Table of Contents

[**Abstract** 2](#_Toc14783930)

[**Acknowledge** 2](#_Toc14783931)

[**Chapter 1** 2](#_Toc14783932)

[**Introduction** 2](#_Toc14783933)

[**1.1 Project Introduction** 2](#_Toc14783934)

[**1.2 Background and Justification** 3](#_Toc14783935)

[**1.3 Overview of Project** 3](#_Toc14783936)

[**1.4 Aims and Objectives** 3](#_Toc14783937)

[**Chapter 2** 4](#_Toc14783938)

[**Analysis** 4](#_Toc14783939)

[**2.1 Introduction** 4](#_Toc14783940)

[**2.2 Analysis Methodology** 5](#_Toc14783941)

[**2.3 Feasibility Study** 6](#_Toc14783942)

[**2.4 Software Requirements Specification** 7](#_Toc14783943)

[**2.4.1 Functional Requirements** 7](#_Toc14783944)

[**2.4.2 Non-Functional Requirements** 7](#_Toc14783945)

[**2.4.3 MoSCoW Prioritization** 9](#_Toc14783946)

[**2.4.4 Hardware Software Specification** 11](#_Toc14783947)

[**2.5 Use Case Diagram** 11](#_Toc14783948)

[**2.6 Initial Class Diagram (NLA)** 14](#_Toc14783949)

[**Chapter 3** 16](#_Toc14783950)

[**Design** 16](#_Toc14783951)

[**1.** **Structural Diagram** 16](#_Toc14783952)

[**1.1 Final Class Diagram** 16](#_Toc14783953)

[**1.2 Data Flow Diagram** 16](#_Toc14783954)

[**2.** **Behavioral Model** 18](#_Toc14783955)

[**2.1 Activity Diagram** 18](#_Toc14783956)

[**3.** **Database Model** 25](#_Toc14783957)

[**3.1 Data Dictionary** 25](#_Toc14783958)

[**3.2 ER Diagram** 26](#_Toc14783959)

[**4.** **Architecture Model** 28](#_Toc14783960)

[**5.** **UI Modeling** 29](#_Toc14783961)

[**5.1 Prototyping** 29](#_Toc14783962)

[**Chapter 4** 33](#_Toc14783963)

[**Coding/Implementation** 33](#_Toc14783964)

[**4.1 Introduction** 33](#_Toc14783965)

[**4.2 Programming Languages** 33](#_Toc14783966)

[**Chapter 5** 34](#_Toc14783967)

[**Other Project Issues** 34](#_Toc14783968)

[**5.1 Limitation of the Project** 34](#_Toc14783969)

[**5.2 Future Work** 34](#_Toc14783970)

[**5.3 Risk Management** 34](#_Toc14783971)

[**5.4 Configuration Management** 35](#_Toc14783972)

[**5.5 User Manual** 35](#_Toc14783973)

[**Chapter 6** 36](#_Toc14783974)

[**Testing** 36](#_Toc14783975)

[**Black Box Testing** 36](#_Toc14783976)

[**White Box Testing** 36](#_Toc14783977)

[**Chapter 7** 48](#_Toc14783978)

[**Conclusion** 48](#_Toc14783979)

[**References and Bibliography** 49](#_Toc14783980)

[**Appendix** 50](#_Toc14783981)

[**User Manual** 50](#_Toc14783982)

# **Table of Figure**

[Figure 1**SWOT Analysis** 7](#_Toc15282909)

[Figure 2 MoSCow 12](#_Toc15282910)

[Figure 3 **Use case Diagram (Actor)** 14](#_Toc15282911)

[Figure 4**Use case Diagram (Admin)** 15](#_Toc15282912)

[Figure 5**Initial Class Diagram** 16](#_Toc15282913)

[Figure 6**Final Class Diagram** 18](#_Toc15282914)

[Figure 7Data Flow diagram 19](#_Toc15282915)

[Figure 8Admin Login Activity diagram 21](#_Toc15282916)

[Figure 9Registration Activity Diagram 22](#_Toc15282917)

[Figure 10Admin action activity diagram 23](#_Toc15282918)

[Figure 11Booking activity diagram 23](#_Toc15282919)

[Figure 12User sequence diagram 26](#_Toc15282920)

[Figure 13Admin sequence diagram 27](#_Toc15282921)

[Figure 14ER Diagram 30](#_Toc15282922)

[Figure 15Three-tier Architecture 31](#_Toc15282923)

[Figure 16**Admin Login** 32](#_Toc15282924)

[Figure 17Admin dashboard 33](#_Toc15282925)

[Figure 18home page 33](#_Toc15282926)

[Figure 19Services 34](#_Toc15282927)

[Figure 20Contact us 34](#_Toc15282928)

[Figure 21 Booking 35](#_Toc15282929)

[Figure 22Configuration Management 39](#_Toc15282930)

[Figure 23User Manual of Home Page 40](#_Toc15282931)

[Figure 24User Manual of Booking 41](#_Toc15282932)

[Figure 25User Manual of Services 42](#_Toc15282933)

[Figure 26**Booked Successfully** 44](#_Toc15282934)

[Figure 27**Inserted into database successfully** 45](#_Toc15282935)

[Figure 28**Admin Login Form** 45](#_Toc15282936)

[Figure 29**Admin Login Invalid** 46](#_Toc15282937)

[Figure 30**User added successfully** 47](#_Toc15282938)

[Figure 31**Inserted user in database successfully** 48](#_Toc15282939)

[Figure 32**New Office added Successfully** 48](#_Toc15282940)

[Figure 33**Inserted new Office successfully** 49](#_Toc15282941)

[Figure 34**Shipment Added Successfully** 50](#_Toc15282942)

[Figure 35**Inserted Shipment in database successfully** 51](#_Toc15282943)

[Figure 36**Searched Courier Successfully** 52](#_Toc15282944)

[Figure 37**Edited Successfully** 53](#_Toc15282945)

[Figure 38**Courier is Tracked Successfully** 54](#_Toc15282946)

[Figure 39**Code of Booking** 58](#_Toc15282947)

[Figure 40**Code of Tracking Courier** 61](#_Toc15282948)

[Figure 41**Code of Admin Login** 62](#_Toc15282949)

[Figure 42**Code of Admin Registration** 64](#_Toc15282950)

[Figure 43**Code of add new office** 66](#_Toc15282951)

[Figure 44**Code of add manager** 69](#_Toc15282952)

[Figure 45**Code of add shipment** 71](#_Toc15282953)

[Figure 46**Code of Search courier** 72](#_Toc15282954)

# **Abstract**

This project is for Courier management that allow user to book courier and track. This website helps the user to book the items which helps the user to save time and booking. This system also can attract the user for visiting the website and giving the information about the services of our courier this system and also can communicate with user and this system can help to gain more user to use system. The user can easily access and get the information any time anywhere they want. This is a reliable cheap and faster method of getting user. And attract them by the system. This system was created for new courier office so this system aim is to provide security and helps to reduce data redundant. All the copy work is replaced by this system so that this new courier can able to run smoothly without any problem.

**Acknowledge**

I would like to express my special thanks of gratitude to my teachers (Mr. Sudeep lal Bajimaya sir), as well who helped me to complete my project. I have got an opportunity to do this wonderful project on topic (Courier Management System) which is completed with help of my teachers and doing research and I came to know about so many new things while doing this project.

Furthermore, I am also very thankful to my friends that they have helped me, suggest me about the task. They have given their full effort to guide me according to task and help me to complete task which will help me to achieve my goals. I have to appreciate the guidance given by other academic staff from the teacher’s department and as well as my family who have fully supported me in my studies.

# **Chapter 1**

### **Introduction**

### **1.1 Project Introduction**

Courier Management System which supports the high accessibility of courier services to all the corporate and to the customer. Courier which means all the goods, messages, mail, package as well as other item which can transfer from one country to another country. The system is being used for day to day activities such as booking a courier, maintain hub details, and maintain company details, process data of businesses and many other things. This Courier Management System project will have different operative system. The login system will have login facility for admin and users as well. After login user can provide all the details for placing orders.

This system will help all the employee of the company and make work systematically and flexibility. Database record will help to recognize the existing customer which will help in future reference.

We have managed a booking or a request of Quote for customer in home page. User need not have to login so there is no login system for users. Customer can view all the materials in the system. As well admin can do all the materials like change, update, delete, add, remove in the system.

### **1.2 Background and Justification**

Courier Management System is an application which handle all the process of courier from one place to another. System will be used day to day for all the courier activities. Courier services became increasingly popular with the arrival of Internet shopping. Being able to order large and multiple items from online sellers required specialist delivery services that would enable customers to not only receive their items but also enable online sellers to offer things such as next day delivery. Something that is only possible with a courier service.

This system helps to customer very easily to request for their goods transfer. Now a day’s new technology is using so people want very easy process while they are in home so, this is one kind of customer service which will satisfy customer with good relation.

System can access all the contain details of Courier and their services. It will track their goods to know the details so, users will feel very easy to use the system.

### **1.3 Overview of Project**

Courier Management System is a web-based which supports the high accessibility of courier services to the corporate and to the customer. All the features mention above are the essential features which will be implementing on the project. Before making a project, features should be identified so that it could be easy to complete project.

### **1.4 Aims and Objectives**

**Aims:**

* The main aim of this project is to use the system and uplift the company standard and business.
* Making everything easier in online system.
* To satisfy customer from our services
* To build web-based application
* To provide security while visiting site so that users may not get fear of phishing.

**Objectives:**

* To Study Feasibility
* To access Proposal.
* To gather requirement.
* To analyze gathered requirement.
* To design system UI, different Activity, Sequence, Use case, ER diagram of our system.
* Implementing or Coding for System.
* Testing the implemented system.
* Deployment of system.
* And final, maintenance will be done for design update and security of the system.

# **Chapter 2**

# **Analysis**

### **2.1 Introduction**

Analysis is the process of studying the components part of system. Its is the process to consider system carefully and using statistical methods for best system design. Analysis is the most important part in development of the system. Without analysis project fails due to lack of system analysis so, different types of research, methods function requirements should be done to develop system. It helps to find different source which breaks a complex topic into smaller part for better understanding.

### **2.2 Analysis Methodology**

The analysis methodology is well organized by analytical functions which gives knowledge about applying data analysis system functionality, making decision.

In analysis phase I am using Strengths, Weaknesses, Opportunities, Threats (SWOT) analysis.

It studies by an organization to identify strengths and weakness as well as opportunities for employees and threats in organization.

**Strengths, Weaknesses, Opportunities, Threats (SWOT) Analysis**

In my project strengths will play a vital role. It provides an advantage over others in project. Were as weakness which will help us to know about the disadvantage of project relative to others. Opportunities where elements in the environment that the business or project could exploit to its advantage. Threats where elements in the environment which could cause trouble in the project. SWOT analysis identifies the internal and external factors where strengths and weakness are internal to the project and opportunities and threats are external to the project.

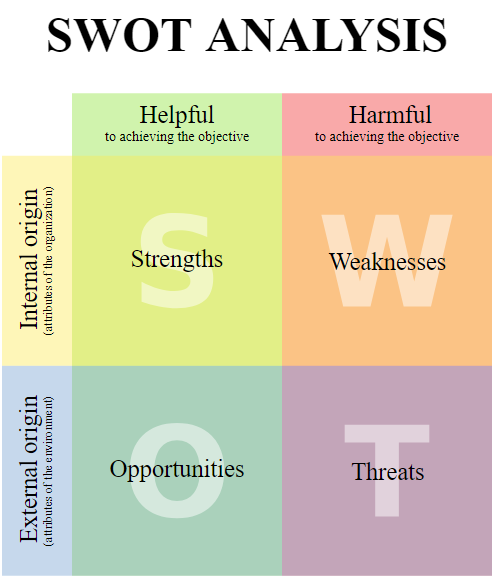


Figure **SWOT Analysis**

The reason for choosing SWOT analysis rather than other paradigms:

**Merits**

* Factor Identification

It is the primary advantage of using SWOT analysis. In project it will allows internal and external factors which are favorable and unfavorable to objectives. This allows an organization or company to understand and support in assessing core competencies.

* Simplicity

Using SWOT analysis for analyzing process individual not require proper technical skills or special training. Any individual or a team with right decision and their knowledge, skills about project will help easily to perform SWOT analysis,

* Wide application

This is one of the advantages of SWOT analysis which has also been used to analyze situation of particular business, specific project, individual or a team which provides overview of internal and external situation.

**Demerits**

* Prone to ambiguity

The SWOT framework does not provide a mechanism dealing with overlaps. Performing SWOT analysis, it generates a long list of strengths, weakness, opportunities and threats.

* Tendency to be subjective

Performing a SWOT analysis, it does not require technical skills, it is more important to put emphasis on fact system should be driven by research and data. SWOT analysis is simple to use so it is disadvantage because it can be quickly designed and performed.

### **2.3 Feasibility Study**

A feasibility study is an important factor for success in project. It is used to determine idea such as technical and legal feasibility also economic justifiable. In my project it will help to study for potential investors and lending institutions. So, there are five types of feasibility study which as explain below for project.

* Technical Feasibility

It focuses on technical resources which is available to project. It will determine technical resources which are capable to meet ideas into systems. In my project I will use hardware, software.

* Economic Feasibility

It focuses on the benefits analysis of project where it will help to determine viability, costs before financial resources which are allocated. It will help to determine positive economic benefits to the system.

* Legal Feasibility

A legal feasibility study proposed project conflicts with legal requirements like laws, data protection, social media laws. It might

* Operational Feasibility – Control efficiency, & services

Operational feasibility study how a project plan satisfies the requirements which identified in the requirements analysis phase of system development.

It helps to analyze and determine whether system will be met by completing the project.

* Schedule Feasibility – Timeline estimations & optimizing resources

Schedule feasibility is the most important for project success, if the need is not completed on time then project will fail. It this organization estimates time for project how much it will take to complete.

### **2.4 Software Requirements Specification**

A software requirements specification (SRS) is a description of a software system to be developed with functional and non-functional requirements. We should have clear understanding of software system so, that this will help to develop project and help us to achieve needs gathering from customers’ requirements.

### **2.4.1 Functional Requirements**

1. **Login Module**
2. **Admin Profile Module**
3. **Customer Module**
4. **Admin Module**
5. **Tracking**
6. **Booking**
7. **Update shipment status**
8. **Cancelation**
9. **Display Module**
10. **Comment**
11. **Logout**

### **2.4.2 Non-Functional Requirements**

1. **Security**
2. **Maintainability**
3. **Supportability**
4. **Reliability**
5. **Scalability**
6. **Usability**
7. **Performance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Requirement Specification** | **Description** | **Rational** | **Dependency** | **Remarks** |
| **FR1** | **Login** | **User can use their Id and password to get access into their account** | **To get user login** | **FR2** | **Must have** |
| **FR2** | **Add New Profile** | **Customer have to create new user if he/she doesn’t have user id & password** | **To create new user profile** | **FR1** | **Must have** |
| **FR3** | **Customer Module** | **Customer can see all the transaction details and day to day activities** | **To have details of dealership** | **FR1, FR2** | **Should have** |
| **FR4** | **Admin Module** | **Admin Module where admin have all the access of courier software system** | **To have details of all the transaction** | **FR1, FR2** | **Must have** |
| **FR5** | **Submit Button** | **The main button which will confirm booking** | **Confirmation Button** |  | **Must have** |
| **FR6** | **Delete** | **If booking is need to be cancel then delete button may be used to delete.** | **Cancellation** | **FR8, FR10** | **Should have** |
| **FR7** | **Comment** | **Where customer can provide full details of goods** | **Feedback from customer** |  | **Could have** |
| **FR8** | **Update** | **Some time details may be update so it is used to update details.** | **To edit or update if necessary** | **FR6, FR10** | **Should have** |
| **FR9** | **Remove User** | **Customer may need to remove after they don’t need account, they can use remove user feature** | **To remove user if necessary** | **FR8, FR6** | **Should have** |
| **NFR10** | **Security** | **Most needed details but be safe and shouldn’t be leak.** | **To protect from hackers** | **FR4** | **Must have** |
| **NFR11** | **Maintainability** | **Need to be maintain software timely** | **To maintain before any issues** | **FR10** | **Should have** |

Table Software requirements

The above listed are functional and non-functional requirements which are used in my project. These are the main requirements of my project to complete. If the requirements are not fulfilled that project will be incomplete.

### **2.4.3 MoSCoW Prioritization**

MoSCoW Prioritization is technique for managing requirements. It is commonly used to help key stakeholders to understand the significance of initiatives in a specific release. MoSCoW stands for ‘M’ for Must have, ‘S’ for Should have, ‘C’ for could have and ‘W’ for Won’t have.

**Must have**

In system labeled as Must have is most needed in system to be success. If even one Must have requirement is not included that the whole project will fail. It occupies 60% in whole project.

**Should have**

In system Should have are important but not that much necessary. Should have requirement can be important as Must have but not there may be another way to satisfy the requirement. It occupies 20% in whole project.

**Could have**

In system Could have are desirable but not necessary and could have satisfaction for a little development cost. This will be included if time and resources permit. It occupies 20% in whole project

**Won’t have**

Requirements which are not appropriate at the time.

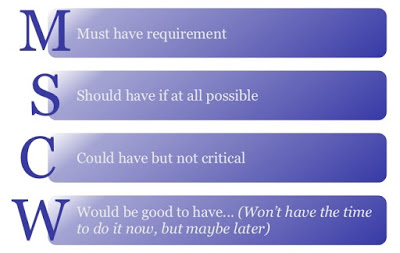


Figure MoSCow

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement Specification** | **MoSCoW Prioritization** |
| **F1** | **Login** | **Must have** |
| **F2** | **Add New Profile** | **Must have** |
| **F3** | **Customer Module** | **Should have** |
| **F4** | **Admin Module** | **Must have** |
| **F5** | **Submit Button** | **Must have** |
| **F6** | **Delete** | **Should have** |
| **F7** | **Comment** | **Could have** |
| **F8** | **Update** | **Should have** |
| **F9** | **Tracking Method** | **Could have** |
| **F10** | **Remove User** | **Should have** |
| **F11** | **Security** | **Must have** |

Table MoSCoW Prioritization

### **2.4.4 Hardware Software Specification**

The most needful requirement for system is Hardware and Software. The physical computer sources are known as hardware. It is a list of hardware compatibility list. Likewise in other hand, Software requirements defines the resource requirements that need to be installed on a computer to provide functioning of an application.

The following are the Hardware and Software specification:

**Hardware Specification**

1. **Laptop**
2. **Operating System**
3. **Architecture**
4. **Memory**
5. **Power Processing**
6. **Hard drive**

**Software Specification**

1. **Installation Tools**
2. **Star UML**
3. **PHP Strom**
4. **XAMPP**
5. **Database**
6. **Chrome**

These are the Hardware and Software which I am going to use in my project. Hardware and Software are inter connected without software hardware is useless and without software hardware cannot be used.

### **2.5 Use Case Diagram**

A use case diagram is a methodology used in system analysis to identify, clarify and organize system requirements. It is a set of actions, services, and functions which need to perform by system. It is drawn in Unified Modeling Language (UML). n this diagram model the functionality is using actors and use cases. Actors represents role that are played by users of the system. The main purpose of use cases is to gather system specific requirements and actors.

The diagram of my project is shown below:

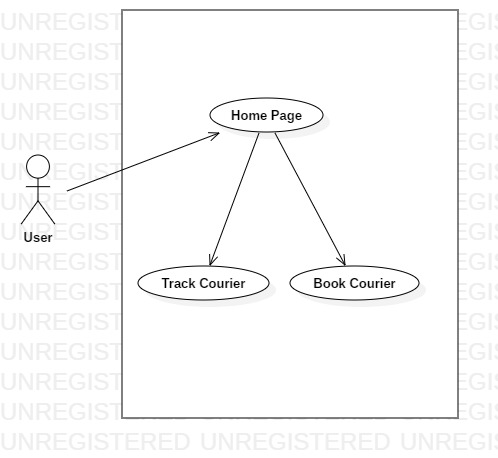


Figure **Use case Diagram (Actor)**

Above use case diagram shows actor role which can be used by users in my project. First, user can login in if he/she have login id, if not they can register their name so that it would be easy for them to use system again. Actor can add delivery, add dispatch, add and view dealership request and also add view receivers. This would make easy for user to know details about their courier package.

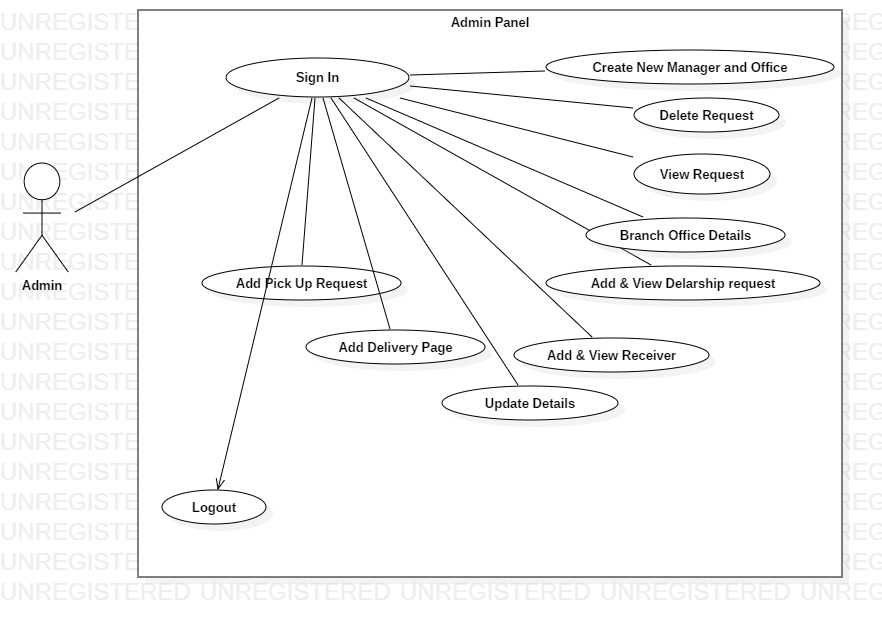


Figure **Use case Diagram (Admin)**

Likewise, admin use case shows in admin panel where after sign up admin can add, view, update and delete request, add and can view dealership request, add delivery page and also add view receiver. This would help admin to have records of all the customers details.

### **2.6 Initial Class Diagram (NLA)**

**Courier Management System**

Our courier service provided 24 hours a day, 7 days a week, 365 days a year. This is online booking courier delivery software system. Courier Management System makes your business our business. Our service will be used for managing day to day activities like booking, pickup, receiving incoming couriers and account management.

Online booking system provides personal user profile for customer. You can book their all details for goods dispatch. If goods are not need to be send then you can cancel the booked. You have access to view dealership details where you can view all the details of goods added for delivery. Online payment also can be done if customer wants to pay online. When first time an order is placed your customer log in portal can be used forever.

It will be auto generated email notification for received parcel, dispatch parcel with complete details.

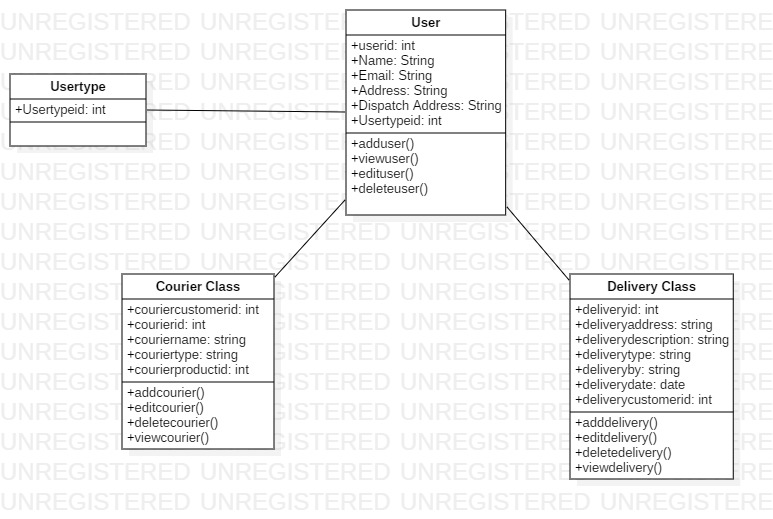


Figure **Initial Class Diagram**

**Natural Language Analysis**

|  |  |  |
| --- | --- | --- |
| **S. NO** | **Noun identification** | **Selected as candidate or not?** |
| 1 | Online | Yes |
| 2 | Business | No |
| 3 | System | No |
| 4 | Pickup | Yes |
| 5 | Notification | No |
| 6 | Customer | Yes |
| 7 | Booking | Yes |
| 8 | Dispatch | Yes |
| 9 | View | No |
| 10 | Access | No |
| 11 | Payment | No |
| 12 | Log | No |
| 13 | Portal | No |

|  |  |
| --- | --- |
| *Filtered list of the Noun* | |
| 1 | Online |
| 2 | Pickup |
| 3 | Customer |
| 4 | Booking |
| 5 | Dispatch |

|  |  |  |
| --- | --- | --- |
| **S. NO** | **Verb identification** | **Selected as candidate or not?** |
| 1 | Service | No |
| 2 | Delivery | Yes |
| 3 | Receiving | Yes |
| 4 | Cancel | Yes |
| 5 | Use | No |

|  |  |
| --- | --- |
| *Filtered list of the Noun* | |
| 1 | Delivery |
| 2 | Receiving |
| 3 | Cancel |

# **Chapter 3**

# **Design**

A design is a construction or activity or plan which is done before any project where prototype activity sequence diagram is done.

## **Structural Diagram**

### **1.1 Final Class Diagram**

Class diagram is the view of an application which visualize, describes the system and construct the executable code for developing software application.

**Justification of Final Class Diagram**

* It analyzes and design the view of application
* It describes the function which will be performed in the system
* Forward and reverse engineering

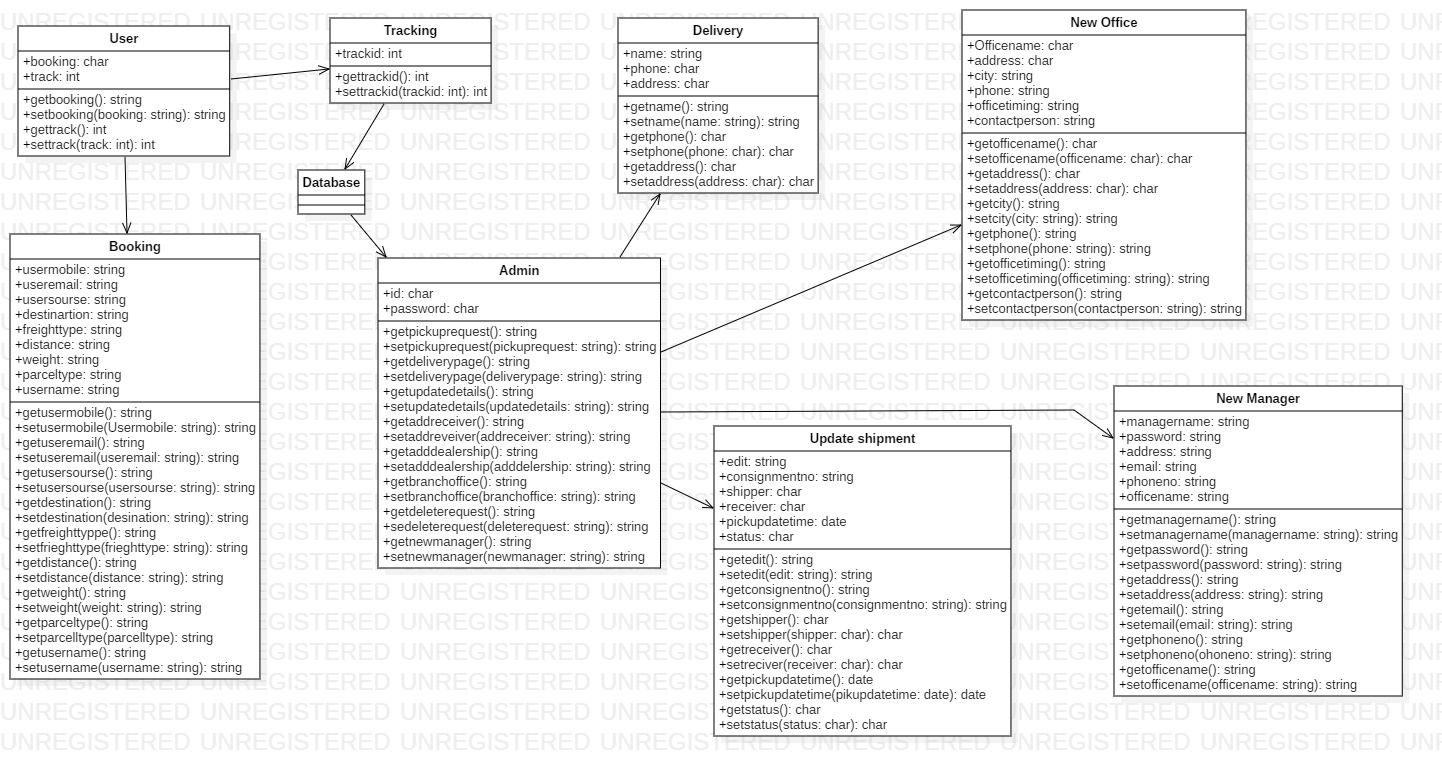


Figure **Final Class Diagram**

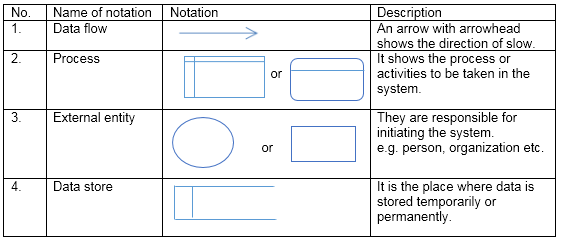
### **1.2 Data Flow Diagram**

Data Flow diagram is a process or a system which provides information about the outputs and inputs of each entity and process. It is use to illustrate the data into the system within the system and out of the system.

**Justification of Data Flow Diagram**

* Simple to create and easy to understand.
* Use to illustrate the system.
* Complex system can be illustrated to different levels of details.

**Notation**



**Table1: Notation for DFD**

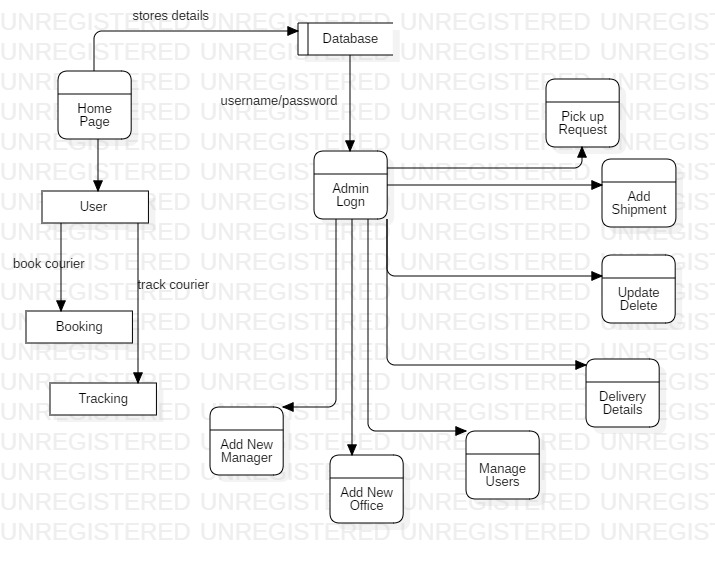


Figure Data Flow diagram

**Description**

The above diagram shows data flow of the system. After login admin can pick up courier request, view dealership, update details, edit, add and search, add new office, new manager. Likewise, user can Book and track courier.

## **Behavioral Model**

### **2.1 Activity Diagram**

Activity diagram is a graphical representation which describes parallel and conditional activities of the project. In the below diagram activity diagram is shown of my project.

I am using this design because it helps to understand how function of system will be working. I will be implementing using these designs.

**Justification of Activity diagram**

* Simple to use and understand
* It has high level understanding of the system

**Notation**

The following notation are used while making activity diagram:

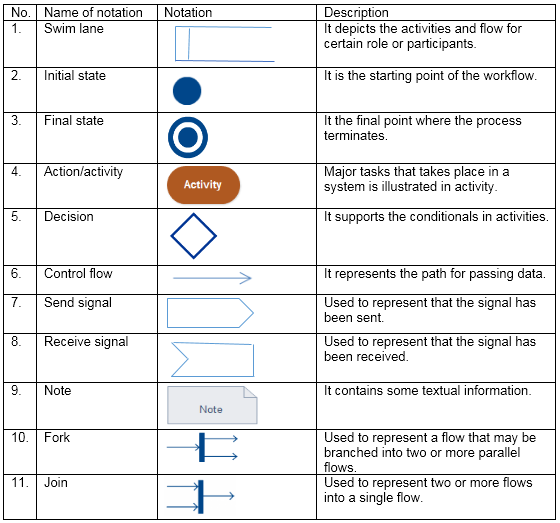


Table2: Notation for activity diagram

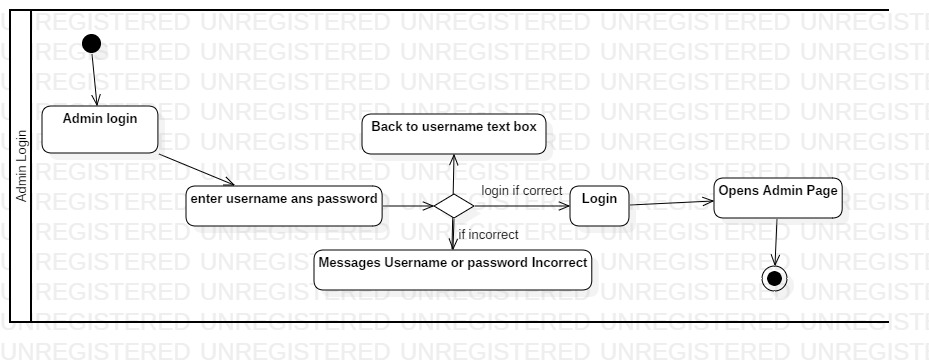


Figure Admin Login Activity diagram

**Description**

The above diagram shows the login activity of user and admin of system. The user or admin after visits the site they login with their username and password. If the provide username and password is correct then user or admin page will open if not then it returns with message ‘Username or Password is incorrect’ and goes back to text box.

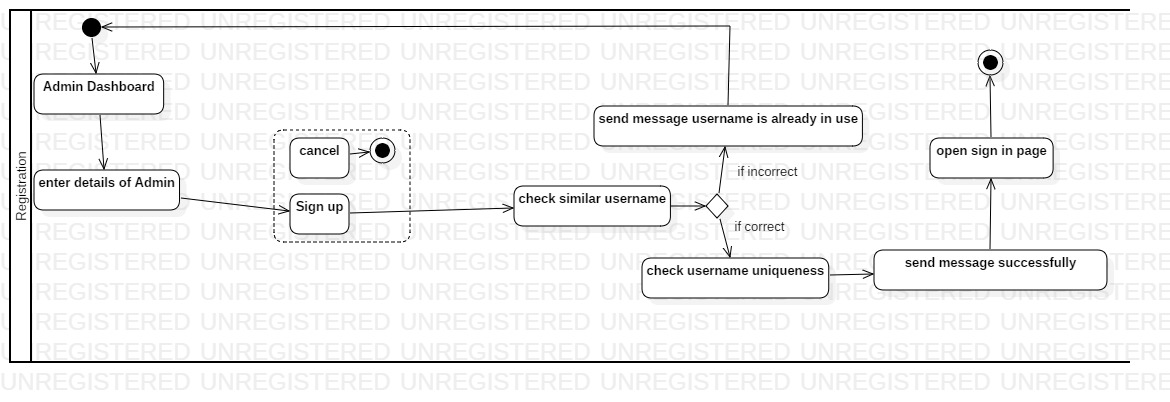


Figure Registration Activity Diagram

**Description**

The above diagram shows the registration activity of system. If admin have to add new user account who doesn’t have their account then admin can create their account by providing their details and login in the page.

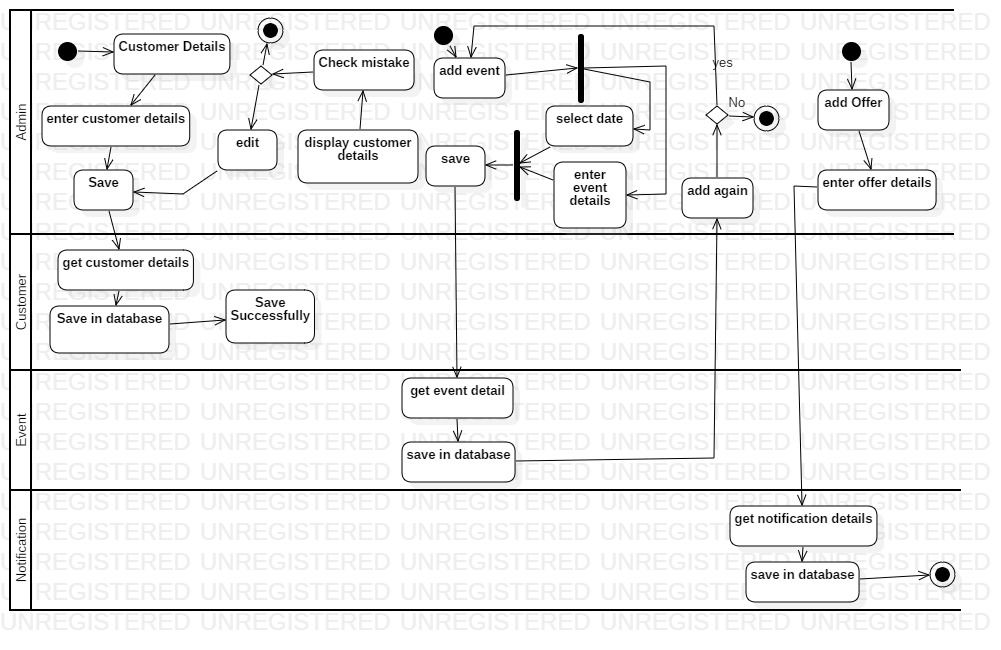


Figure Admin action activity diagram

**Description**

The above diagram shows the action which is done by admin in the system. Admin can add and edit the details of customer. They can add event and other offers. In the diagram it shows the process how the system does the work.

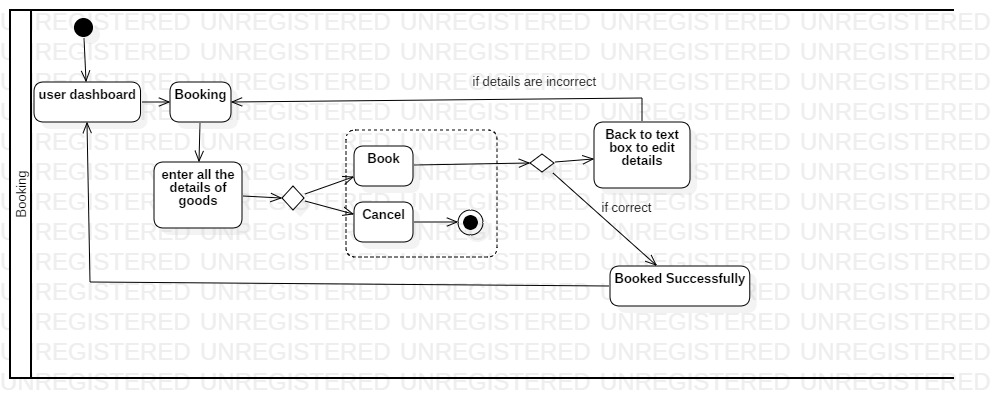


Figure Booking activity diagram

**Description**

The above diagram shows the booking activity where user add booking of their courier. It shows the activity from home page and enters all the details of goods and book if the provided details are correct than system will provide a message of booked successfully if not then it returns back to booking form.

* 1. **Sequence Diagram**

A sequence diagram is a behavioral design which shows how operations are carried out between the object which are managed in a time sequence

**Justification of Sequence diagram**

* It represents all the details of use case diagram
* It shows the inside part of a system functionality
* A team can make different functionality viewing all these sequence diagrams.

**Notation**

The all the notation that has been used while making activity diagram are listed below:

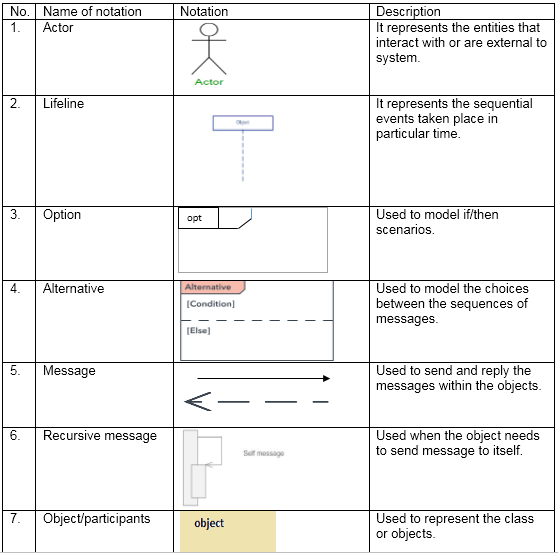


Table3: Notation of sequence diagram

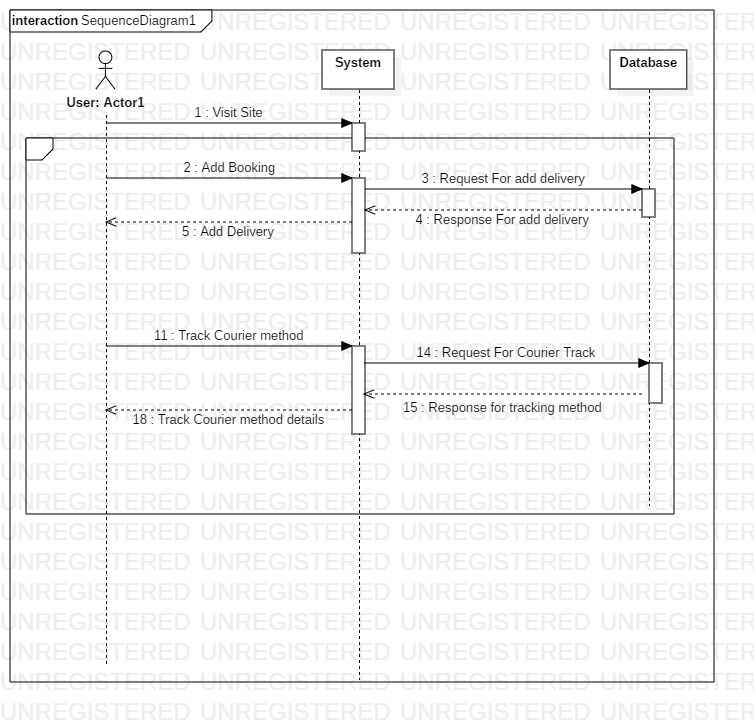


Figure User sequence diagram

**Description**

In the above diagram it shown the user activity where in the page they can book, track courier, the information after they process system will send message to database and requests every task in the user details.

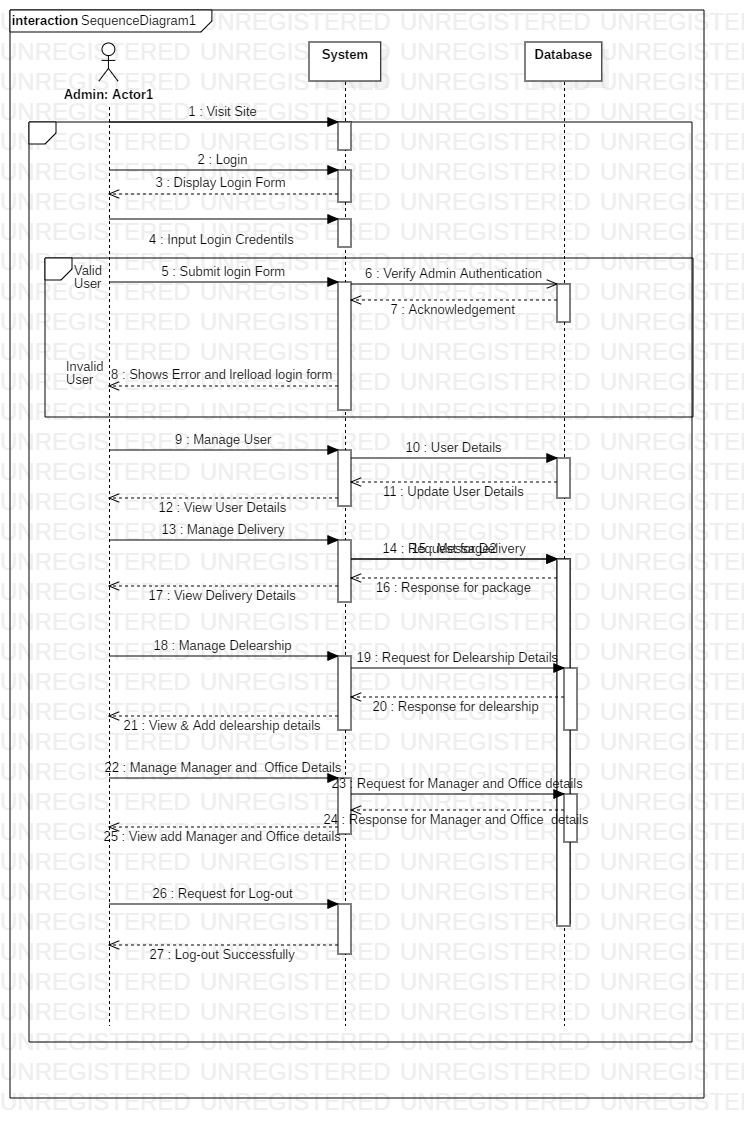


Figure Admin sequence diagram

**Description**

The above diagram shows the admin sequence diagram where it has shown the action which will be done by admin when the task will perform. In the system if admin do any process then system directly direct to database and response the details to admin page.

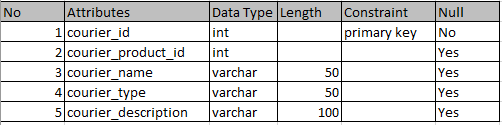
## **Database Model**

### **3.1 Data Dictionary**

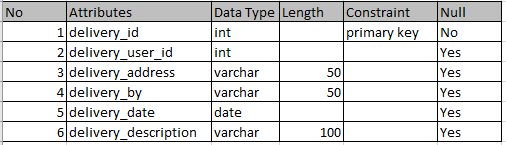
Data dictionary is the metadata about the tables. Data dictionary can be useful in any developing program.

The following are the data dictionary of the system:

* Data Dictionary for Courier



* Data Dictionary for Delivery



* Data Dictionary for Shipment

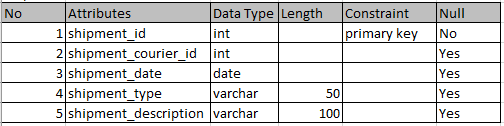


Table Data Dictionary

### **3.2 ER Diagram**

Entity Relationship diagram is a database design tool which have different entities and their relationship with each other. It is a data modelling technique for defining the business process.

**Justification of ER Diagram**

* Easy to implement the database and tables
* It can be used for organizing the relational database
* It can be useful for troubleshooting the problems

**Notation**

The following are the notation used in ER diagram:

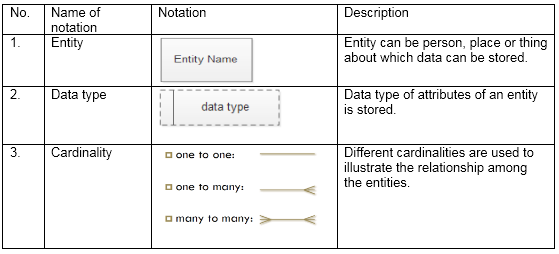


Table4: Notation of ER diagram

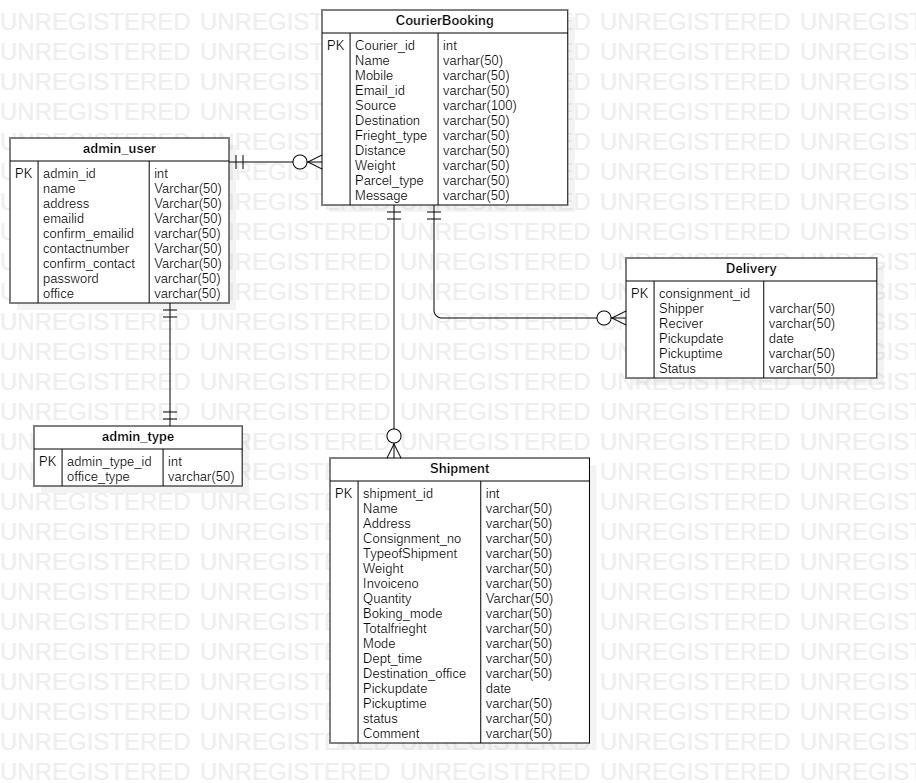


Figure ER Diagram

**Description**

The relationship between the entities with different attributes has been shown in the above diagram. User register is one to one relation with user type. Whereas user register has one to many relations with courier. Courier have one to many relations with shipment and delivery.

## **Architecture Model**

Architecture is process of planning designing and constructing any structures. The architecture for courier management system is base on the three-tier architecture. The three tiers are:

1. Presentation tier

Presentation tier is the front-end layer of project which consists user interface. HTML, css, JavaScript are applied in this layer.

1. Application tier

This tier consists of the functional business logic. It is core PHP, C#, java etc.

1. Data tier

This tier consists of data storage. For example: Oracle, SQL, MySQL

**Justification of Architecture Model**

* Flexible in managing the data.
* Change in business logic doesn’t impact the whole system.
* It is more secured architecture.

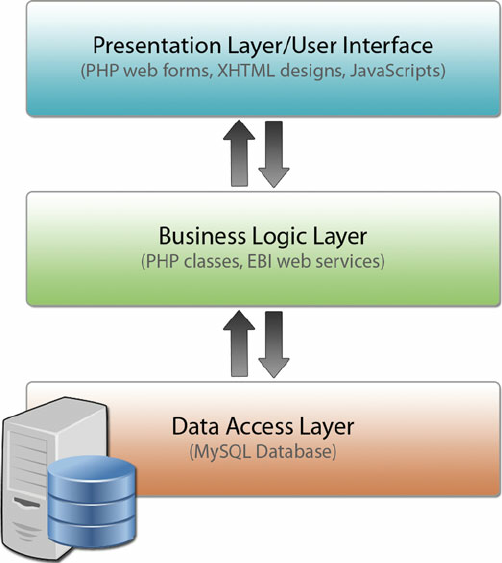


Figure Three-tier Architecture

The above is shown the three-tier architecture of my project where the data stored in data layer is presented in presentation layer on which the users interacts through the medium of application layer.

## **UI Modeling**

User Interface modelling is one of the most important part of design development which include the visual model and notation which will be using in main application. It is the first phase design before main application. It is the fun process as a software development.

### **5.1 Prototyping**

Prototyping is early model of the final application which is built to test a concept before final. It can be considered as the blueprint of the system.

I have designed the UI interface in balsamiq which are shown below:

**Justification of Prototyping**

* Detection of error before implementing
* Usability testing
* Better understanding of the software
* Improve efficiency

**Login**

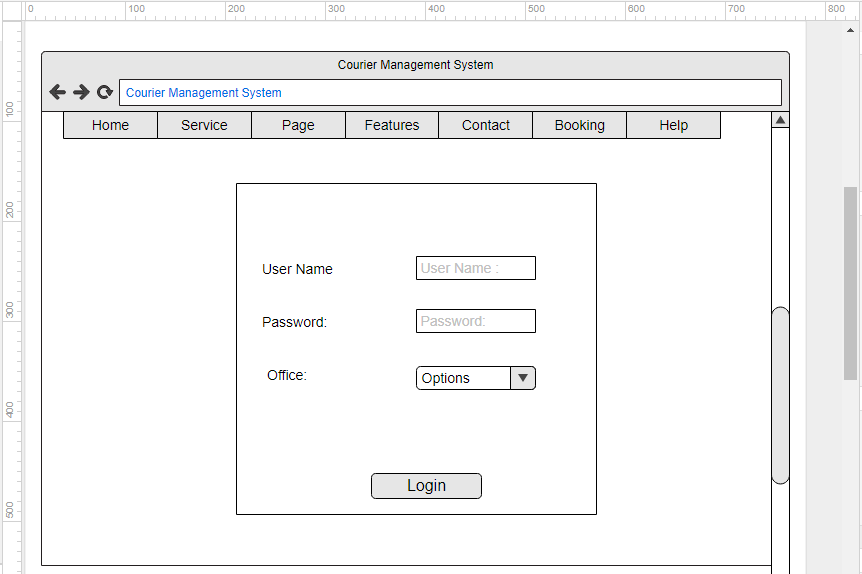


Figure **Admin Login**

**Admin dashboard**

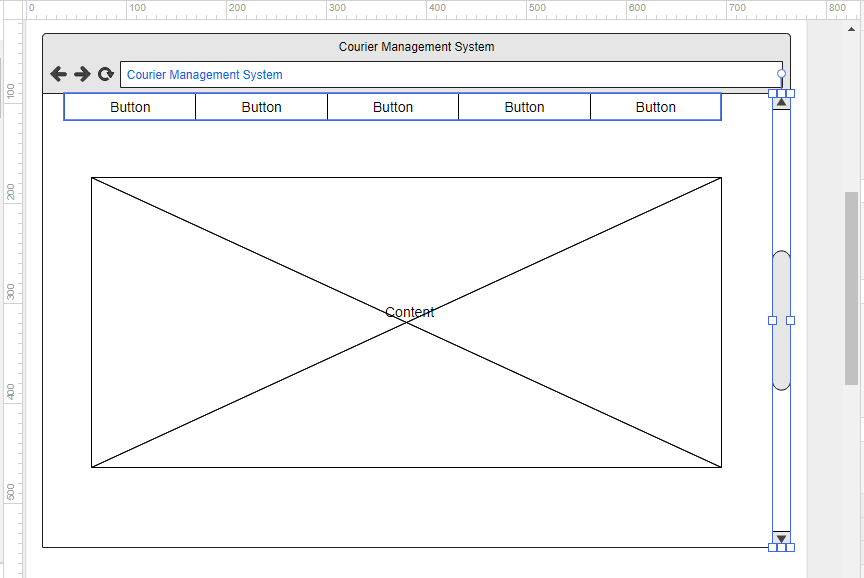


Figure Admin dashboard

**Home page**

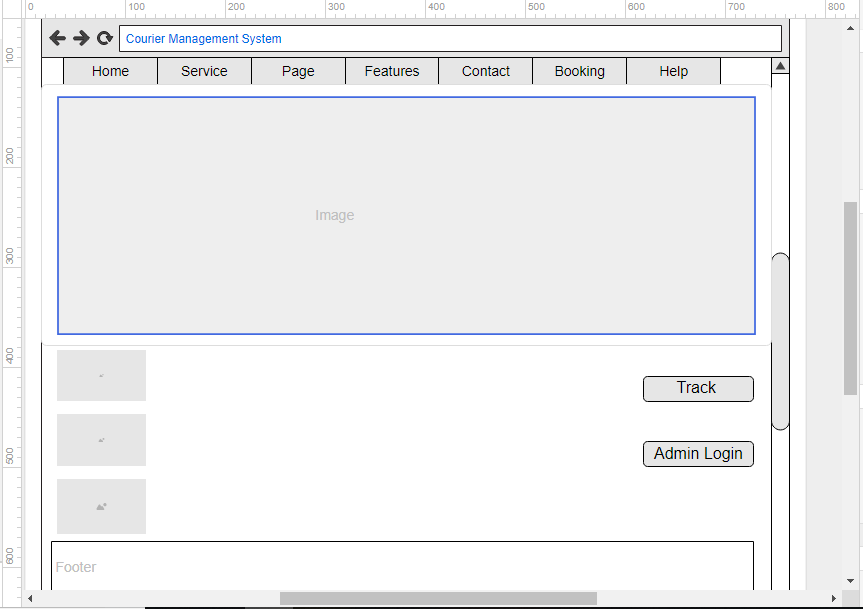


Figure home page

**Services**

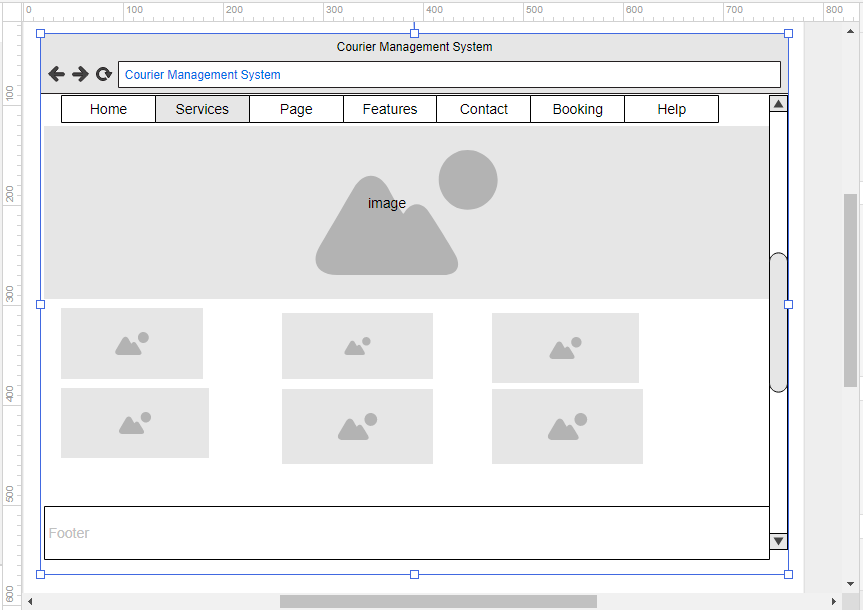


Figure Services

**Contact Us**

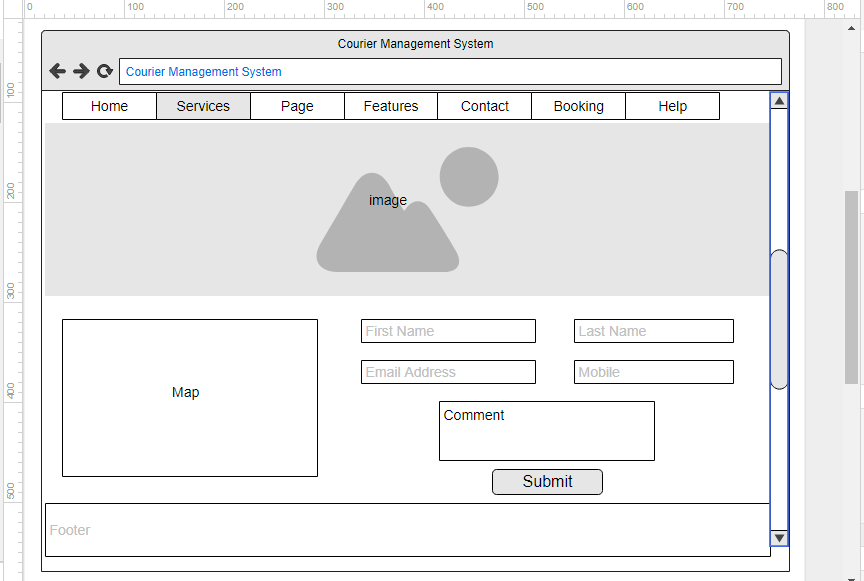


Figure Contact us

**Booking**

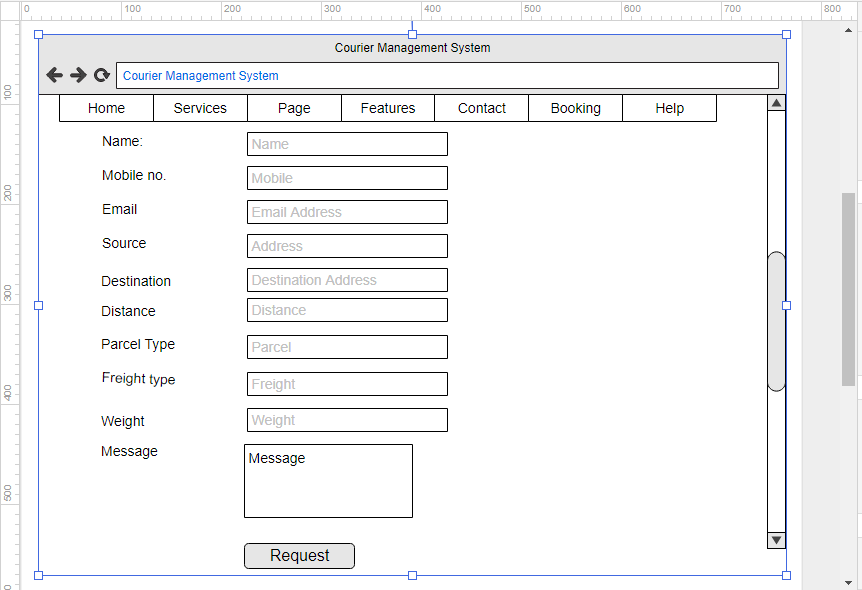


Figure Booking

# **Chapter 4**

# **Coding/Implementation**

### **4.1 Introduction**

This is a stage of a development process of SDLC. After completion of analysis and design development part has been started. In this phase coding is done with the use of PHP programming language. Implementation will be done as per analyze data and design of the system.

### **4.2 Programming Languages**

In this project development I have used PHP, and side by side JavaScript, MySQL, HTML and CSS to make the system more relevant. PHP is for server-side programming and Object-Oriented programming. Server and client side will be handled by JavaScript. For designing of the project and for interface. For storing data MySQL for the data security. As a whole I have used various types of programming language PHP, JavaScript, CSS for front-end and back-end of my project.

Code structure are listed with code screenshot in Appendix.

# **Chapter 5**

# **Other Project Issues**

### **5.1 Limitation of the Project**

* This system can only bring essential features.
* Once the delivery process is processed it cannot be undone.
* Delivery might be delay if there is any technical problem or natural disturbance.

### **5.2 Future Work**

Project has been completed and mentioned function has not been completed. In the proposal I have mentioned user manual and payment method which will be added while system upgrading. It is also next part of the system development.

There can be more features with user manual and online payment method which will be update in next updates. This feature can made users more friendly and private as well. Feedback from users can help us to know about our system upgrades. User manual will help users to know every shipment and details. But in this system only admin can have the details of courier. In this system further works are also needed so current system have some function and not included or left function can be included in future work.

### **5.3 Risk Management**

Risk Management is the process of managing risk which may occurs in the process of making project. It is the process of controlling, and identifying threats which may occurs in the project. It may come from various sources including threats, financial marks, development, accident, natural causes and disasters. So, for proper plans and strategies can be applied to minimize loss. It follows following steps to minimize risk:

* Risk Identifying
* Risk Analyzing
* Risk Evaluating
* Risk Overcome
* Risk Monitoring and reviewing

Risk, Likelihood, Consequences and Impact are listed below and shown in table.

* Virus
* Database failure
* Risk in Resources
* Risk in Financial
* Disk Problem

|  |  |
| --- | --- |
| Consequence | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

|  |  |
| --- | --- |
| Likelihood | Value |
| Low | 1 |
| Medium | 2 |
| High | 3 |
|  |  |

Table4.1 Risk Likelihood and Consequences Values

**Impact = Likelihood x Consequences**

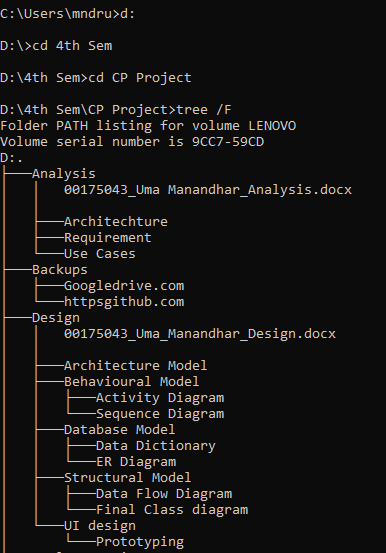
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Consequences** | **Impact** | **Action** |
| **Virus** | **2** | **1** | **2** | **Installed and regular scanned Antivirus.** |
| **Database Failure** | **1** | **4** | **4** | **Regular backup of data in database.** |
| **Risk in Resources** | **2** | **3** | **6** | **Properly using available resources** |
| **Risk in Financial** | **2** | **5** | **10** | **Preplanning and conducting budget wise system** |
| **Disk Problem** | **1** | **5** | **5** | **Daily Backup of the project.** |

Table 4.2 Risk Management Table

### **5.4 Configuration Management**

Configuration Management is a system process of identifying establishing and maintaining consistency of products performance, its functionality and physical attributes with its requirement operational and design information. This process is mainly used by military organization to manage changes of complex system. It is a tool for providing control of deliverables and avoiding mistakes.

The structure for my project is shown below:



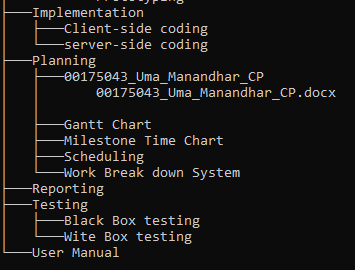


Figure Configuration Management

### **5.5 User Manual**

A user manual is a system which provides guide or a description on how to use the system. A new user might not know how to use the system so, I have created a user manual which will be easy for new user to use system. For my courier management system, the following are the manual contains all the description and guide for the use of system.

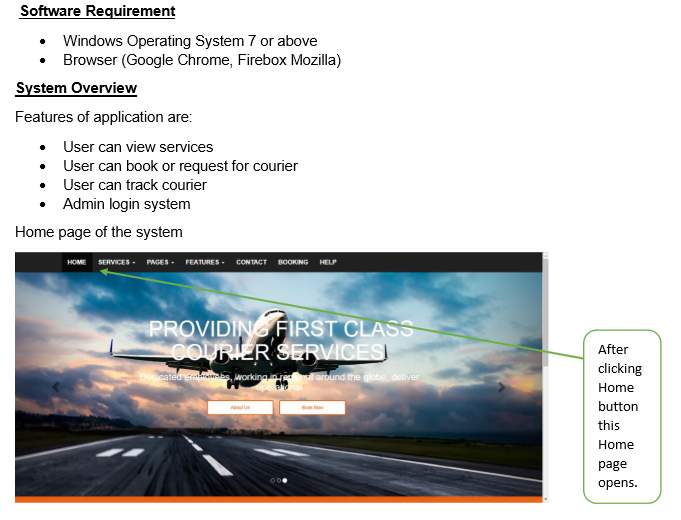


Figure User Manual of Home Page

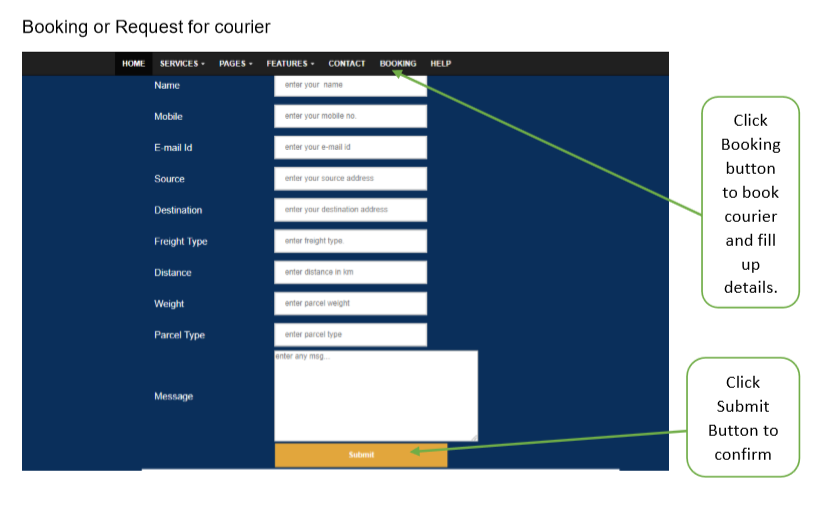


Figure User Manual of Booking

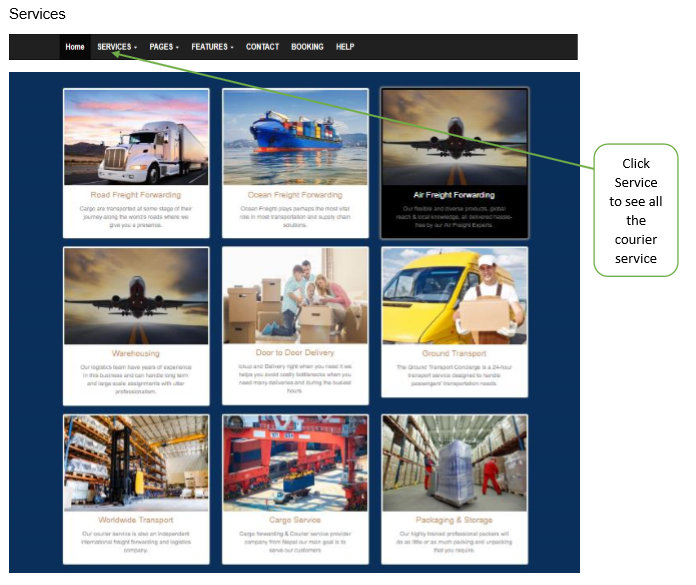


Figure User Manual of Services

# **Chapter 6**

# **Testing**

Testing is one of the stages of software development lifecycle. This process is frequently performed process in system. It is a never-ending process in very system or business. Today’s technology seems needs of customer is changing day by day so, system must also change with accordingly. So, testing must be done on very update and upgrades before submitting to customer so that it would find bugs, vulnerabilities in the system.

There are many types of testing and two of them are done in two different ways which are shown below:

### **Black Box Testing**

Black Box testing is one of the main testing which is done by providing Input to the system and expected output. It is done outside of the system. Graphical and interface will be done in this testing which is also successfully tested in my project.

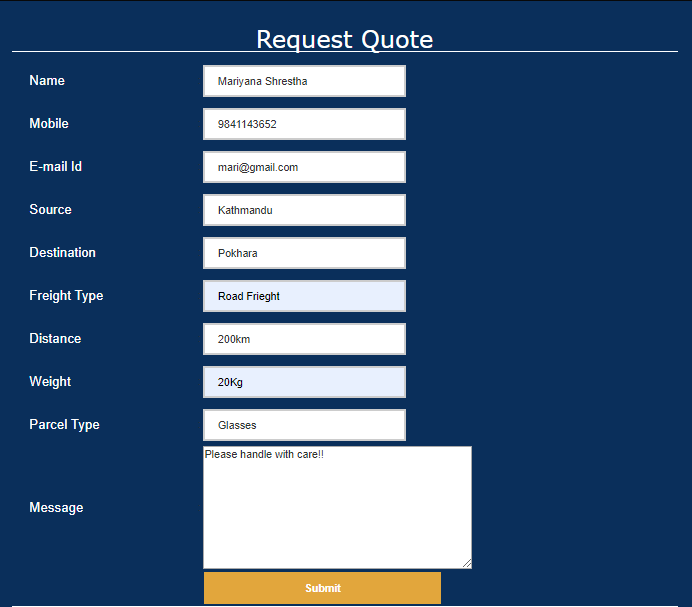
### **White Box Testing**

White box testing is another tester which is done in internal work of the system. This is done to test how system is working and finding errors, bugs in the coding part.

**Black Box testing for my Project:**

User Booking for Courier

|  |  |
| --- | --- |
| Test Id | Test1 |
| Test Purpose | To validate Booking form |
| Tested data | Shown in figure |
| Expected Result | Validation of Booking form |
| Actual Result | Booked form validate successfully |
| Result | Pass |



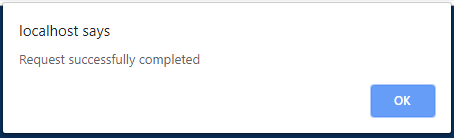


Figure **Booked Successfully**

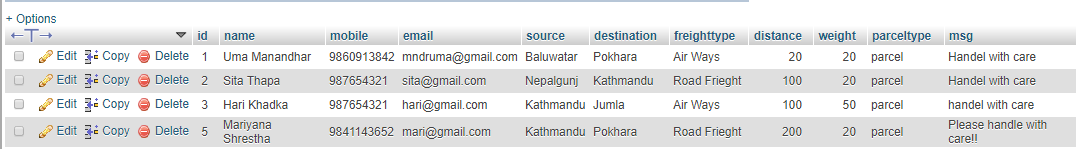


Figure **Inserted into database successfully**

**Admin Login:**

|  |  |
| --- | --- |
| Test Id | Test2 |
| Test Purpose | To validate admin login form |
| Tested data | Shown in figure |
| FExpected Result | Validation of login Form |
| Actual Result | Form validate successfully |
| Result | Pass |

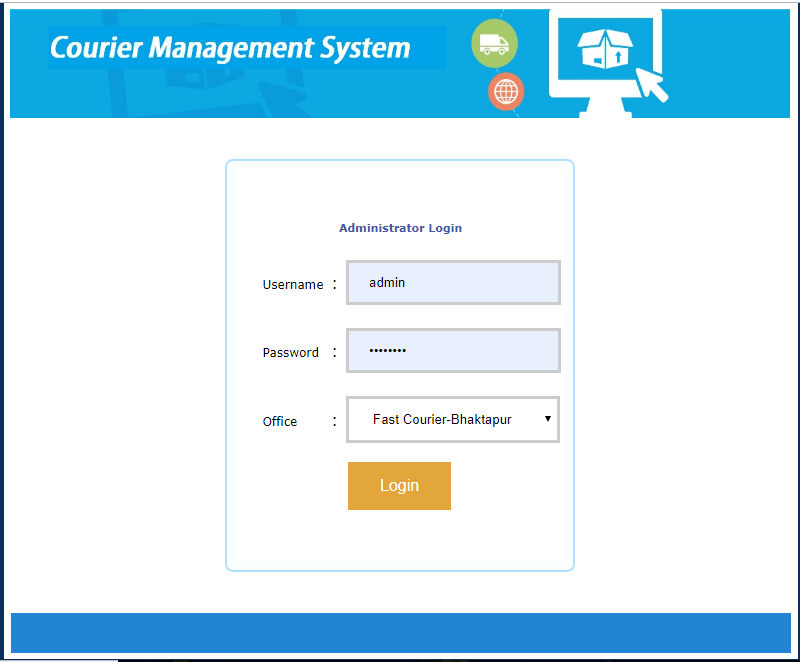


Figure **Admin Login Form**

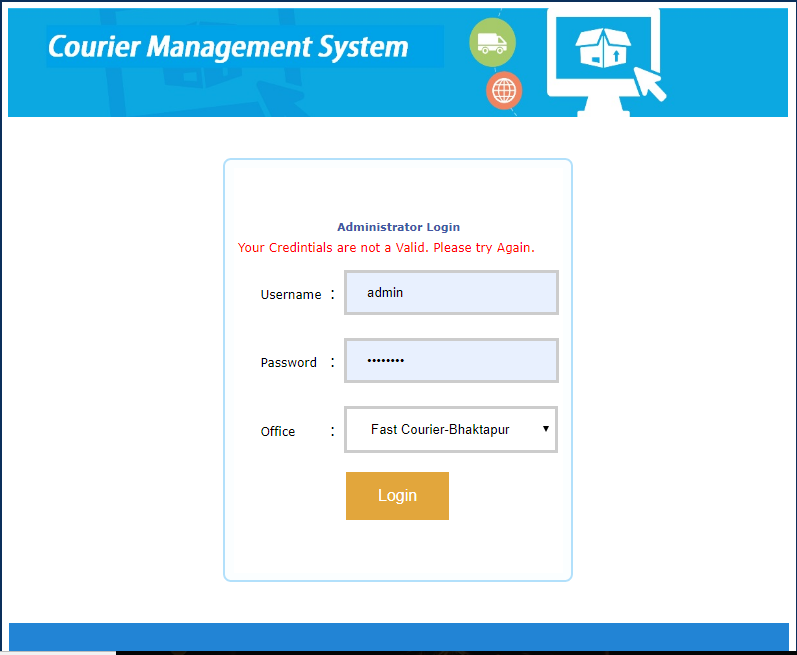
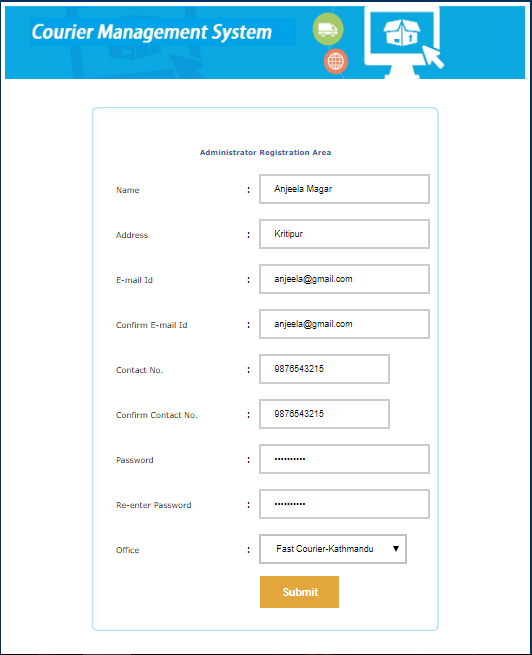


Figure **Admin Login Invalid**

Add user:

|  |  |
| --- | --- |
| Test Id | Test2 |
| Test Purpose | To validate user, add for office |
| Tested data | Shown in below |
| Expected Result | Validation of user add form |
| Actual Result | User added form validate successfully |
| Result | Pass |



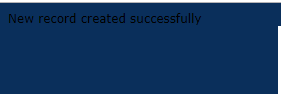


Figure **User added successfully**

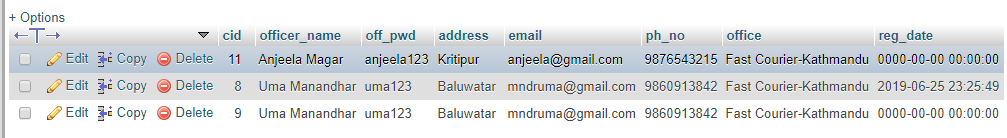


Figure **Inserted user in database successfully**

Add New Office

|  |  |
| --- | --- |
| Test Id | Test3 |
| Test Purpose | Adding new office / Branch |
| Tested data | Shown in below |
| Expected Result | Validation for adding new office |
| Actual Result | New office added successfully |
| Result | Pass |

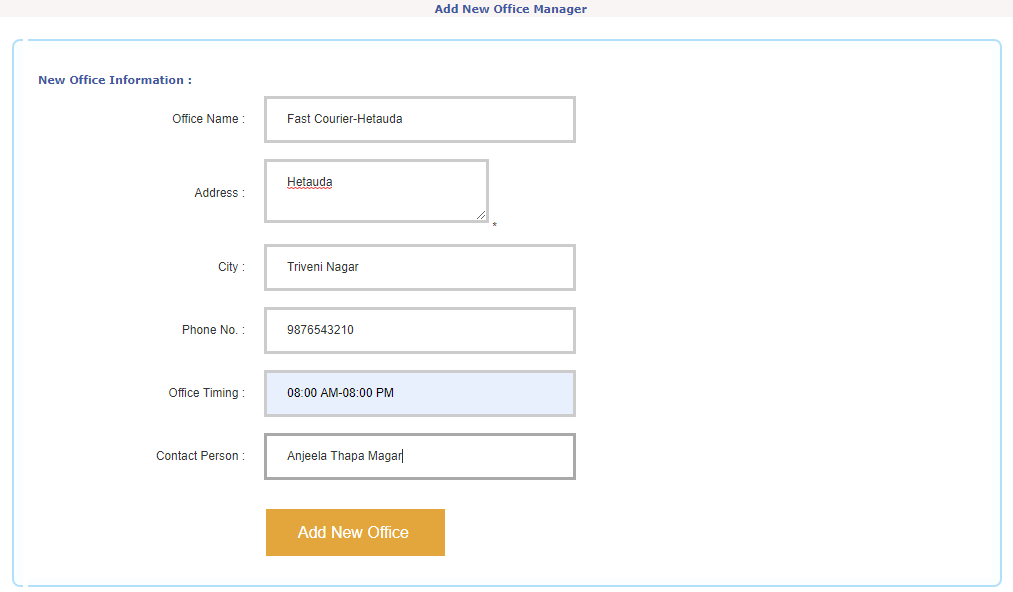




Figure **New Office added Successfully**

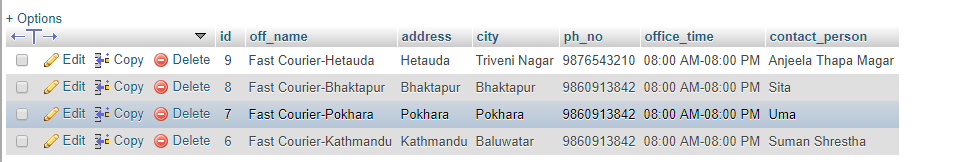
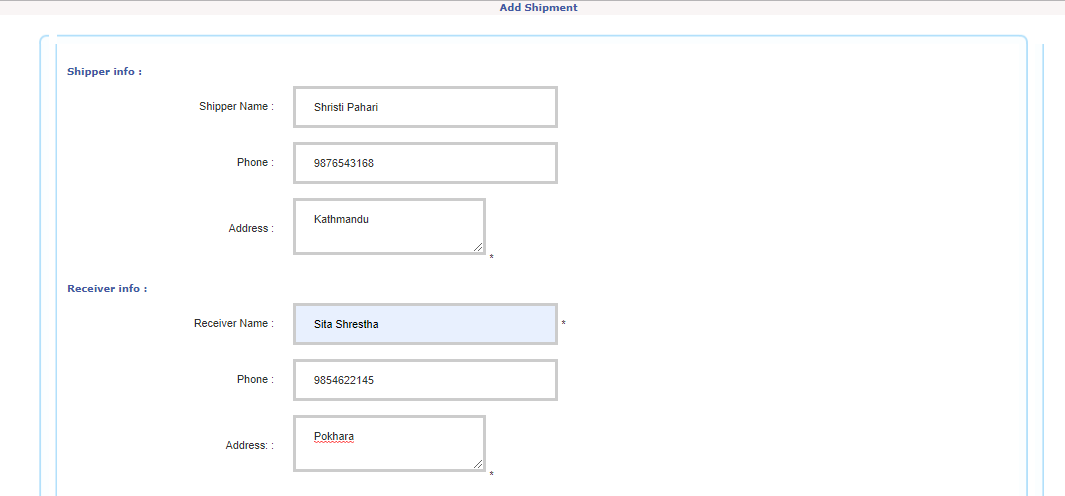
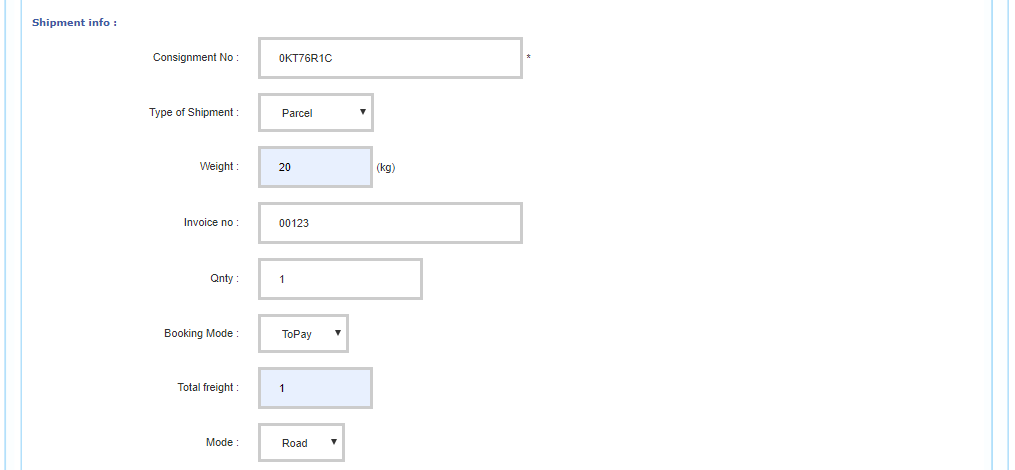


Figure **Inserted new Office successfully**

Add Shipment

|  |  |
| --- | --- |
| Test Id | Test4 |
| Test Purpose | Adding Shipment |
| Tested data | Shown in below |
| Expected Result | Validation for adding Shipment |
| Actual Result | Shipment added successfully |
| Result | Pass |







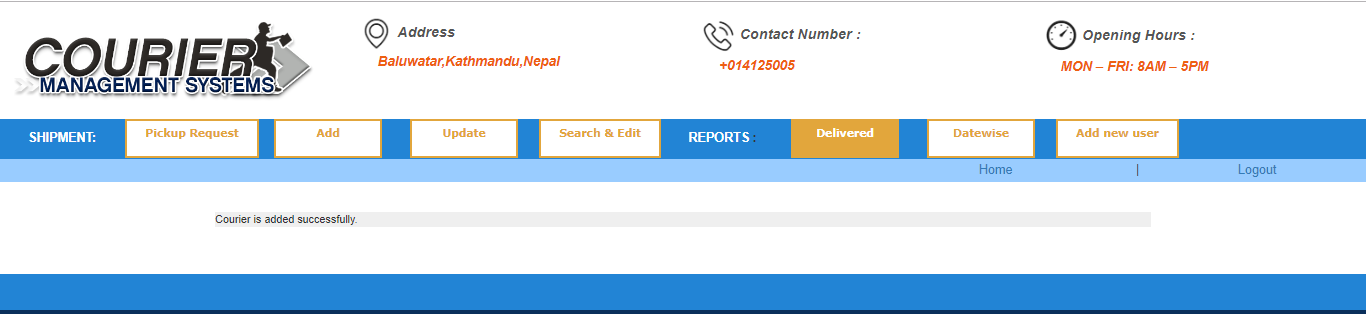


Figure **Shipment Added Successfully**

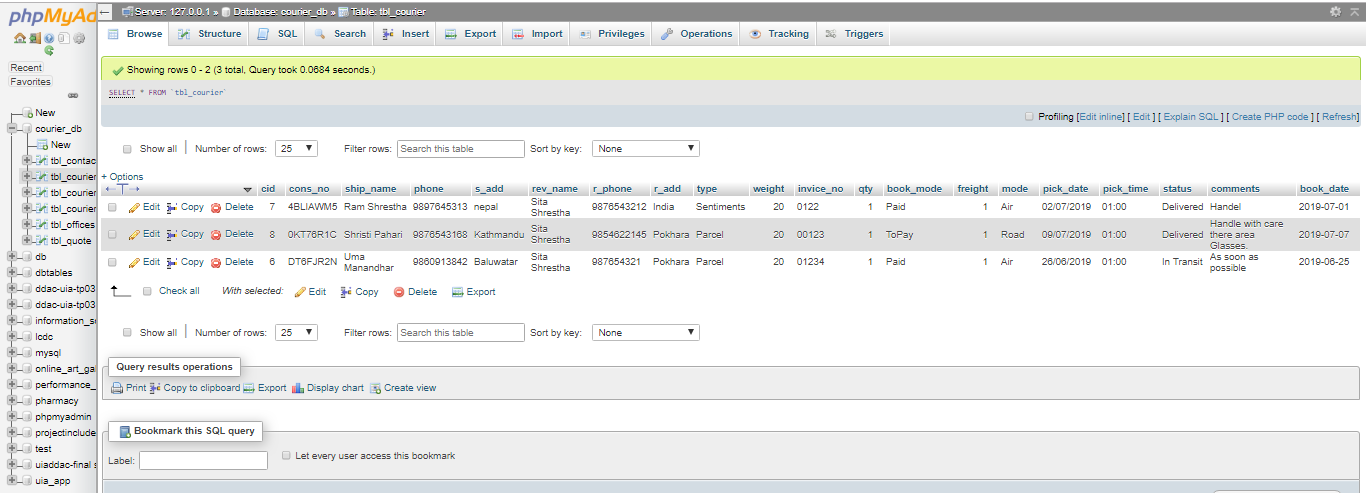
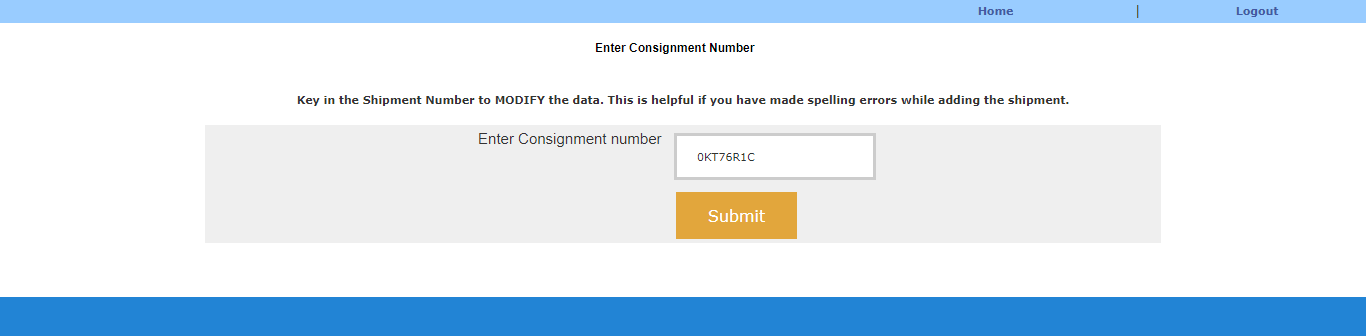


Figure **Inserted Shipment in database successfully**

Search Courier

|  |  |
| --- | --- |
| Test Id | Test5 |
| Test Purpose | Search Courier |
| Tested data | Shown in below |
| Expected Result | Validation for Searching courier with consignment number |
| Actual Result | Courier searched successfully |
| Result | Pass |



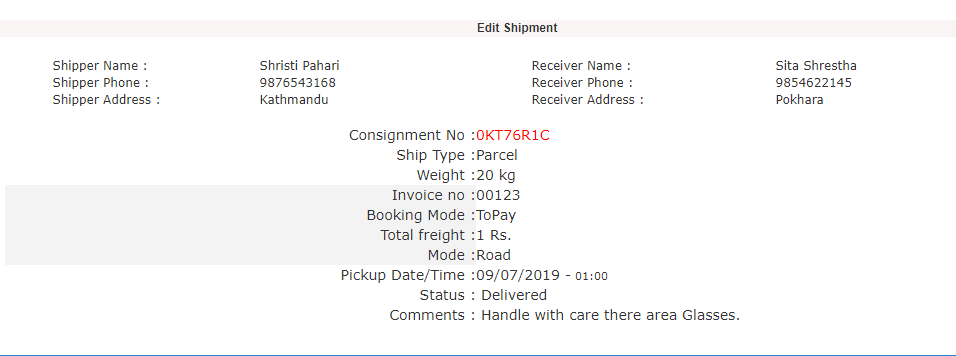
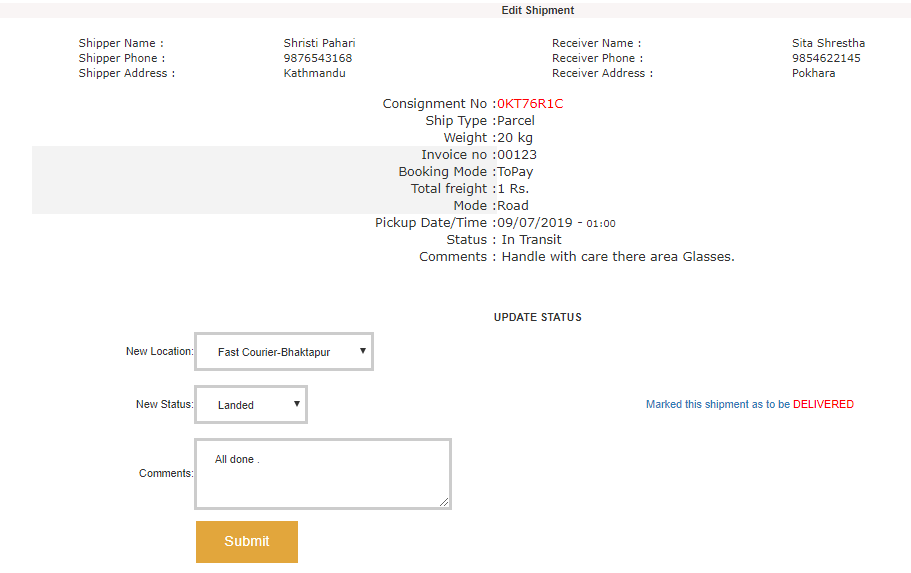


Figure **Searched Courier Successfully**

Edit Shipment

|  |  |
| --- | --- |
| Test Id | Test6 |
| Test Purpose | Edit Shipment |
| Tested data | Shown in below |
| Expected Result | Validation for Editing Shipment |
| Actual Result | Shipment edited successfully |
| Result | Pass |



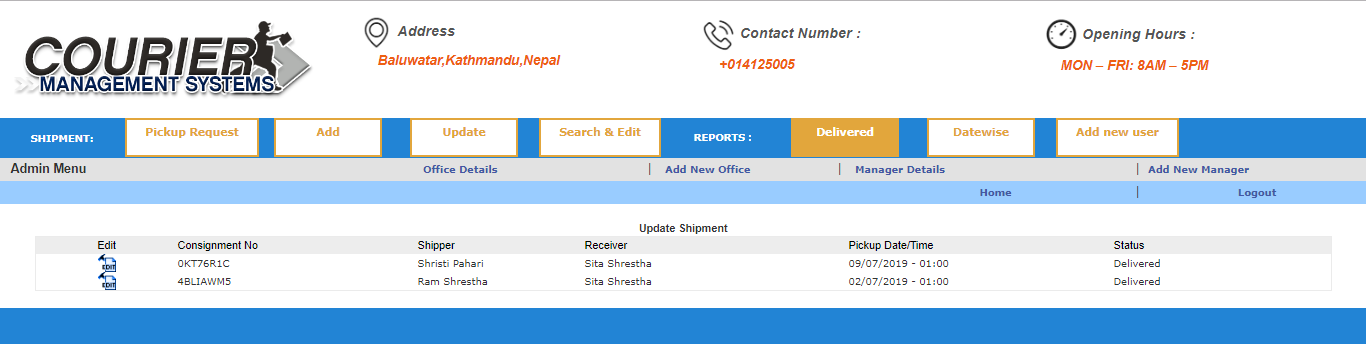
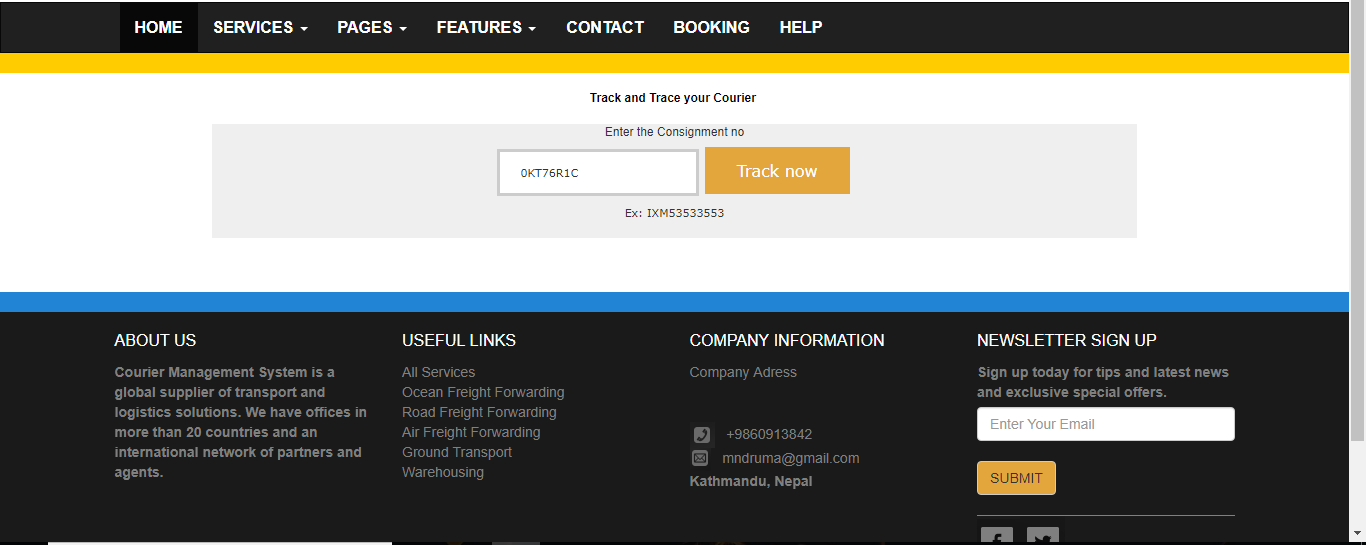


Figure **Edited Successfully**

Track Courier

|  |  |
| --- | --- |
| Test Id | Test7 |
| Test Purpose | Track Courier |
| Tested data | Shown in below |
| Expected Result | Validation for tracking courier |
| Actual Result | Courier tracked successfully |
| Result | Pass |

S

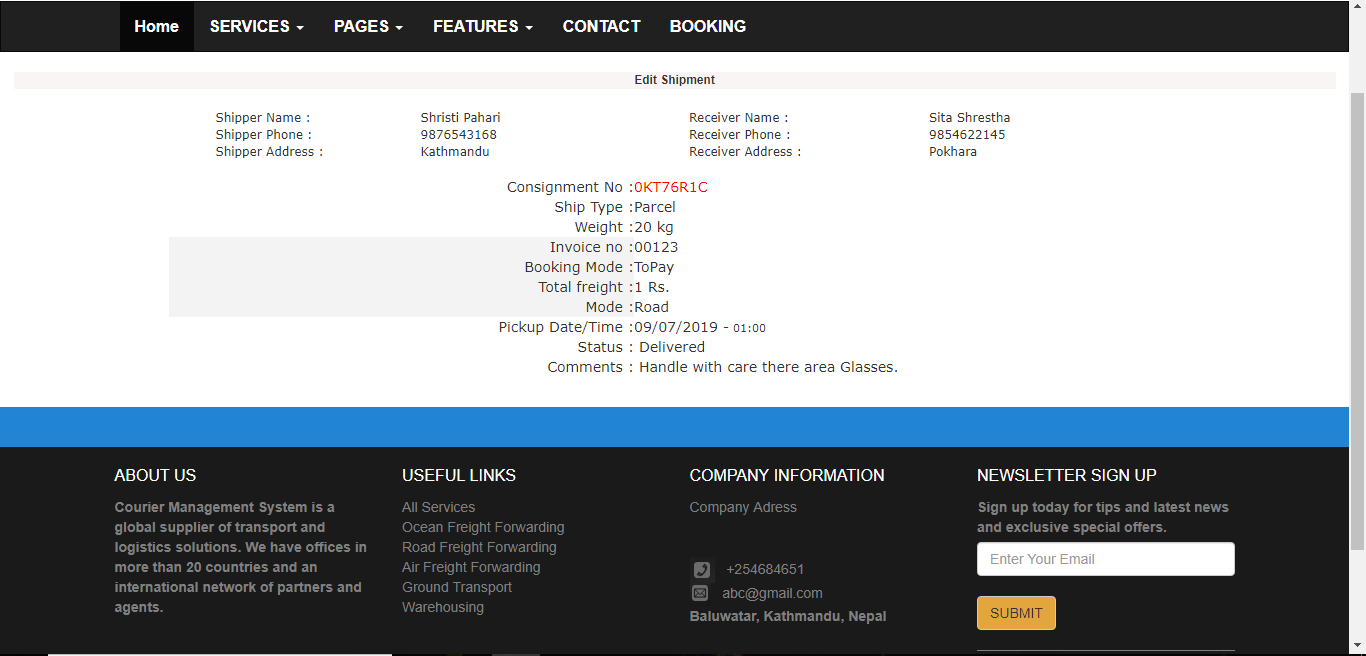


Figure **Courier is Tracked Successfully**

# **Chapter 7**

# **Conclusion**

Finally, the project has been competed as per scheduled. For the first I proposed a system that will be made. In this process, I introduced aim & objectives of project and scheduled task to ever phase of software development. After proposal, analysis was started where, use case diagram, requirements specification, architecture of system, prioritization of functionalities and initial class diagram was done. After completion of analysis, I started design of the system where, final class diagram of system, activity diagram, sequence diagram, database design like ER diagram and Data Dictionary. As the design has been completed the main implementation or coding of the system was started. PHP core has been used for coding so all the details has been updated in above document related implementation. After implementing the system, I started testing where black box has been tested to my system. I tested essential all the functionalities of the system. At the end the system has been documented and my project “Courier Management System” has been completed.

# **References and Bibliography**

**Sharma, L.** (2017, January 29). *Configuration Management*.

URL : <http://toolsqa.com/software-testing/configuration-management/>

Database Model

URL: <https://www.lucidchart.com/pages/database-diagram/database-models>

# 

# **Appendix**

## **Code Architecture**

**Booking**

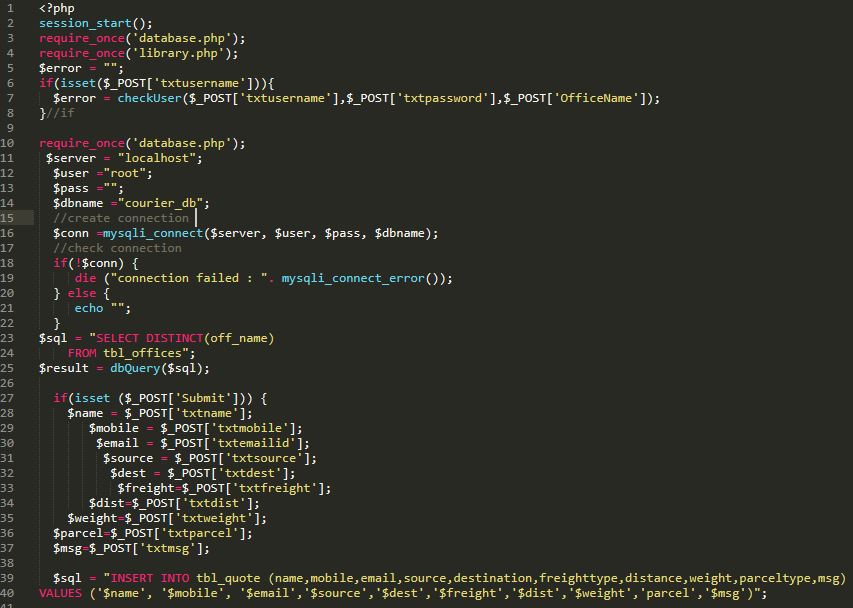
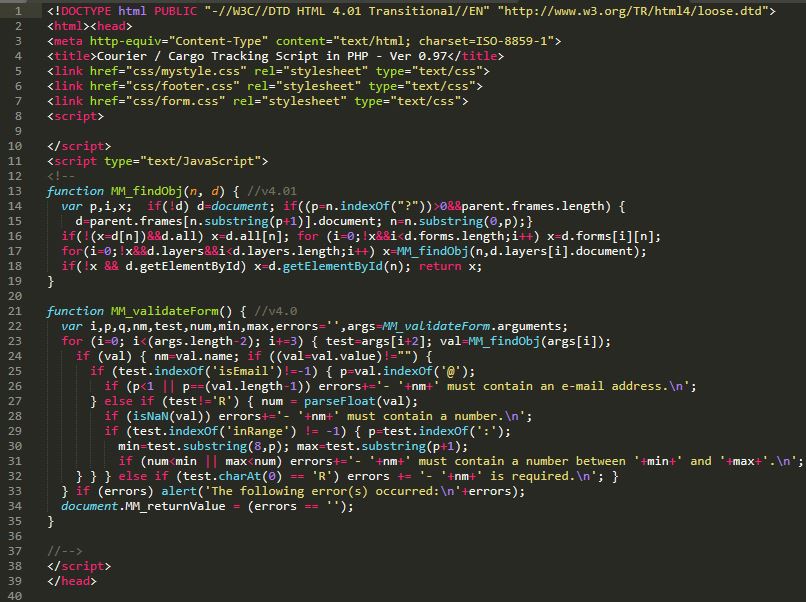


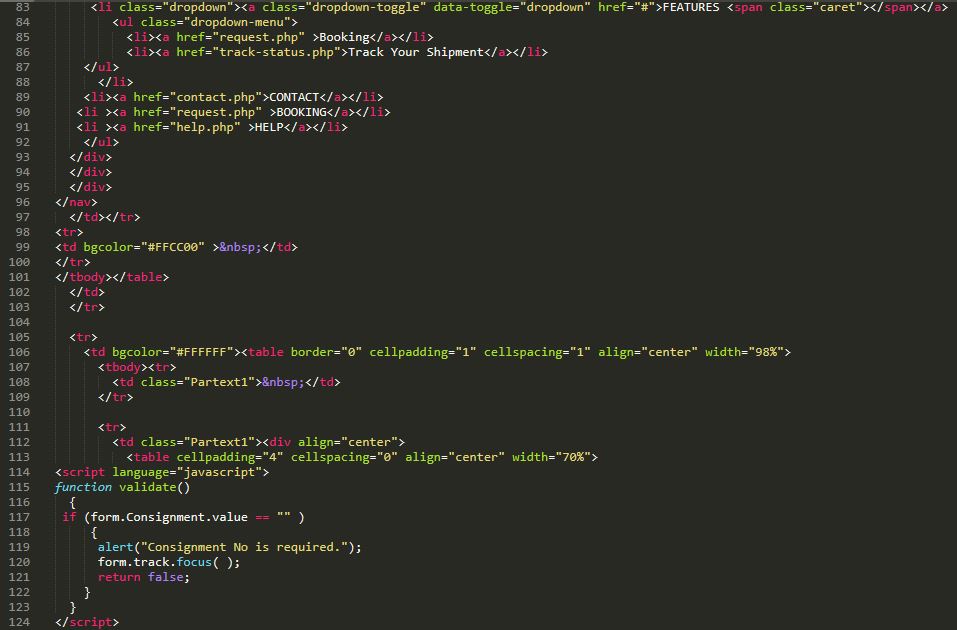


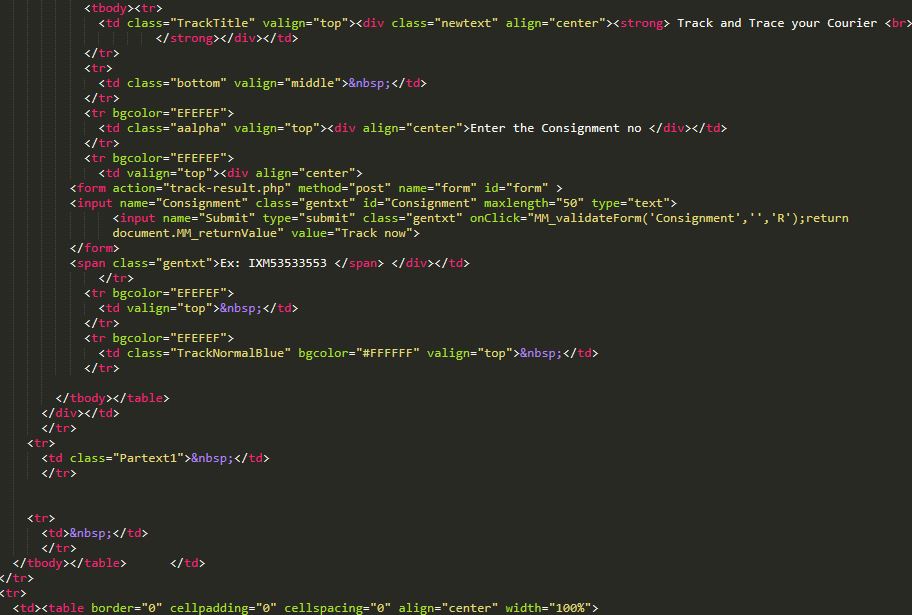
Figure **Code of Booking**

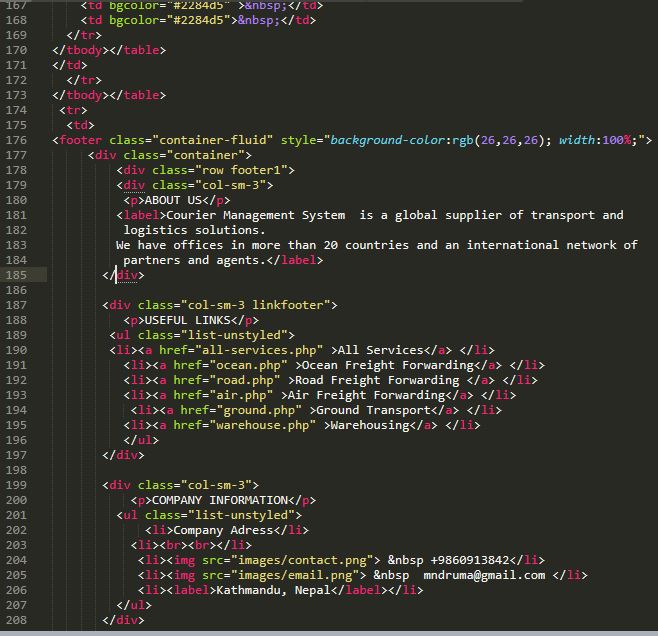
**Track**











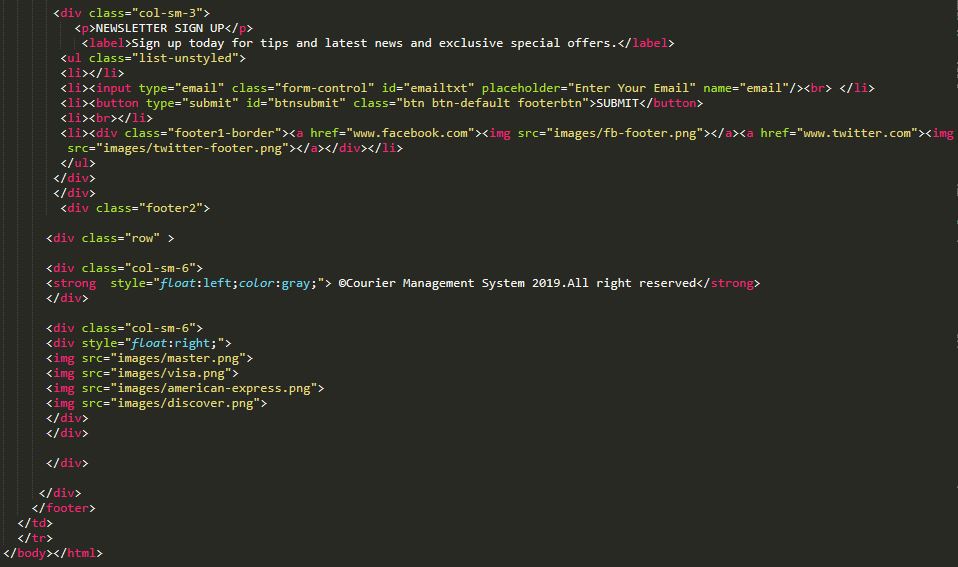


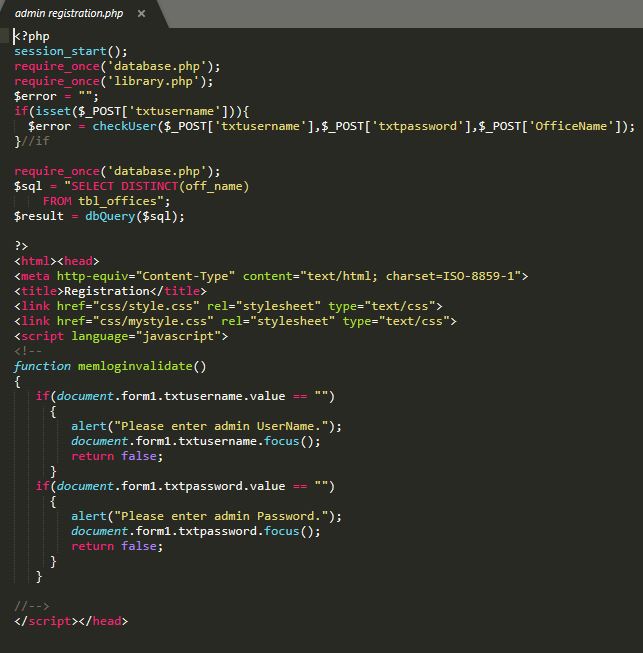
Figure **Code of Tracking Courier**

**Admin Login**

;

Figure **Code of Admin Login**

**Admin Registration**







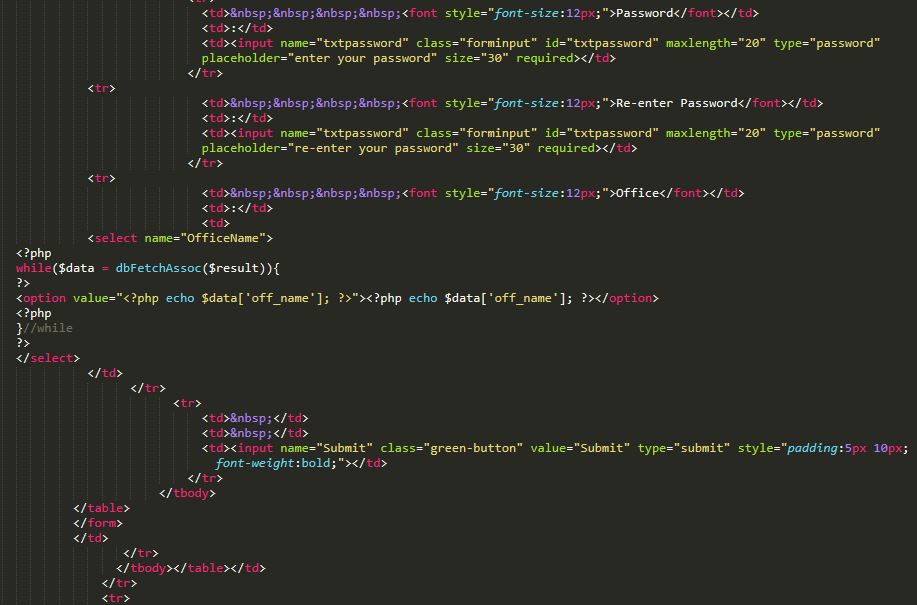
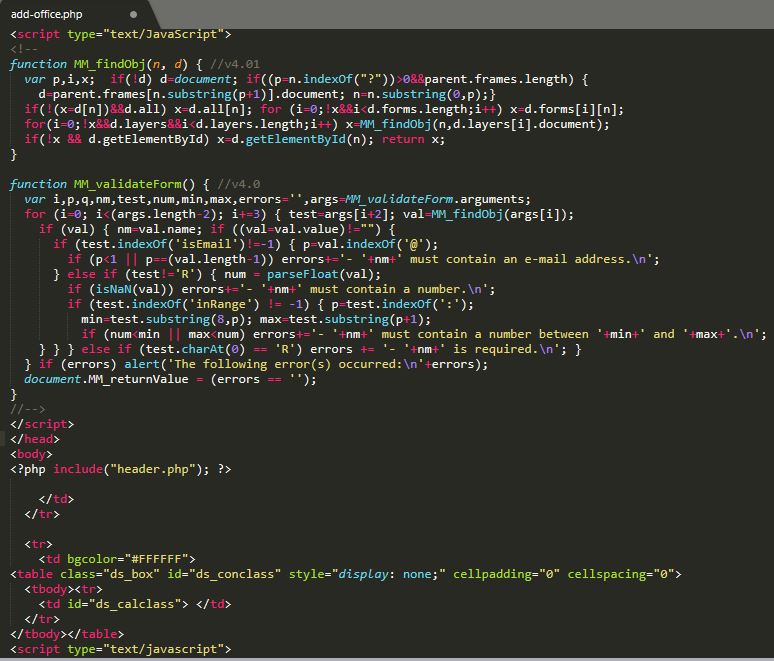
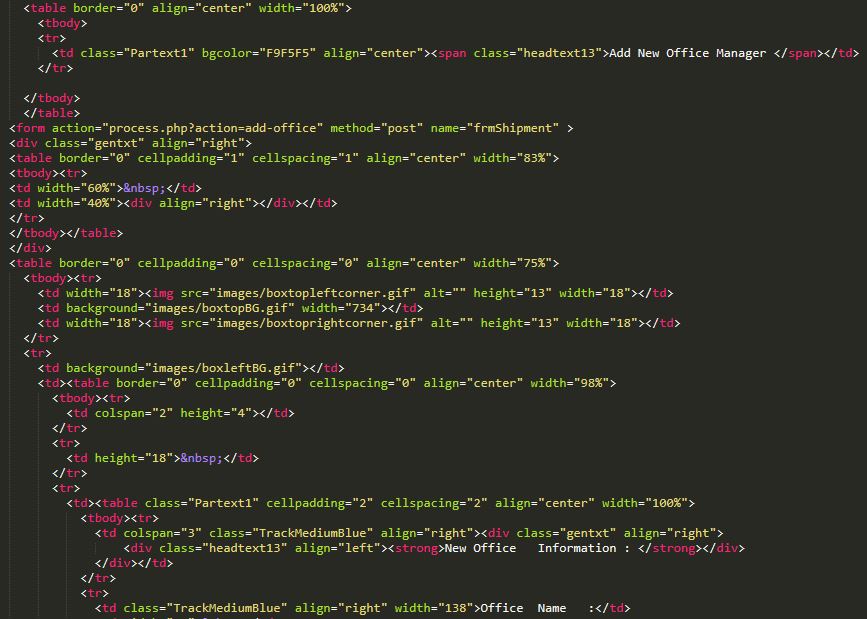


Figure **Code of Admin Registration**

**Add new Office**





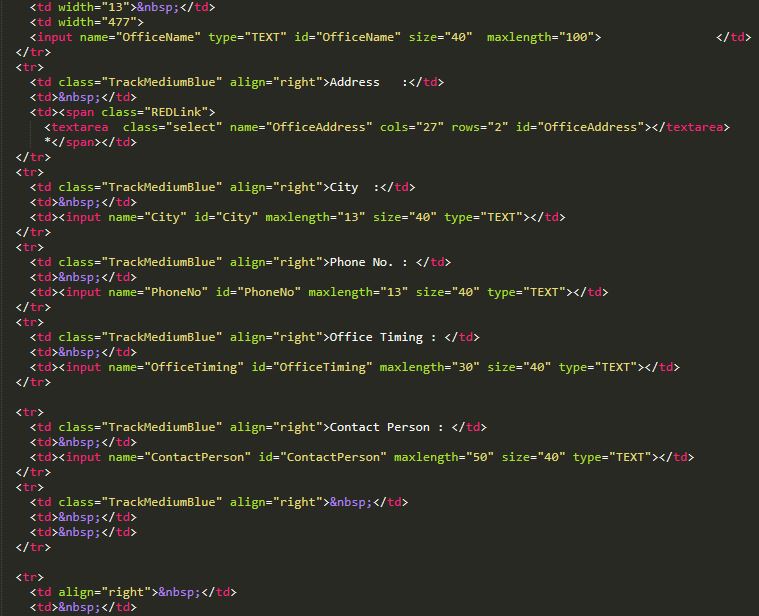
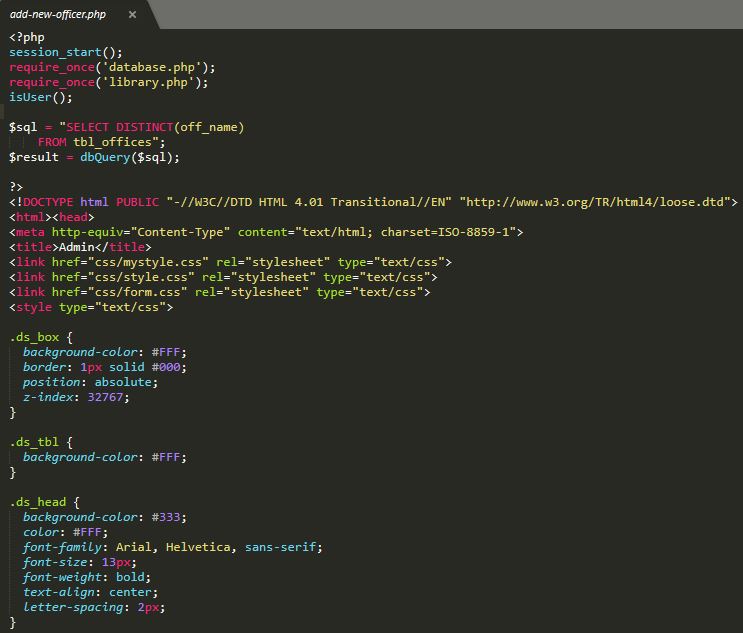




Figure **Code of add new office**

**Add Manager**





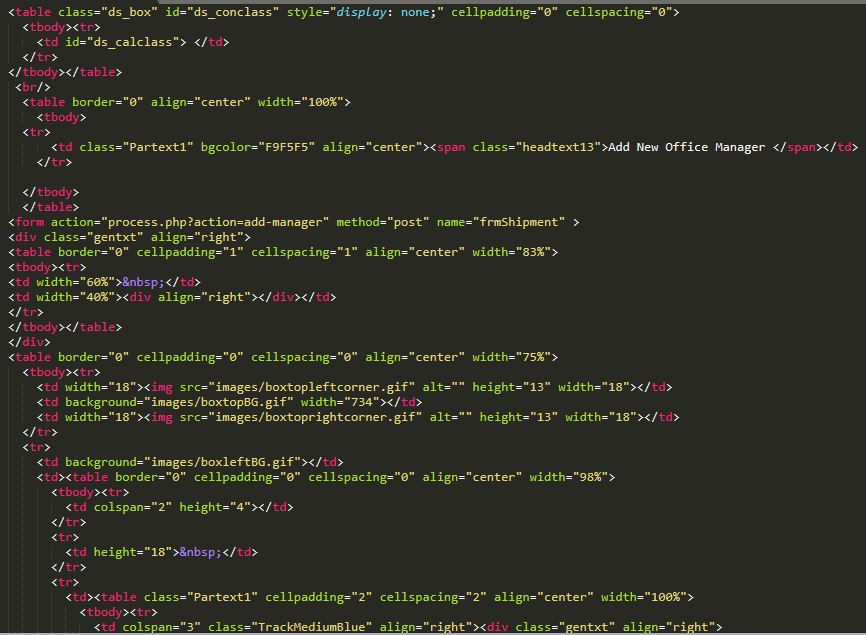
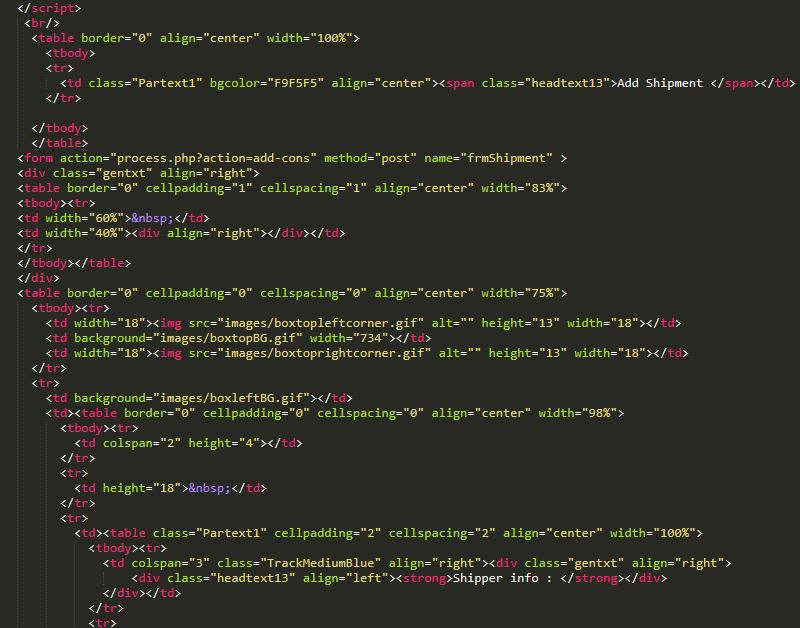


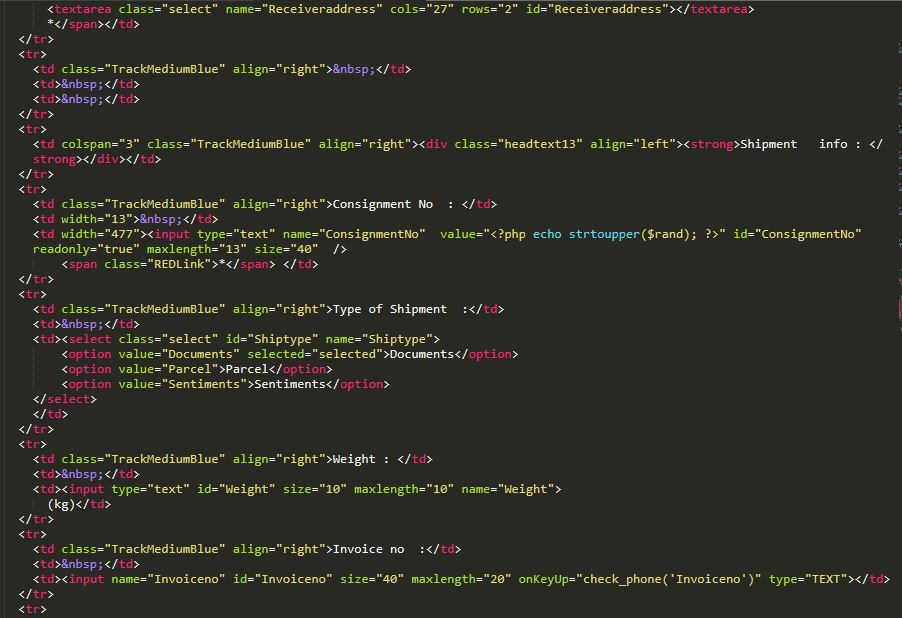


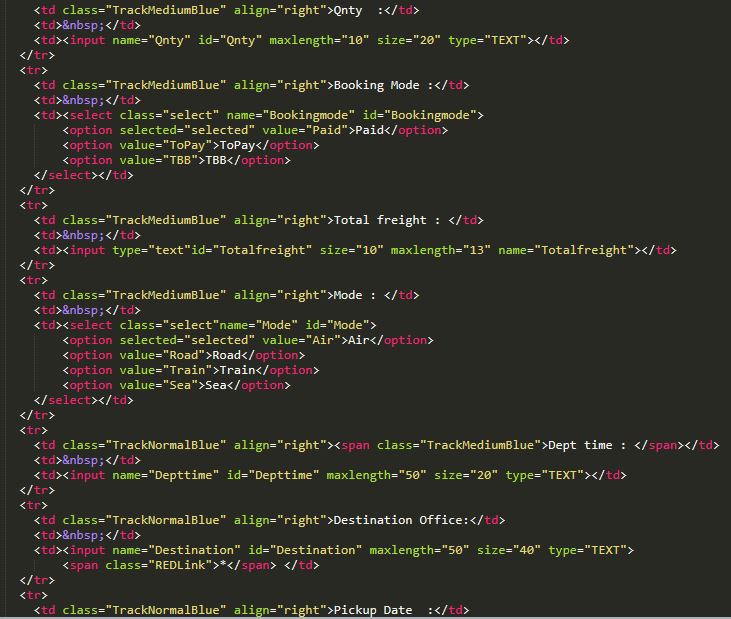


Figure **Code of add manager**

**Add Shipment**







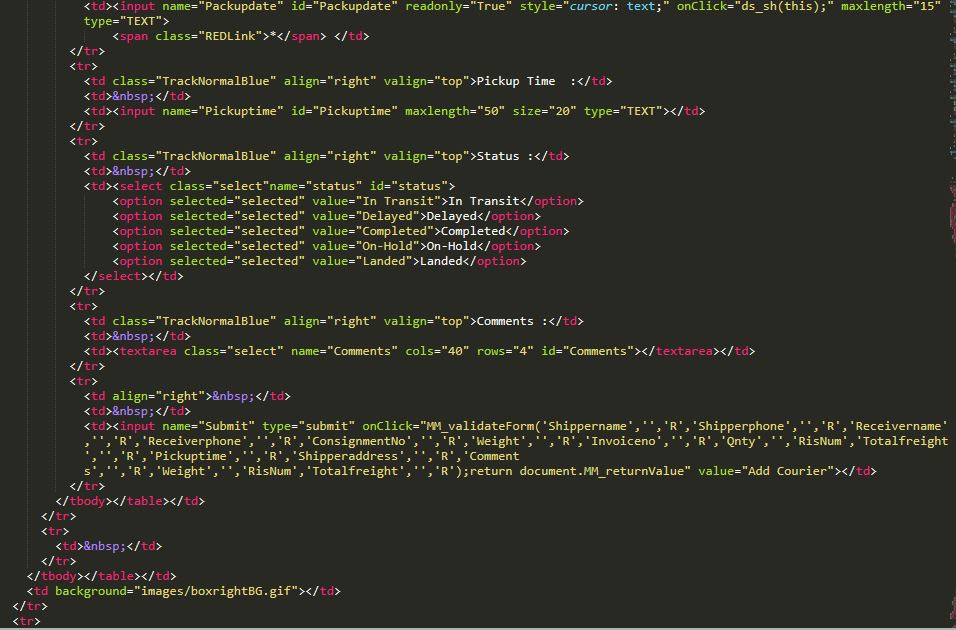
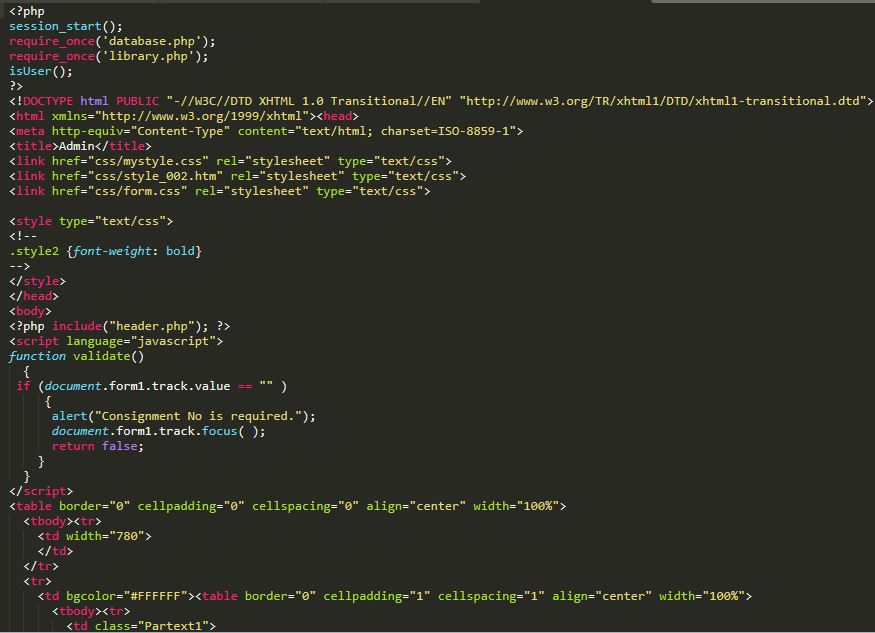


Figure **Code of add shipment**

**Search Courier**



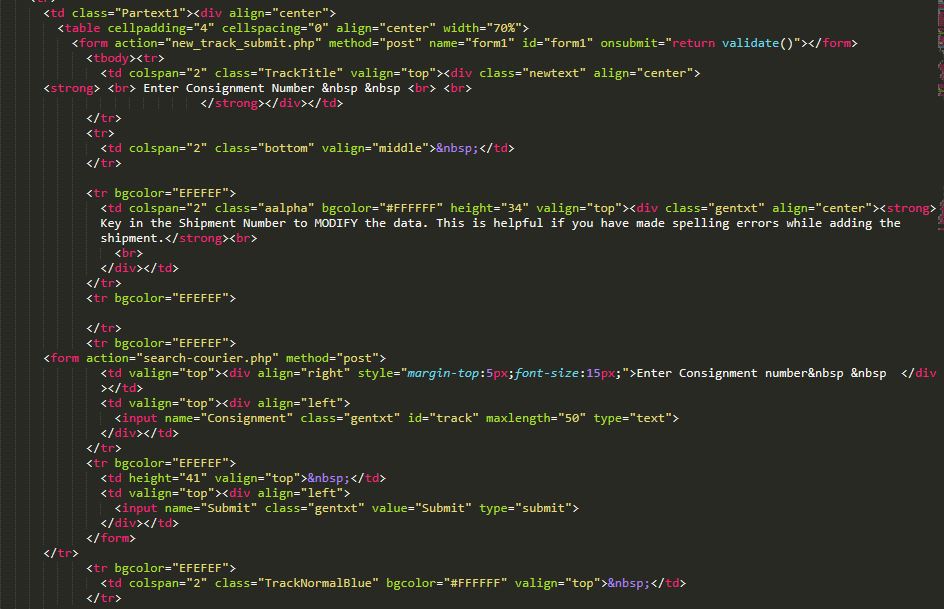


Figure **Code of Search courier**