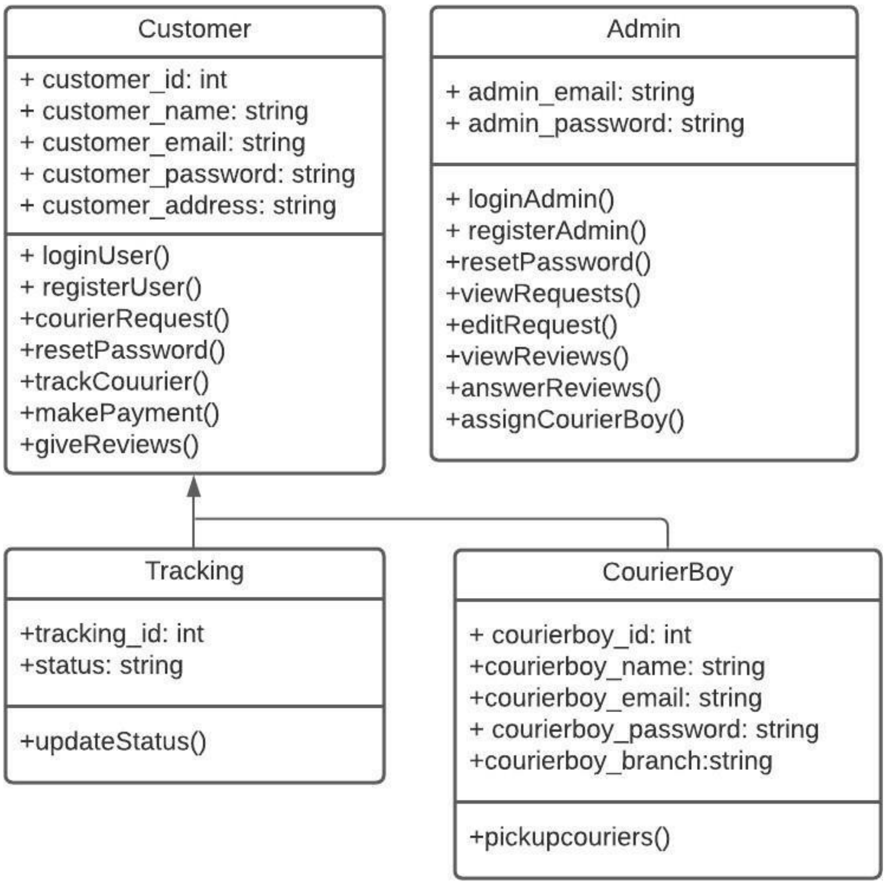
The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

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

Fig 6: Class diagram for FASTO Courier Management System



4.2.6 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

Notations of a collaboration diagram

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. The four major components of a collaboration diagram are:

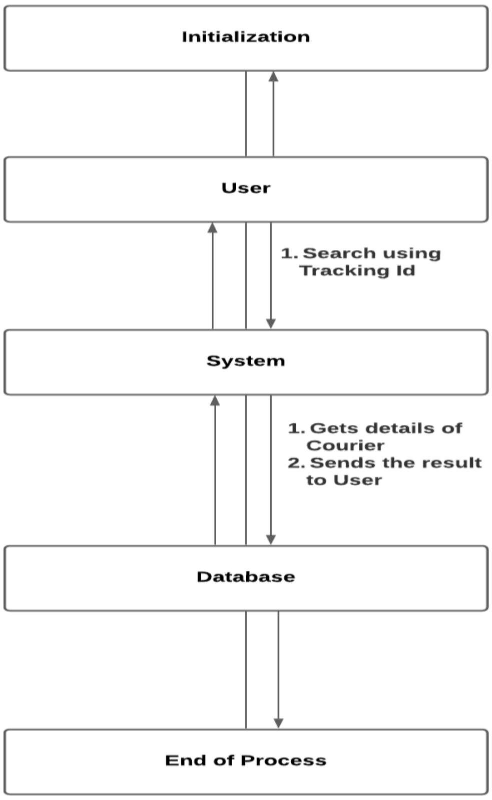
1. Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.

2. Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.

3. Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.

4. Messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

Fig 7: Collaboration diagram for FASTO Courier Management System


Initialization
User
1. Search using Tracking Id
System
1. Gets details of Courier
2. Sends the result to User
Database
End of Process

4.2.7 STATE CHART DIAGRAM

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Activity diagram explained in the next chapter, is a special kind of a State chart diagram. As State chart diagram defines the states, it is used to model the lifetime of an object.

Purpose of State chart Diagram

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events.

State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

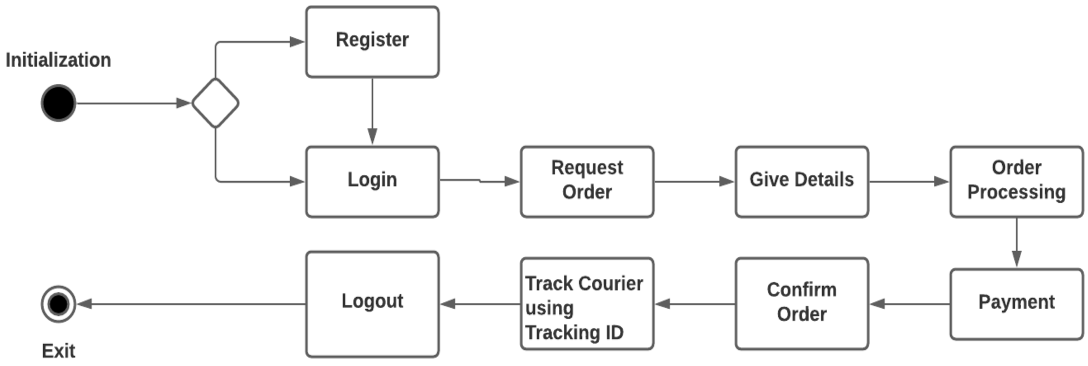
State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using State chart diagrams −

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

Fig 8: State chart diagram for FASTO Courier Management System



4.2.8 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

Purpose of Deployment Diagrams

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.

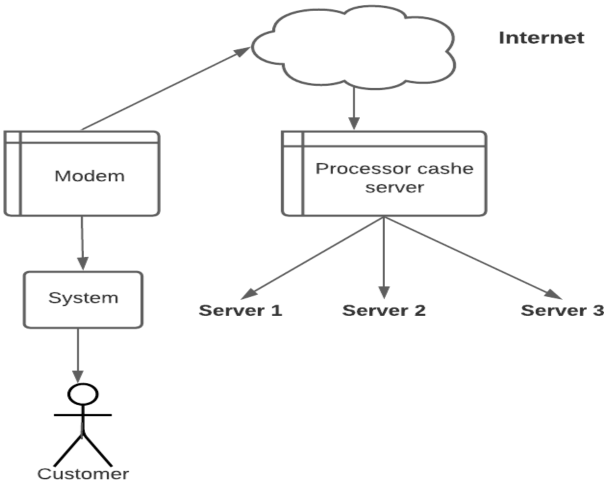
Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as −

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

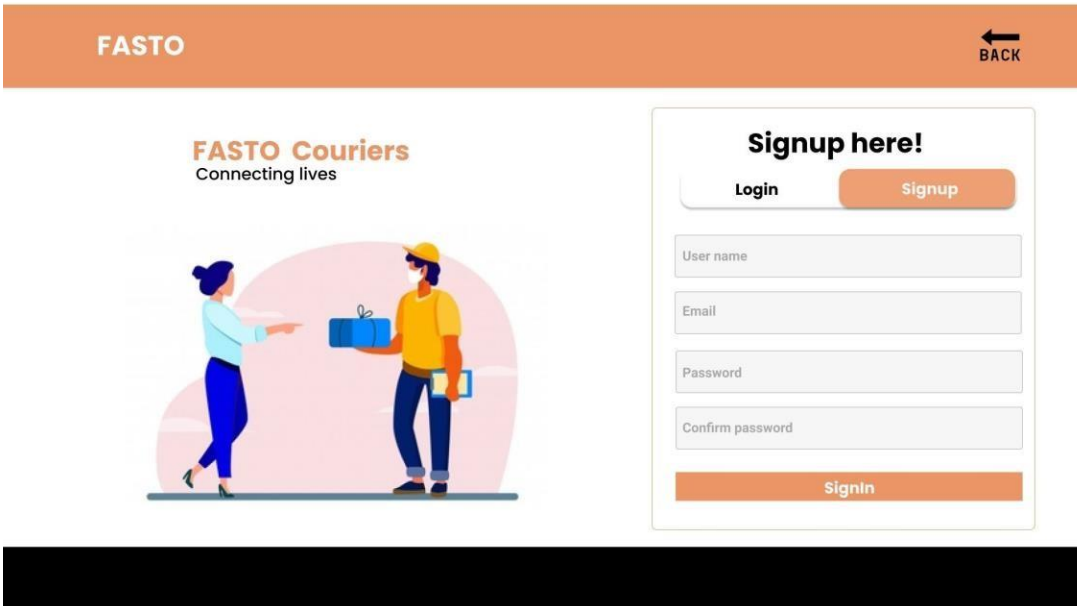
Fig 9: Deployment diagram for FASTO Courier Management System



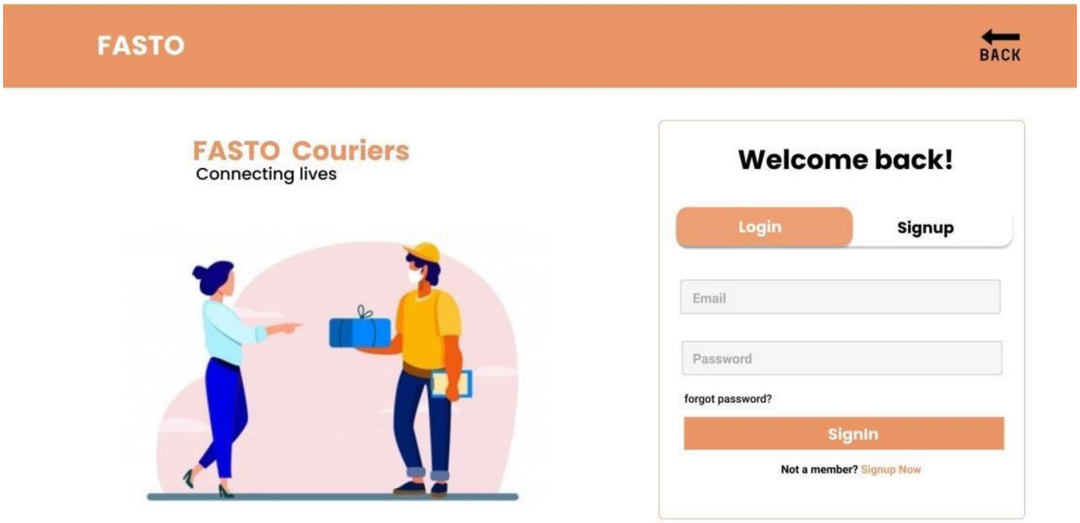
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1 INPUT DESIGN

Form Name : Customer Registration

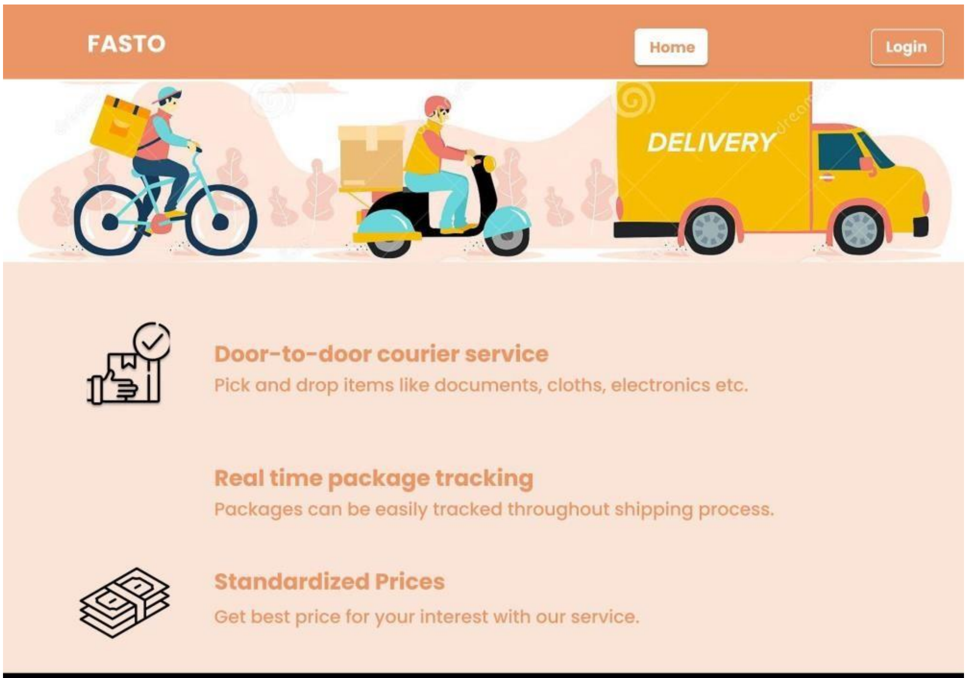


Form Name : Customer login



[redacted]

Form Name : Landing page



Door-to-door courier service

Pick and drop items like documents, cloths, electronics etc.

Real time package tracking

Packages can be easily tracked throughout shipping process.

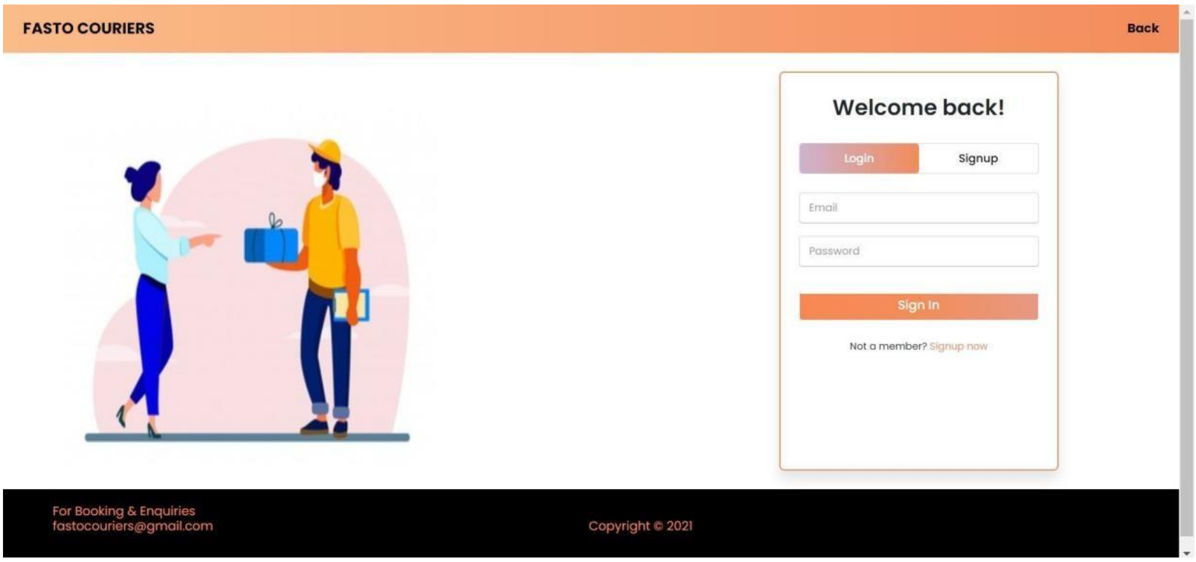
Standardized Prices

Get best price for your interest with our service.

[redacted]

4.3.2 OUTPUT DESIGN

Customer Login

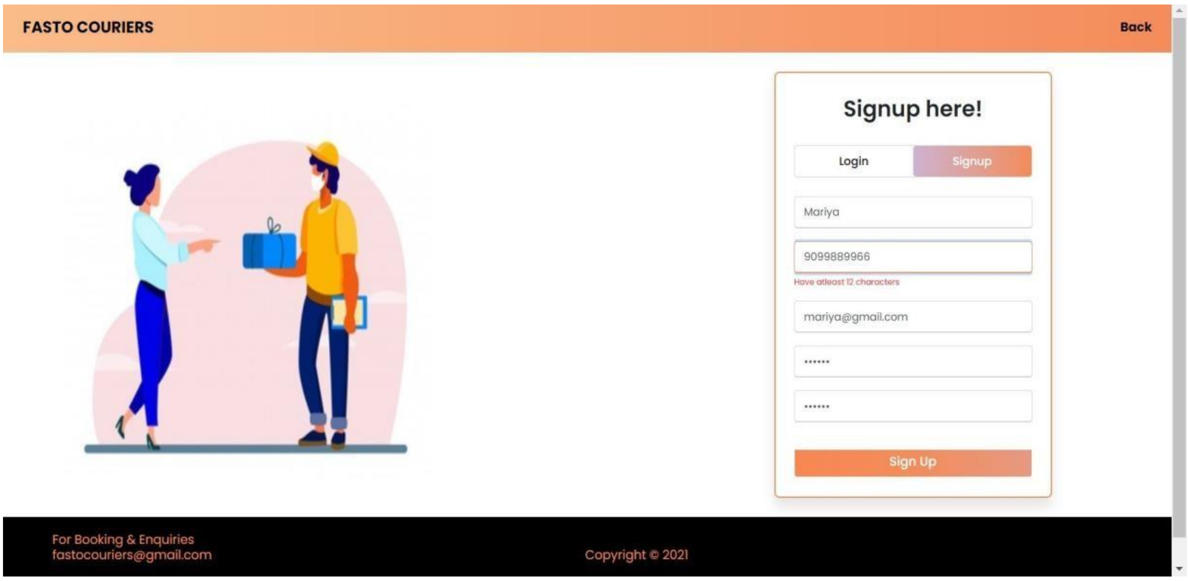


Not a member? Signup now

For Booking & Enquiries
fastocouriers@gmail.com

Copyright © 2021

Customer Registration



Mariya

9099889966

Have atleast 12 characters

mariya@gmail.com

••••••

••••••

For Booking & Enquiries
fastocouriers@gmail.com

Copyright © 2021

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of relatedvalues.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values

forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.
Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies.

Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

4.4.2.1 Normalize the data.

4.4.2.2 Choose proper names for the tables and columns.

4.4.2.3 Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normalform if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key of the table in the database.

TABLE DESIGN

Table No 01

Table Name : tbl_login

Primary Key : id

Foreign Key :

Table Description : To store user Login information

Fieldname	Data Type	Size	Description
id	Int	10	Primary key of login table
username	Varchar	30	Unique username to login
aadharid	Varchar	50	Unique Aadhar card number
email	Varchar	30	Email to login
password	Varchar	30	Password to login
role	Varchar	30	Designation of user
status	Varchar	60	Enable or Disable User

Table No 02

Table Name : tbl_ assigncourier

Primary Key : assign_id

Foreign Key : assign_courierid, assigned_courierboy

Table Description: To store assigned courier information

Fieldname	Data type	Size	Description
assign_id	Int	11	Primary key
assign_courierid	Int	30	Foreign key with reference from tbl_courier
assigned_courierboy	Int	30	Foreign key with reference from tbl_courierboy

Table No 03

Table Name : tbl_pickupdetails

Primary Key : pickup_id

Table Description: To store pickup details of courier

Fieldname	Data type	Size	Description
pickup_id	Int	50	Primary key
pickup_loc	Varchar	20	Pickup location
pickup_addr	Varchar	20	Pickup address
pickup_ins	Varchar	20	Pickup instruction
pickup_sender	Varchar	20	Senders name
pickup_addrtype	Varchar	20	Pickup address type (Home or Work)
pickup_mobile	Number	20	Pickup mobile number
pickup_date	Date		Pickup date
status	Varchar	100	Status of the courier pickup, To check whether courier is picked or not.

Table No 04

Table Name : tbl_deliverydetails

Primary Key : delivery_id

Table Description: To store delivery details of courier

Fieldname	**Date type**	**Size**	**Description**
delivery_id	Int	10	Primary Key
delivery_loc	Varchar	200	Delivery location
delivery_addr	Varchar	200	Delivery Address
delivery_ins	Varchar	200	Delivery Instruction
delivery_receiver	Number	100	Receivers name
delivery_addrtype	Number	100	Address type(Home or work)
delivery_mobile	Varchar	100	Delivery mobile number
Status	Varchar	10	Foreign key with reference from district table

Table No 05

Table Name : tbl_courier

Primary Key : courier_id

Foreign Key : pickup_id, delivery_id

Table Description: To store courier details

Fieldname	Data Type	Size	Description
courier_id	Int	10	Primary key
pickup_id	Int	20	Foreign key with reference from tbl_pickupdetails
delivery_id	Int	50	Foreign key with reference from tbl_deliverydetails
courier_image	Varchar	50	Image of courier
courier_cat	Varchar	20	Courier category
courier_weight	Int	100	Weight of courier
courier_price	Int	100	Courier Price
status	Varchar	100	Status of the Courier

Table No 06

Table Name : tbl_payment

Primary Key : pay_id

Foreign Key :

Table Description: To store payment for each courier

Fieldname	Data type	Size	Description
id	Int	11	Primary key
name	Varchar	50	Payee name
amount	Int	11	Amount Paid
payment_status	Varchar	50	Status of Payment
payment_id	Varchar	50	Payment Id
added_on	datetime		Date Payed

Table No 07

Table Name : tbl_staff

Primary Key : staff_id

Foreign Key : idnumber

Table Description: To store Staff details

Fieldname	Data type	Size	Description
Staff_id	Int	11	Primary key
idnumber	Int	11	Foreign key with reference from tbl_login
fullname	Varchar	100	Fullname of staff
phone	Int	50	Staff Phone number
dob	Varchar	50	Date of birth
gender	Varchar	50	Gender
doj	Varchar	50	Joining date

Table No 08

Table Name : tbl_courierboy

Primary Key : courierboy_id

Foreign Key :idnumber

Table Description: To store Courier boy details

Fieldname	Data type	Size	Description
courierboy_id	Int	11	Primary key
idnumber	Int	11	Foreign key with reference from tbl_login
fullname	Varchar	100	Full name of courier boy
phone	Int	50	Phone number
dob	Varchar	50	Date of Birth
gender	Varchar	50	Gender
doj	Varchar	50	Date of Joining

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide,important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

5.2.2 Selenium Automation Testing

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

A test performed using Selenium is usually referred to as Selenium automation testing. However, Selenium is not just a single tool but a collection of tools, each catering to different Selenium automation testing needs. In this tutorial you will learn all about Selenium and the various types of Selenium automation testing tools.

One disadvantage of Selenium automation testing is that it works only for web applications, which leaves desktop and mobile apps out in the cold. However, tools like Appium and HP’s QTP, among others, can be used to test software and mobile applications.

Selenium’s benefits are vast, and its drawbacks are few, but it’s essential to understand the framework’s drawbacks before incorporating it into your project.

- Automation of Windows-based applications is not possible.
- No in-built reporting mechanism.
- Selenium WebDriver does not support parallel testing.

5.2.2.1 Test Case for Login

Test Case 1	
Project Name: FASTO Courier Management System	
Login Test Case	
Test Case ID: Fun_1	**Test Designed By: Stephy Grace Oommen**
Test Priority (Low/Medium/High): High	**Test Designed Date:** 19-05-2022
Module Name: Login Screen	**Test Executed By: Ms. Gloriya Mathew**
Test Title: Verify login with valid email and password	**Test Execution Date:** 20-05-2022
Description: Test the Login Page	
Pre-Condition: User has valid email id and password	

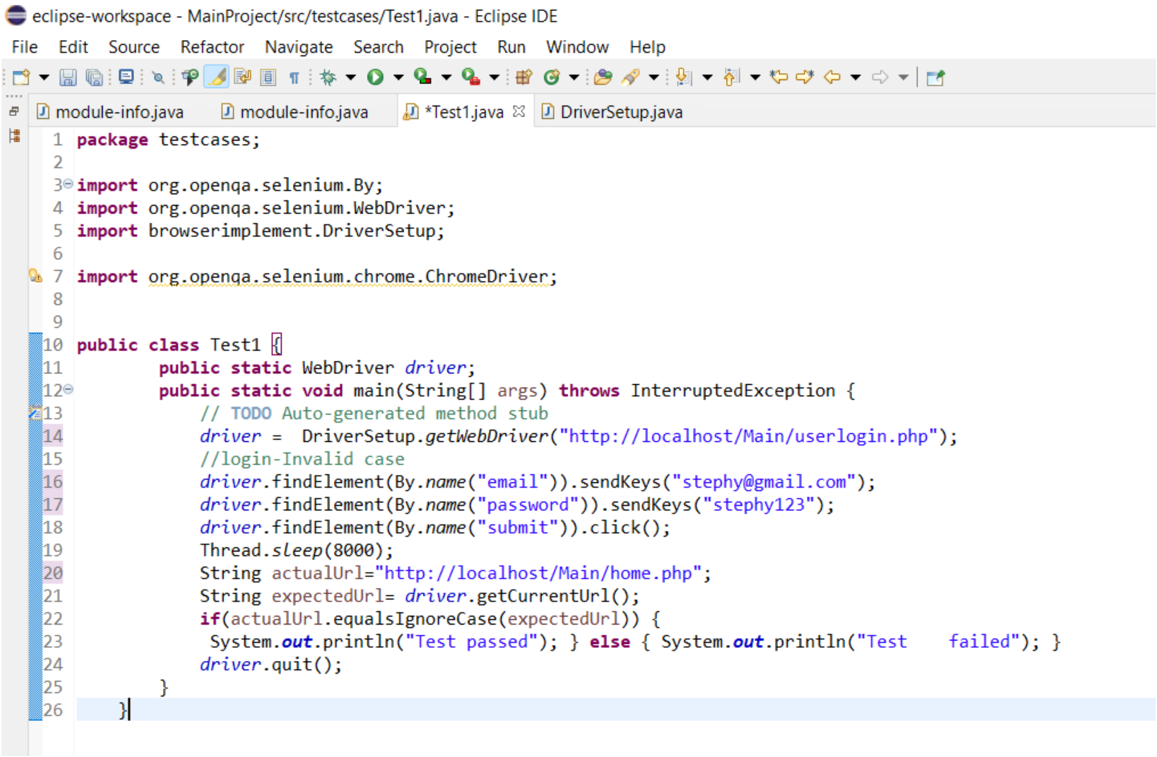
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displayed	Login page displayed	Pass
2	Provid eValid Email Id	User Name: stephy@gmail.com	User should beable to Login	User Logged in and navigated to Sub admin Dashboard with records	Pass
3	Provide Valid Password	Password: stephy123			

4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: telbin@gmail.Com Password: User12345	User should notbe able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

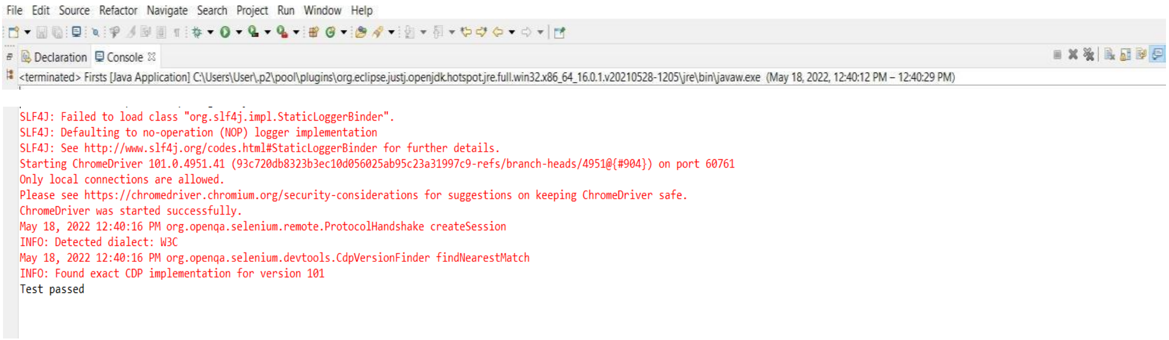
Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimplement.DriverSetup;
import org.openqa.selenium.chrome.ChromeDriver;
public class Test1 {
public static WebDriver driver;
public static void main(String[] args) throws InterruptedException {
// TODO Auto-generated method stub
driver =  DriverSetup.getWebDriver("http://localhost/Main/userlogin.php");
//login-Invalid case
driver.findElement(By.name("email")).sendKeys("stephy@gmail.com");
driver.findElement(By.name("password")).sendKeys("stephy123");
driver.findElement(By.name("submit")).click();
Thread.sleep(8000);
String actualUrl="http://localhost/Main/home.php";

String expectedUrl= driver.getCurrentUrl();

if(actualUrl.equalsIgnoreCase(expectedUrl)) {

System.out.println("Test passed"); } else { System.out.println("Test    failed"); }

driver.quit();

}

}
```

Output


eclipse-workspace - MainProject/src/testcases/Test1.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
module-info.java
module-info.java
*Test1.java
DriverSetup.java

```
package testcases;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import browserimplement.DriverSetup;

import org.openqa.selenium.chrome.ChromeDriver;


public class Test1 {
        public static WebDriver driver;
        public static void main(String[] args) throws InterruptedException {
            // TODO Auto-generated method stub
            driver =  DriverSetup.getWebDriver("http://localhost/Main/userlogin.php");
            //login-Invalid case
            driver.findElement(By.name("email")).sendKeys("stephy@gmail.com");
            driver.findElement(By.name("password")).sendKeys("stephy123");
            driver.findElement(By.name("submit")).click();
            Thread.sleep(8000);
            String actualUrl="http://localhost/Main/home.php";
            String expectedUrl= driver.getCurrentUrl();
            if(actualUrl.equalsIgnoreCase(expectedUrl)) {
             System.out.println("Test passed"); } else { System.out.println("Test    failed"); }
            driver.quit();
        }
    }
```


File Edit Source Refactor Navigate Search Project Run Window Help
Declaration
Console

```
<terminated> Firsts [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe  (May 18, 2022, 12:40:12 PM – 12:40:29 PM)

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

5.2.2.2 Test Case for Update Password

Test Case 2

Project Name: FASTO Courier Management System	
Update Password Test Case	
Test Case ID:updatePassword	**Test Designed By: Stephy Grace Oommen**
Test Priority(Low/Medium/High): High	**Test Designed Date:** 19-05-2022
Module Name:Login Screen	
Test Title : Update Password	**Test Execution Date:** 20-05-2022
Description: Login to system and update password, if some error occurs, test will fail	**Test Executed by: Ms. Gloriya Mathew**
Pre-Condition : User has valid user name and password	

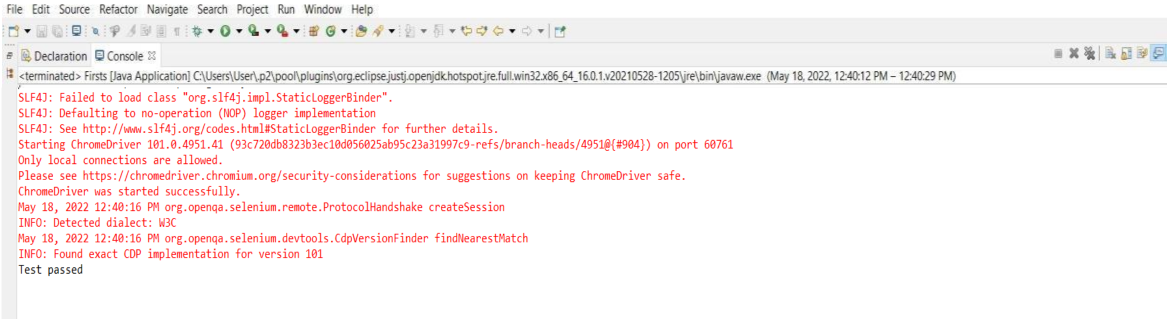
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to LoginPage		Login Page should be present	Login page displayed	Pass
2	Provide Valid Username	User Name: stephy@gmail.com	User should be ableto Login	User Logged in and navigated to user Dashboard.	Pass
3	Provide Valid Password	Password : stephy			
4	Click on Login button				
5	Provide house informations	Input password	User will be redirected to dashboard	User will be redirected to dashboard	Pass
7	Click on Update button				
8	Provide invalid informations	Input invalid password	User will be redirected to dashboard	User will be stay on that page showing error message	Pass
9	Click on Update button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;
```

```
public class UpdatePassword {
public static WebDriver driver;
public static void main(String[] args) {
driver =  DriverSetup.getWebDriver("http://localhost/MAIN/userlogin.php");
driver.findElement(By.name("login")).click();
driver.findElement(By.name("email")).sendKeys("stephy@gmail.com");
driver.findElement(By.name("password")).sendKeys("stephy");
driver.findElement(By.name("submit")).click();
driver.get("http://localhost/MAIN/userpasswordreset.php");

driver.findElement(By.name("email")).sendKeys("stephy@gmail.com");
driver.findElement(By.name("password")).sendKeys("stephy");
driver.findElement(By.name("cpassword")).sendKeys("stephy123");
driver.findElement(By.name("update")).click();

String actualUrl="http://localhost/MAIN/home.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed");
} else {
System.out.println("Test failed");
}
}
}
```

Output


File Edit Source Refactor Navigate Search Project Run Window Help
Declaration
Console

```
<terminated> Firsts [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe  (May 18, 2022, 12:40:12 PM – 12:40:29 PM)
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

5.2.3 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.4 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions thatwill fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.5 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply meansconvening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of theirrole in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system all work is automated and there is no wastage of time. Provides lots of advantages like Tracks courier, view profile of contractors, enhanced user interface, payment options, add feedback, up to date courier status, and complaint status and may more.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using html & css, usage of responsive templates, and management of database using mysql . The entire system is secured.

Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project. Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

7.2 FUTURE SCOPE

The system should be designed in such a way that the payment should be done in online mode and in offline mode and Customers can able to do advanced search options. Customers should be able to add complaints and feedbacks etc. Staff is able to view project details and add daily progress report etc. Data security can be enhanced in a much better way.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- Pankaj Jalote, “So*ftware engineering*: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- https://app.diagrams.net
- https://www.sourcecodester.com/php/14611/courier-management-system-using-phpmysqli-source-code.html
- https://www.tpcglobe.com/pickup.aspx
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Customer

sendcourier.php

```
<!--PHP CONNECTION & FETCH USER DETAILS-->
<?php
include('dbcon.php');
session_start();
$var=$_SESSION['name'];
$res=mysqli_query($con,"SELECT*FROM`tbl_login` whereusername='$var'");
while($r=mysqli_fetch_array($res))
{
$uname=$r['username'];  //username from table field & $uname is a random name
 }
if(isset($_SESSION["fastoSession"]) !=session_id()){
header("Location:index.php");
die();
}
else{
?>
<!DOCTYPEhtml>
<html>
<head>
<metacharset="UTF-8"/>
<metahttp-equiv="X-UA-Compatible"content="IE=edge"/>
<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css"integrity="sha512-
iBBXm8fW90+nuLcSKlbmrPcLa0OT92xO1BIsZ+ywDWZCvqsWgccV3gFoRBv0z+8
dLJgyAHIhR35VZc2oM/gI1w=="crossorigin="anonymous"/>
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<title>Send Courier</title>
<linkrel="icon"href="images/iconbox.png">
<linkrel="stylesheet"href="style.css">
<linkrel="stylesheet"href="css/header.css">
<linkrel="stylesheet"href="css/courier.css">
<scriptsrc="https://kit.fontawesome.com/a076d05399.js"></script>
<scriptsrc="js/sweetalert.js"></script>

  <!--<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">-->
<scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
</head>
<body>
```

```
 <nav>
<divclass="menu">
<divclass="logo">
<ahref="#">FastoCouriers</a>
</div>
<ul>
<li><ahref="home.php">Home</a></li>
<li><ahref="userprofile.php">Profile</a></li>
<li><ahref="logout.php">Logout</a></li>
<listyle="margin-left:100px;color:black"><b>
<iclass="fas fa-user-alt"></i>
<span><?phpecho$var;?></span></b></li>
</ul>
</div>
</nav>

<divclass="row">
<divclass="col-75">
<divclass="container">
<formaction="sendcourier.php"method="POST"enctype="multipart/form-data">
<divclass="row">
<!--PHP code for Pickup-->
<divclass="col-50">
<br><h3>Pickup</h3><br>
<labelfor="pickupname"><iclass="fa fa-user"></i>Senders Name</label>
          <inputtype="text"class="form-control
pickupname"name="pickupname"id="pickupname"placeholder="Enteryour
Name"required>
<divclass="row">
<divclass="col-50">
<labelfor="pickupdate">Pickup Date</label>
              <inputtype="date"name="pickupdate"id="pickupdate"placeholder="Prefered
pickup date"min="<?phpechodate('Y-m-d') ?>"required/>
</div>
</div>
<labelfor="pickupaddress"><iclass="fa fa-address-card-o"></i>Address</label>
<inputtype="text"id="pickupaddress"name="pickupaddress"placeholder="542
W. 15th Street"required>
          <labelfor="pickupinstructions"><iclass="fa              fa-institution"></i>Nearest
Landmark</label>
          <inputtype="text"class="form-control
pickupinstructions"id="pickupinstructions"name="pickupinstructions"placeholder="Lan
dmark"required>
<divclass="row">
<divclass="col-50">
<labelfor="pickup">Address Type</label>

<selectname="pickupaddrtype"required>
<optionvalue=""disabledselected>Choose</option>
<optionvalue="Home">Home(Anytime)</option>
```

```
<optionvalue="Office">Office(Between 9am-6pm)</option>
</select>
</div>
</div>
</div>
<!--PHP code for Delivery-->
<divclass="col-50">
<br><h3>Delivery</h3><br>
<labelfor="deliveryname"><iclass="fa fa-user"></i>Receivers Name</label>
        <inputtype="text"class="form-control deliveryname"
name="deliveryname"id="deliveryname"placeholder="Enter Recepient Name"required>
<divclass="row">
<divclass="col-50">
<labelfor="deliverycontact">Mobile</label>
          <inputtype="text"class="form-control
deliverycontact"name="deliverycontact"id="deliverycontact"placeholder="Recepient
Phone number"required/>
</div>
</div>
<labelfor="deliveryaddress">Address</label>

<inputtype="text"name="deliveryaddress"id="deliveryaddress"placeholder="Enter
delivery address"required>
<divclass="row">
<divclass="col-50">
<labelfor="deliveryinstructions">Nearest Landmark</label>
          <inputtype="text"class="form-control
deliveryinstructions"name="deliveryinstructions"id="deliveryinstructions"placeholder="
Nearest landmark"required/>
</div>
<divclass="col-50">
<labelfor="delivery">Address Type</label>
<selectname="deliveryaddrtype"required>
<optionvalue=""disabledselected>Choose</option>
<optionvalue="Home">Home(Anytime)</option>
<optionvalue="Office">Office(Between 9am-6pm)</option>
</select>
</div>
</div>
</div>
<!--PHP code Package-->
<divclass="col-50">
<br><h3>Package</h3><br>
<divclass="row">
<divclass="col-50">
<labelfor="weight"><b>Package Weight</b></label>

<selectname="packageweight"onchange="calculateAmount(this.value)"required>
<optionvalue=""disabledselected>Choose your option</option>
<optionvalue="1">Below 5Kg</option>
```

```
<optionvalue="2">6-9Kg</option>
<optionvalue="3">10-15Kg</option>
</select>
</div>
<divclass="col-50">
<label><b>Amount</b></label>
<inputtype="text" name="tot_amount"id="tot_amount"readonly>
</div>
</div>
<labelfor="packagesize"><b>Upload Package Image</b></label>
<label class="uploadimage"for="upload">Choose file</label>
        <inputtype="file"id="upload"name="upload"accept="image/x-
png,image/jpeg"hidden/>
<br>
<labelfor="packagesize">Courier Category</label>
        <inputtype="checkbox"id="checkbox-
1"name="category"value="Document" />
<labelfor="checkbox-1">Documents</label>
        <inputtype="checkbox"id="checkbox-
2"name="category"value="Medical"/>
<labelfor="checkbox-2">Medical</label>
        <inputtype="checkbox"id="checkbox-
3"name="category"value="Fragile"/>
<labelfor="checkbox-3">Fragile</label>
        <inputtype="checkbox"id="checkbox-
4"name="category"value="Non-Fragile"/>
<labelfor="checkbox-4">Non-Fragile</label>
</div>
</div>
<inputtype="submit"value="Pickup Now"name="order" class="btn">
</form>

<!--PHP code for Pickup, Delivery, Package-->
<?php
if(isset($_POST['order']))
{
$pickuplocation=$_POST['pickuplocation'];
$pickupaddress=$_POST['pickupaddress'];
$pickupinstructions=$_POST['pickupinstructions'];
$pickupname=$_POST['pickupname'];

$pickupcontact=$_POST['pickupcontact'];
$pickupaddrtype=$_POST['pickupaddrtype'];
$pickupdate=$_POST['pickupdate'];
$deliverylocation=$_POST['deliverylocation'];
$deliveryaddress=$_POST['deliveryaddress'];


$deliveryinstructions=$_POST['deliveryinstructions'];
$deliveryname=$_POST['deliveryname'];
```

```
        $deliverycontact=$_POST['deliverycontact'];
        $deliveryaddrtype=$_POST['deliveryaddrtype'];

        $category=$_POST['category'];
        $packageweight=$_POST['packageweight'];
        $tot_amount=$_POST['tot_amount'];
        $upload=$_FILES['upload']['name'];

       move_uploaded_file($_FILES['upload']['tmp_name'],"images/".$_FILES['upl
          oad']['name']);

     $query1="INSERT  INTO`tbl_pickupdetails`(`pickup_loc`,  `pickup_addr`,
   `pickup_ins`, `pickup_sender`, `pickup_addrtype`, `pickup_mobile`, `pickup_date`,
   `status`)
    VALUES('$pickuplocation','$pickupaddress','$pickupinstructions','$pickupname','$
    pickupaddrtype','$pickupcontact','$pickupdate','Pending')";
        $query_run1=mysqli_query($con,$query1);
   $respickupdetails=mysqli_query($con,"SELECTpickup_id
    FROM`tbl_pickupdetails`");
        while($r1=mysqli_fetch_array($respickupdetails))
         {
         $pickup_id=$r1['pickup_id'];
         }
   $query2="INSERT        INTO`tbl_deliverydetails`(`delivery_loc`,
 `delivery_addr`,                    `delivery_ins`,              `delivery_receiver`,
              `delivery_addrtype`,
`delivery_mobile`,                                                                `status`)
 VALUES('$deliverylocation','$deliveryaddress','$deliveryinstructions','$deliveryna
    me','$deliveryaddrtype','$deliverycontact','Pending')";
        $query_run2=mysqli_query($con,$query2);
$resdeliverydetails=mysqli_query($con,"SELECTdelivery_id
    FROM`tbl_deliverydetails`");
        while($r2=mysqli_fetch_array($resdeliverydetails))
         {
         $delivery_id=$r2['delivery_id'];
         }
        $query3="INSERT                INTO`tbl_courier`(`pickup_id`,        `delivery_id`,
 `courier_image`, `courier_cat`, `courier_weight`, `courier_price`, `status`)
    VALUES('$pickup_id','$delivery_id','$upload','$category','$packageweight','$tot_a
    mount','Pending')";
        $query_run3=mysqli_query($con,$query3);
            {

         $_SESSION['status']="Place Your Order";
         $_SESSION['status_code']="success";
 /echo       '<script      type="text/javascript">alert("User       Rgistered
    Successfully!!!Go To Login..!!")</script>';
         }
  else
         {
            $_SESSION['status']="Unsuccessfull";
                  //echo
                  '<script       type="text/javascript">alert("ERROR!!Try
                  Again!!")</script>';
           }
        }
        ?>
        <!--PHP code for sweet alert-->
        <?php
        if(isset($_SESSION['status']) &&$_SESSION['status'] !='')
        {
        ?>
        <script>
        swal({
        title:"<?phpecho$_SESSION['status']; ?>",
        text:"Pay Now",
        icon:"<?phpecho$_SESSION['status_code']; ?>",
        type:"success"
        }).then(function(){
        //redirect to another page
        window.location.href="payment.php";
        });
        </script>
        <?php
        unset($_SESSION['status']);
        }
        ?>
        </div>
      </div>
    </div>

    <!--JS FOR CALCULATIONS-->
    <script>
    functioncalculateAmount(val) { //Character Validation
    vartot_price=val*100;
    vardivobj=document.getElementById('tot_amount');
    divobj.value=tot_price;
    }
    $('.pickupname,.deliveryname').keyup(function() { //Character Validation
    $('span.error-keyup-2').remove();
    varinputVal=$(this).val();
    varcharacterReg=/^\s*[a-zA-Z,\s]+\s*$/;
    if(!characterReg.test(inputVal)) {
    }
    });
    $('.deliveryinstructions,.pickupinstructions').keyup(function()  {  //Character
    Validation
    $('span.error-keyup-2').remove();varinputVal=$(this).val();
```

```
varcharacterReg=/^\s*[a-zA-Z,\s]+\s*$/;
if(!characterReg.test(inputVal)) {
   $(this).after('<span class="error error-keyup-2" style="color:red">Enter
valid character input</span>');
}
});
$('.deliverycontact,.pickupcontact').keyup(function() { //Phone number
validation
$('span.error-keyup-2').remove();
varinputVal=$(this).val();
varphoneno=/^\(?([0-9]{3})\)?[-. ]?([0-9]{3})[-. ]?([0-9]{4})$/;
if(!phoneno.test(inputVal)) {
$(this).after('<span class="error error-keyup-2" style="color:red"> Number
must be 10 digit </span>');

}
});
</script>
</body>
</html>
<?php
}
?>
```

viewassignedcouriers.php

```
<!--PHP CONNECTION & FETCH STAFF DETAILS-->
<?php
include('dbcon.php');
session_start();
$var=$_SESSION['name'];
$role="Courier Boy";
$res=mysqli_query($con,"SELECT*FROM`tbl_login`
whereusername='$var' androle='$role'");
while($r=mysqli_fetch_array($res))
{
$uname=$r['username'];
$uid=$r['id'];  //username from table field & $uname is a random name
}
if(isset($_SESSION["fastoSession"]) !=session_id()){
   header("Location:index.php");
   die();
}
else{
?>

<!DOCTYPEhtml>
<htmllang="en">
```

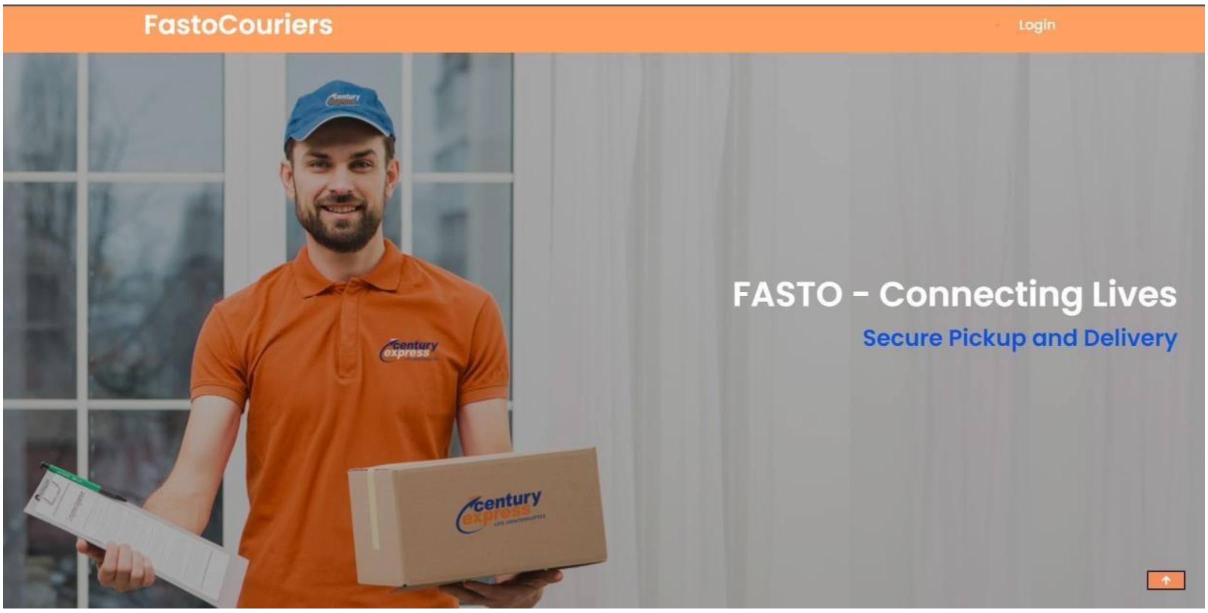
```
<head>
<metacharset="UTF-8"/>
<metahttp-equiv="X-UA-Compatible"content="IE=edge"/>
   <metaname="viewport"content="width=device-width, initial-
scale=1.0"/>
   <linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css"integrity="sha512-
iBBXm8fW90+nuLcSKlbmrPcLa0OT92xO1BIsZ+ywDWZCvqsWgccV3g
FoRBv0z+8dLJgyAHIhR35VZc2oM/gI1w=="crossorigin="anonymous"/>
<linkrel="stylesheet"href="dashboard.css"/>
<scriptsrc="js/sweetalert.js"></script>
<title>Courier List</title>
<linkrel="icon"href="images/iconbox.png">


</head>
<!--BODY BEGINS-->
<body>
<headerid="header">
<iclass="fas fa-bars"id="nav-toggler"></i>
<div>
<iclass="fas fa-user-alt"></i>
<span><b><?phpecho$role;?></b></span>
              </div>
            </header>
         <!--Header-->
  <divclass="top_navbar">
            <divclass="logo">FastoCouriers</div>
  </div>

<divclass="sidebar">
<divclass="sidebar_inner">
<ul>
<li>
<ahref="logout.php">
<spanclass="icon"><iclass="fa
<spanclass="text">Logout</span>
</a>
</li>
</ul>
</div>
</div>
<!--TABLE TO VIEW ALL CUSTOMERS-->
<sectionclass="viewtable">
<center>
<h1style="margin-top:2%"><b></b></h1>
<tableid="viewtableid"style="margin-top:3%; margin-left:10%">
<tr>
```

```
    <th>&nbsp;&nbsp;&nbsp;&nbsp;Courier</th>
    <th>&nbsp;&nbsp;&nbsp;Category</th>
    <th>&nbsp;&nbsp;Status</th>
    <thcolspan="2">&nbsp;&nbsp;&nbsp;Action</th>
    </tr>

    <!--PHP CONNECTION & FETCH COURIER DETAILS-->
    <?php
    $rr=mysqli_query($con,"SELECT`assign_courierid`
FROM`tbl_assigncourier`      WHERE`assigned_courierboy`='$uid'");
    while($rc=mysqli_fetch_array($rr))
    {
    $ascid=$rc["assign_courierid"];
    $res=mysqli_query($con,"SELECT`courier_id`,
`pickup_id`,`courier_image`, `courier_cat`,
`courier_weight`, `courier_price`, `status` FROM`tbl_courier`

WHERE`courier_id`='$ascid'");
    while($r=mysqli_fetch_array($res))
    {
    $couid=$r["courier_id"];
    $pickid=$r["pickup_id"];
    $courier_image=$r["courier_image"];


$rcc=mysqli_query($con,"SELECT`pickup_id`, `pickup_date`, `pickup_loc`,
`pickup_addr`,     `pickup_ins`,     `pickup_sender`,     `pickup_addrtype`,
`pickup_mobile`   FROM`tbl_pickupdetails`   WHERE`pickup_id`='$pickid'
AND`status`='Pending'");
    while($rrc=mysqli_fetch_array($rcc))
    {
    $pick_dt=$rrc["pickup_date"];
    $pick_sed=$rrc["pickup_sender"];
    $pick_loc=$rrc["pickup_loc"];
    $pick_addr=$rrc["pickup_addr"];
    $pick_ins=$rrc["pickup_ins"];
    $pick_addty=$rrc["pickup_addrtype"];
    $pick_ph=$rrc["pickup_mobile"];
    ?>
<formaction="viewassignedcouriers.php?action=idd&id=<?phpecho$couid;?
>"method="POST">
<td><imgsrc="images/<?phpecho$courier_image;?>"height="100"width="1
00"/></td>
        <td><b><?phpecho$courier_cat;?></b></td>
        <td><b><?php

if($courier_weight=='1'){
            echo"Below 5 Kg";
        }
        elseif($courier_weight=='2'){
            echo"6-9 Kg";
        }
        else{
            echo"10-15 Kg";
        }
        ?></b></td>
        <td>
        <bstyle="color:green;">Rs.<?phpecho$courier_price;?>/-</b>
        </td>
        <td><b><?phpecho$pick_dt;?></b></td>
        <td><b><?phpecho$pick_sed;?></b></td>
        <td><b><?phpecho$pick_loc;?></b></td>
        <td><b><?phpecho$pick_addr;?></b></td>
        <td><b><?phpecho$pick_ins;?></b></td>
<td><inputtype="submit"class="button-
assignapprove"name="applybtn"value="Apply"/></td>
    </tr>
    </form>
    <?php
    }}}
    ?>
    <!--PHP code for fetching idnumber from tbl_courierbot and insert it into
assigned_cboy_id-->
    <?php
    if(isset($_GET['action'])=="idd")
    {
    $couid=$_GET['id'];
    $sql4="SELECT`idnumber` FROM`tbl_courierboy` where`idnumber` ='$couid'";
    $row4=mysqli_query($con, $sql4);
    if($res4=mysqli_fetch_assoc($row4))
    {
    $boyID=$_POST['courierboyname'];
    }
    }
    ?>
    <!--PHP code for sweet alert-->
    <?php
    if(isset($_SESSION['status']) &&$_SESSION['status'] !=''){
    ?>
    <script>
    swal({
    title:"<?phpecho$_SESSION['status']; ?>",
    //text: "You clicked the button!",

    icon:"<?phpecho$_SESSION['status_code']; ?>",
    button:"Ok",
    });
    </script>
    <?php
    unset($_SESSION['status']);
    }
    ?>
    </table>
    </center>
</section>
</body>
</html>
<?php
}
?>
```

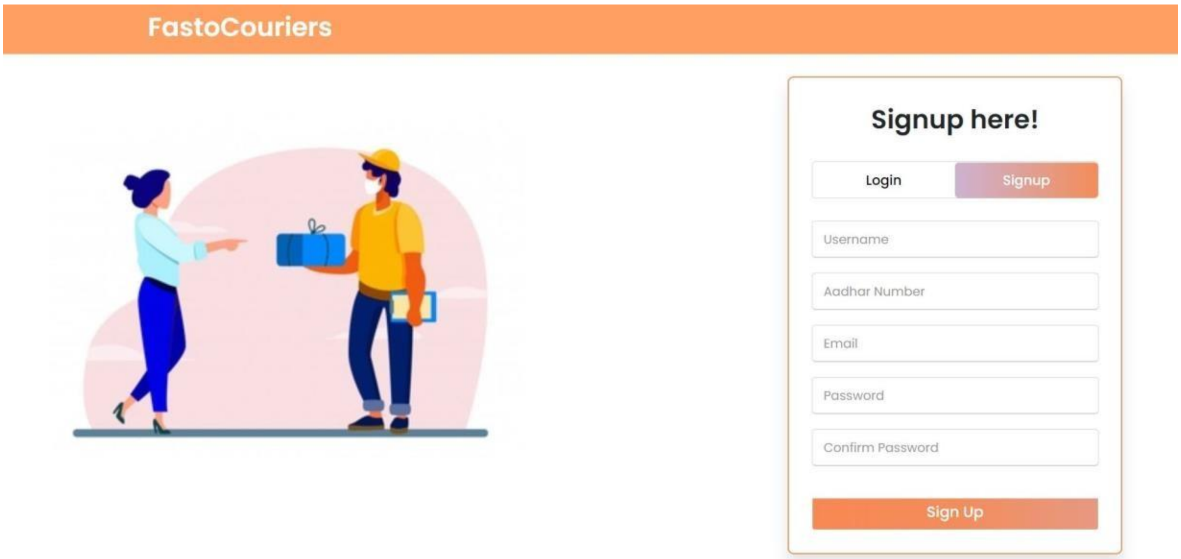
9.2 Screen Shots

CUSTOMER PAGES

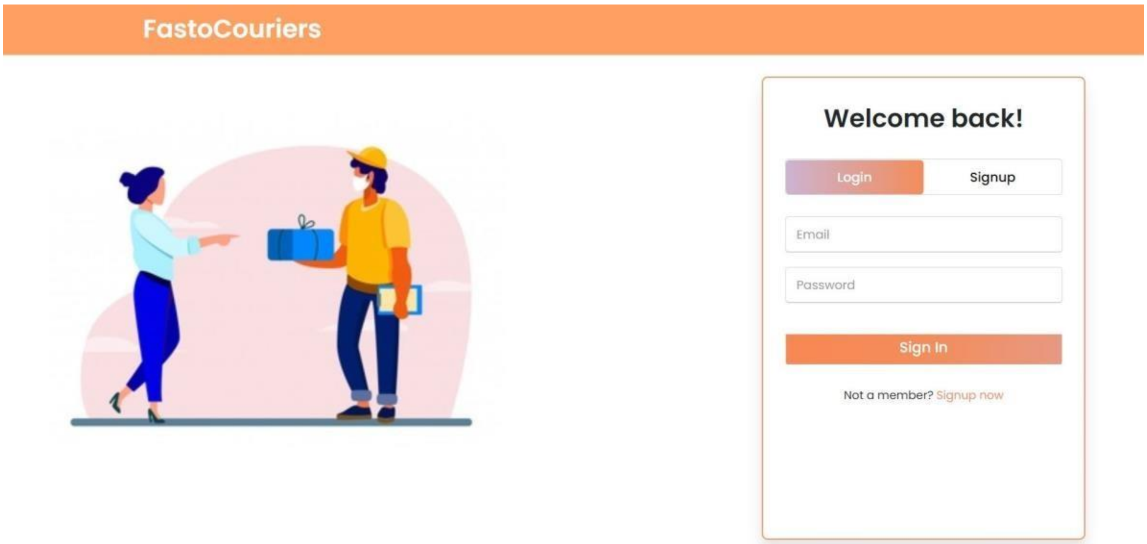
Landing page



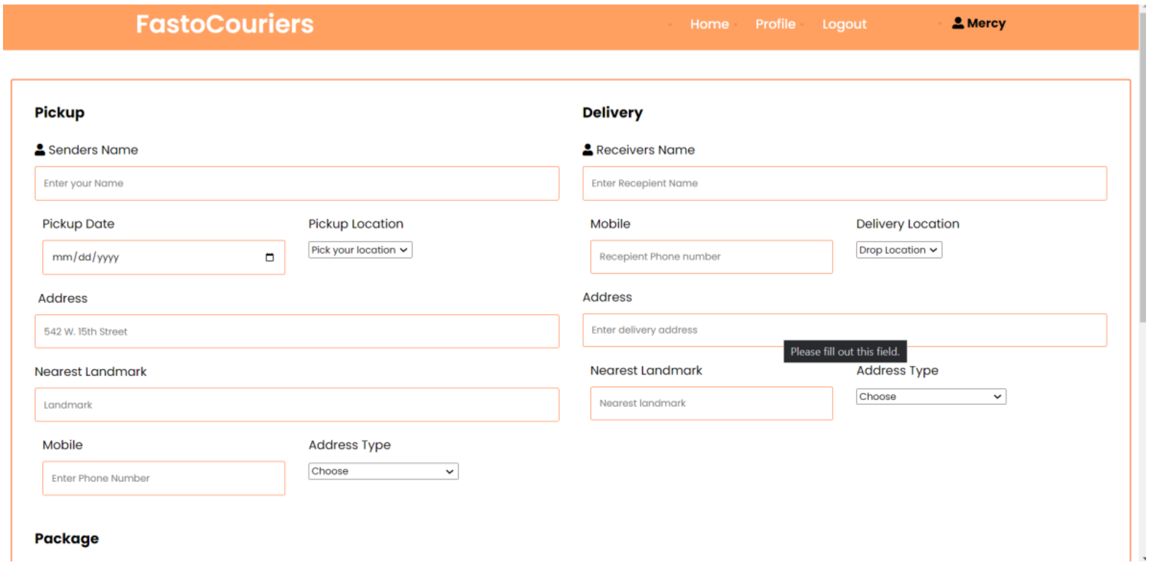
Customer Registration page



Customer Login page



Request Courier pickup



Pickup

Senders Name: Enter your Name

Pickup Date: mm/dd/yyyy

Pickup Location: Pick your location

Address: 542 W. 15th Street

Nearest Landmark: Landmark

Mobile: Enter Phone Number

Address Type: Choose

Package

Delivery

Receivers Name: Enter Recepient Name

Mobile: Recepient Phone number

Delivery Location: Drop Location

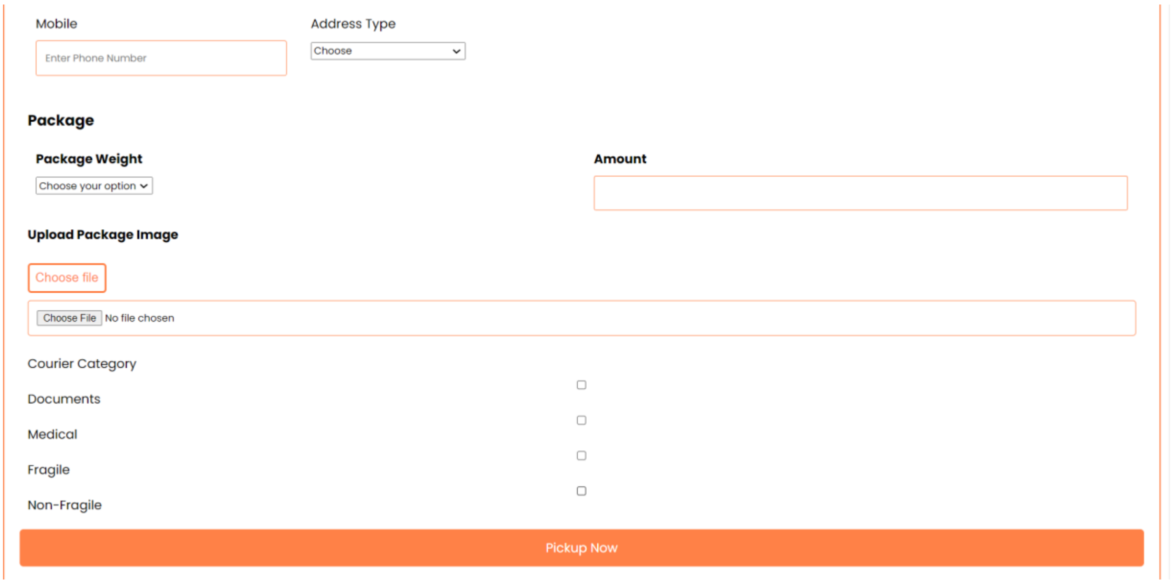
Address: Enter delivery address

Please fill out this field.

Nearest Landmark: Nearest landmark

Address Type: Choose

Request Courier pickup



Mobile

Enter Phone Number

Address Type

Choose

Package

Package Weight

Choose your option

Amount

Upload Package Image

Choose file

Choose File No file chosen

Courier Category

Documents □

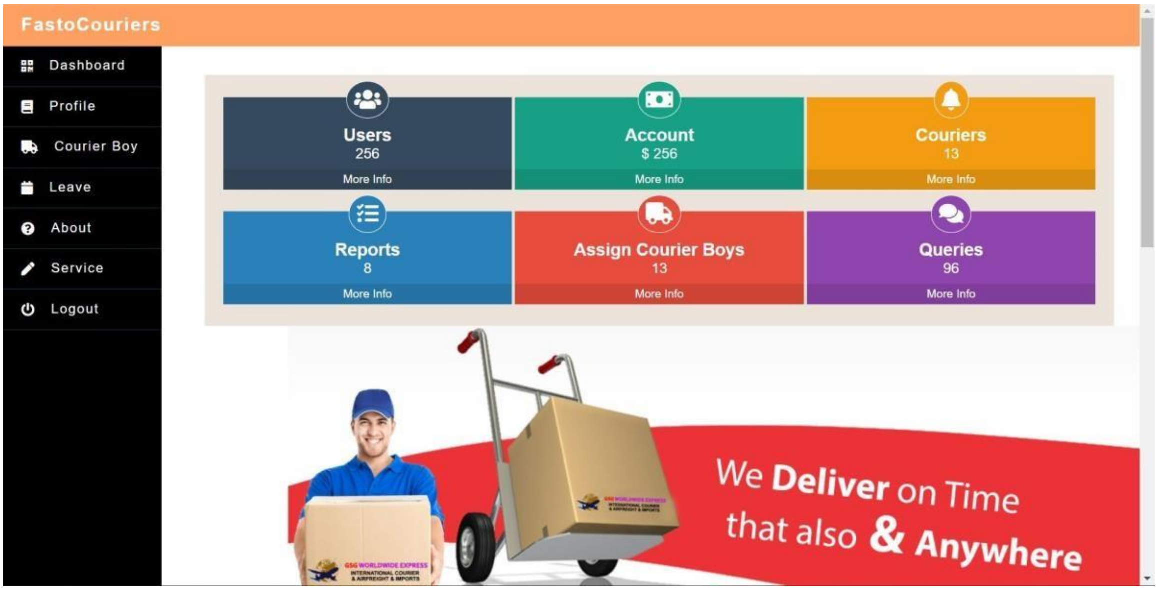
Medical □

Fragile □

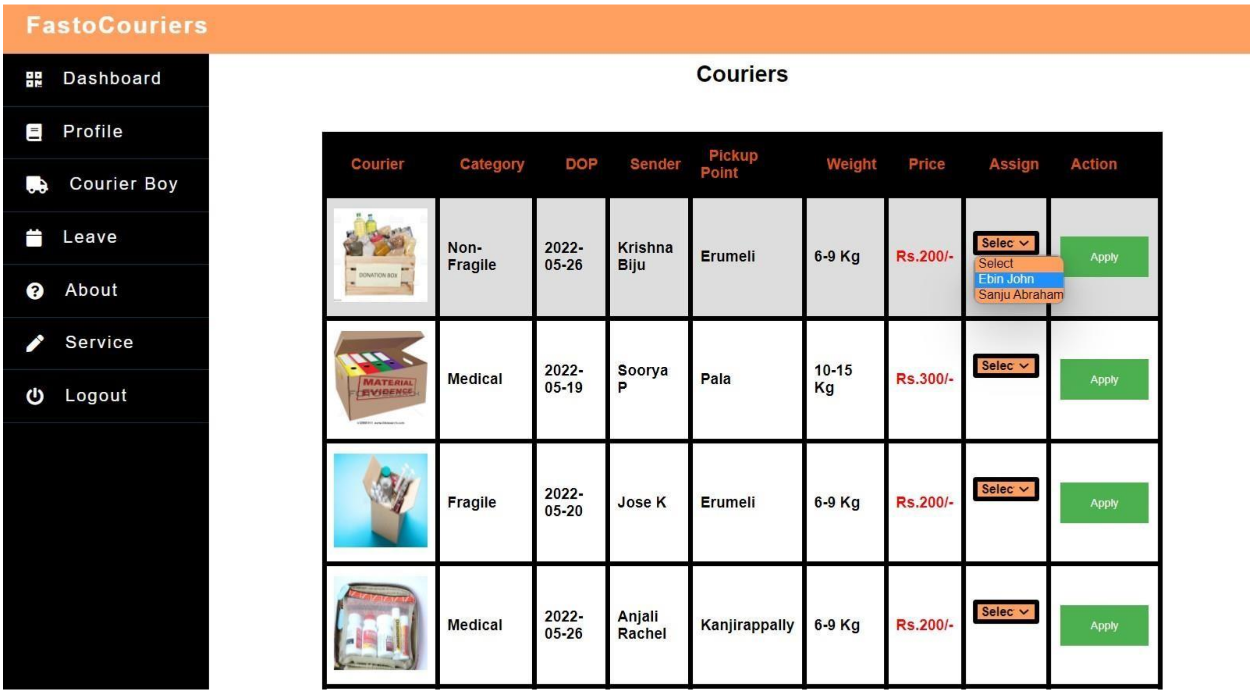
Non-Fragile □

STAFF PAGES

Dashboard



Staff Manage Courier

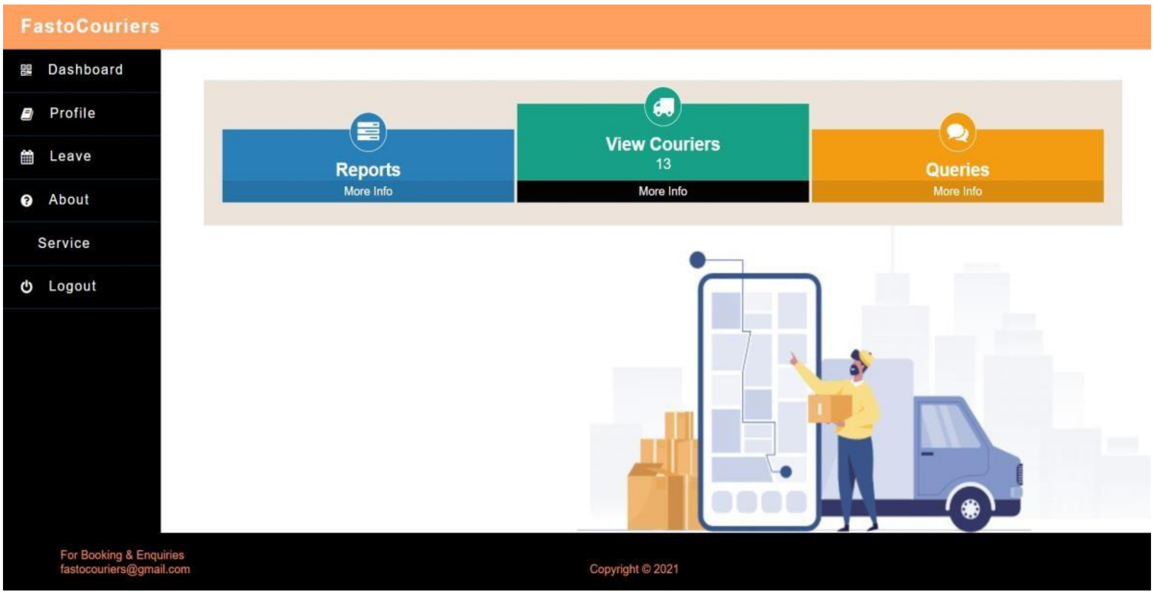


Couriers

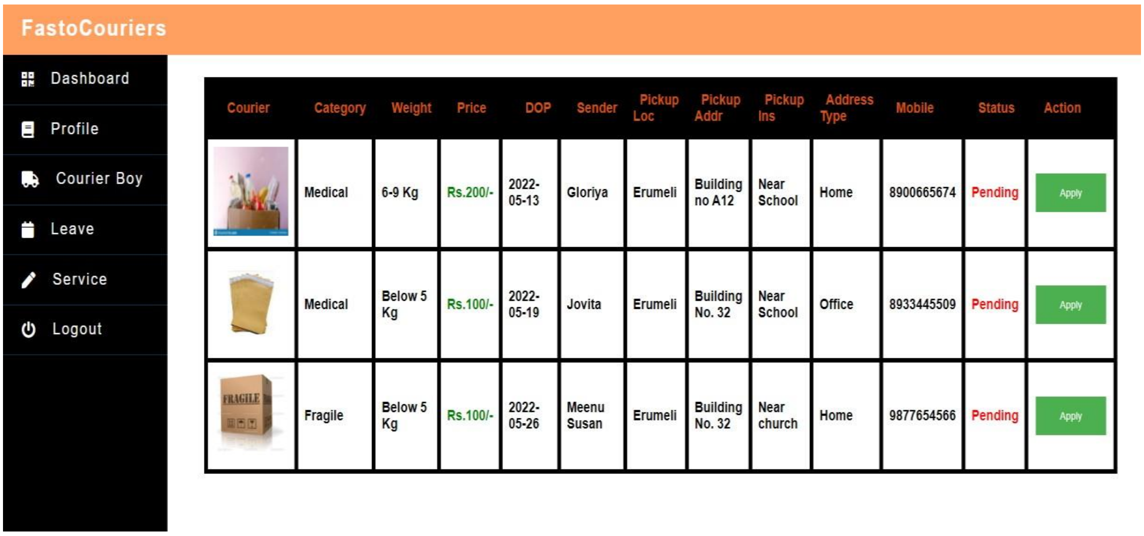
Courier	Category	DOP	Sender	Pickup Point	Weight	Price	Assign	Action
	Non-Fragile	2022-05-26	Krishna Biju	Erumeli	6-9 Kg	Rs.200/-	Selec ⌄ Select Ebin John Sanju Abraham	Apply
	Medical	2022-05-19	Soorya P	Pala	10-15 Kg	Rs.300/-	Selec ⌄	Apply
	Fragile	2022-05-20	Jose K	Erumeli	6-9 Kg	Rs.200/-	Selec ⌄	Apply
	Medical	2022-05-26	Anjali Rachel	Kanjirappally	6-9 Kg	Rs.200/-	Selec ⌄	Apply

DELIVERY BOY PAGES

Delivery boy Dashboard



Assigned couriers of Delivery boy



Courier	Category	Weight	Price	DOP	Sender	Pickup Loc	Pickup Addr	Pickup Ins	Address Type	Mobile	Status	Action
	Medical	6-9 Kg	Rs.200/-	2022-05-13	Gloriya	Erumeli	Building no A12	Near School	Home	8900665674	Pending	Apply
	Medical	Below 5 Kg	Rs.100/-	2022-05-19	Jovita	Erumeli	Building No. 32	Near School	Office	8933445509	Pending	Apply
	Fragile	Below 5 Kg	Rs.100/-	2022-05-26	Meenu Susan	Erumeli	Building No. 32	Near church	Home	9877654566	Pending	Apply