# Chapter 1: Introduction

## Chapter 1.1: Project Introduction

This project **Royal Tattoo Service** is an online service where clients or customers can easily book tattoo artist for several hours and can also browse tattoo’s as well as tattoo products such as ink machines, etc. in the website. This system makes booking of artist’s a lot less easy as customers can easily find artist and can easily buy quality number of products and tattoos.

## Chapter 1.2: Justification and Background of the Project

In Nepal there are not many tattoo services where customers can easily buy tattoos, products and book artists. Even though there are few of them customers cannot book artists properly as some have bugs while others interface design is bad and some are just not good. Some of the websites I have seen does not even work properly and booking is a bit off and have many processes to complete. So, in order to make booking a lot easier and faster and benefit the customers I have made this project.

## Chapter 1.3: Overview of the Project

Royal Tattoo Service is a web application where users can book artists, buy tattoos from artists and also innovate their own design. This system is compatible with mobile devices and performs very well. Here in this system authentication is needed so that only registered users can book artists or buy tattoos and products. After buying these tattoos users can check their transaction details in their dashboard where they will find all of the transaction done by them.

## Chapter 1.4: Aims and Objectives

## Aims

* To make it easier for customers to book tattoo artists and even buy tattoos.
* Helping the sellers of tattoo products in Nepal.
* Provide user satisfaction while using the website.

## Objectives

* Making of a user-friendly system.
* Providing of detailed information about product and artists.
* Making of booking and transaction automated.
* Enabling of importing photos of a better design of tattoo for customers.

# Chapter 2: Analysis

## Chapter 2.1: Introduction

Simply analysis is a thorough study of anything complex in order to have a better understanding of it. In software development it is the first phase of the life cycle. It is used to find the best way or steps that can be used for the completion of the system. Analysis helps in identifying of the requirements needed for the system to succeed, which helps in developing of system easier as everything is understood. It also helps in maximizing the overall quality of the system. It can be used in many ways like identifying of issues and bugs that may appear at the development stage of a system.

## Chapter 2.2: Analysis Methodology

It is simply an organized, conceptual and a well-managed process for the analysis of requirements. There are numerous procedures some are Soft approach analysis methodology, Hard approach analysis methodology and Combined approach analysis methodology. Here combined approach is a fusion of Hard approach and Soft approach.

Here for this system I have used a Hard approach because it focuses mainly on the technical aspects. This approach is also for large complex systems but it can also be used in smaller systems. This approach uses SSADM (Structured System Analysis and Design Methodology). There are three views of a system according to SSADM:

**The Process View** which outlines all the functions that are carried out by the system and it also shows how the inputted data is moving around the system and how it is actually being processed.

**The Data View** outlines each and every data and information that the system uses.

**The Event View** outlines all the events that sets the process running in a system and also the effect of outer events on the data of the system.

There are several advantages of SSADM they are:

* Each step needs to be completed before progressing on towards the next one.
* It assures thorough planning of the system which can help in making of a better system.

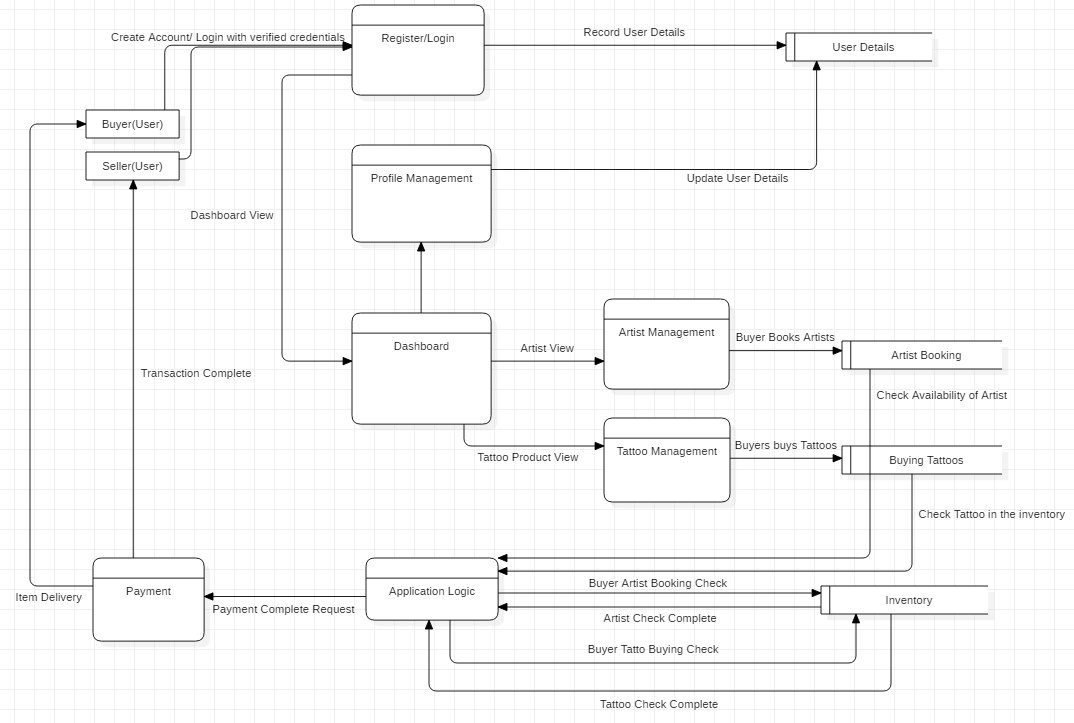


Figure Data Flow Diagram

## Chapter 2.3: Feasibility Study

It is a part of analysis which examines and decides whether or not a project is legally, financially, technically and socially feasible. To evaluate it a feasibility study is from where information such as resource availability, cost estimation for the development of the system, benefits of the system after it is developed and maintenance costs are mostly examined during the study. There are few of the areas that are examined by the study they are: -

* Technical Feasibility

It focuses mainly on the accessibility of technical resources in the development phase of the project. There are technical resources needed which is capable of developing of system like software, good hardware, storage etc. Since I have the technical resources that are needed my system is feasible.

* Social Feasibility

It focuses mainly on the social factors such as political condition, surroundings of the area in which is covered by the system. There are rules and restrictions held in the area which effects the system. This system may not affect the society in any way possible so it is clearly feasible.

* Legal Feasibility

It focuses mainly on whether or not the system is restricted within the border of legal requirements such as law of data protection, Copyright Law, Social media Law etc. Because of these rules it can help us enhance in the development of the system. Since there are no rules broken and all functions are considered legal my system is feasible.

* Financial Feasibility

It focuses mainly on the cost of the system whether or not the system has been cost- effective. The study used which is cost-benefit analysis views all the benefits of the system. Since I am doing this project for academic purposes there are no financial issues so my system is feasible.

## Chapter 2.4: SRS (Software Requirement Specification)

SRS is a set of documentation that describes the features as well as the behaviors of a system. It contains the required functionality within the system for user satisfaction. It has functional and non-functional requirements from which a system is developed.

## Chapter 2.4.1: Functional Requirement

Functional Requirements are the requirements that define what the system should do. It includes lots of functions and features to meet the requirement specification of the users and make them satisfied. There are many functional requirements involved in my system they are: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FR ID | FR Description | Data required | Rational | Dependency |
| FR1 | Admin signup | Name, username, password, email, phone number | Admin login data. |  |
| FR2 | Admin login | Username, password | Security and privacy. | FR1 |
| FR3 | Upload Tattoos and artists | Name, price, quality, manufactured date, details | Uploading tattoos made by artists and artists that are available for booking. | FR1, FR2 |
| FR4 | View Tattoos and artists | Name, price, quality, manufactured date, details | Viewing available tattoos and artists. | FR1, FR2 |
| FR5 | Update Tattoos and artists | Name, price, quality, manufactured date, details | Updating data of tattoos and artists. | FR1, FR2, FR4 |
| FR6 | Delete Tattoos and artists | Name, price, quality, manufactured date, details | Updating data of tattoos and artists. | FR1, FR2, FR4 |
| FR7 | View user details | Name of user, username, email | Get data of users and their interaction with the system. | FR1, FR2 |
| FR8 | Deploy message about payment completion. | Email, message, time, schedule | Inform users about their completion of transaction. | FR1, FR2 |
| FR9 | Manage users | Name of user, username, email | Suspend and delete unwanted and untrusted users. | FR1, FR2, FR7 |
| FR10 | User registration | Name, username, password, email, phone number | Users login data. |  |
| FR11 | User login | username, password | Security and privacy. | FR10 |
| FR12 | View artist booked schedules | Username, time, date | View time of artist booked by which user. | FR10, FR11 or FR1, FR2 |
| FR13 | View All Tattoos | Name, Price, Quality, Details | View all the available tattoos. | FR10, FR11 or FR1, FR2 |
| FR14 | Book artists | Username, time, date | Book the artist for a specified date and time. | FR10, FR11, FR12 |
| FR15 | Buy Tattoos | Username, price, details | Buy the tattoos at any time possible. | FR10, FR11, FR13 |
| FR16 | Manage profile | Name, username, password, email, phone number | Update and change the personal data and details of user | FR10, FR11 |
| FR17 | Contact with Admin | Username, message. | Contact with admins through chat or contact forms for informative purpose. | FR10, FR11 |

## Chapter 2.4.2: Non-Functional Requirement

Non-Functional Requirement specifies on how the system should behave within the restriction. They are very important for development as it helps in making of a system better in many ways such as robustness, effectiveness, reliability, fast, secure and User- Friendly. These are some of the important non-functional requirements: -

|  |  |  |
| --- | --- | --- |
| ID | NFR Title | NFR Description |
| 1 | Security | It is one of the most important function in a system as users must be authenticated by the system with strong passwords with good encryption algorithm and only valid users can login. |
| 2 | Performance | The system should have 0 latency and lagging of system should rarely happen. The system should be able to process the inputs of the users at real time. |
| 3 | User friendly | The system should be easy to use with a good user interface. There should be a system guide which guide’s new users about the system and how to use it. |
| 4 | Reliability | The system should have 0% failures and should run all the time. It should be very reliable. |
| 5 | Availability | The system should be easily accessible and available everywhere. |
| 6 | Scalability | The system should work on all the devices ranging from mobile devices to laptops that have large screens and resolution. |
| 7 | Data Integrity | The system should secure the important data and content of the website where only admins should have authorization to change or delete the data. |
| 8 | Maintainability | The system should be easy to maintain. There should be a successful repair action on problems appearing in the system. |

## Chapter 2.4.3: Moscow Prioritization

It is simply a process of arranging of functions according to their importance in the system. In this prioritization the functions are arranged as must have, should have, could have and would have kept in the system. Here is the prioritization of the functions: -

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Features | Prioritization | Rational |
| 1 | User/Admin registration | Must have | Fundamental function within the system. |
| 2 | User/Admin Login | Must have | Fundamental function within the system. |
| 3 | Upload /Update/Delete Products | Must have | Fundamental function within the system. |
| 4 | Manage Users | Should have | For convenient data handling. |
| 5 | Inform Users after completion of transaction. | Should have | Fundamental function within the system. |
| 6 | Manage Profile | Could have | For convenient data handling. |
| 7 | Buy Tattoos | Must have | Fundamental function within the system. |
| 8 | Book Artists | Must have | Fundamental function within the system. |
| 9 | View User Details | Would have | For convenient data handling. |
| 10 | Chat with admins | Would have | For convenient data handling. |
| 11 | Security | Must have | Fundamental function within the system |
| 12 | Reliability | Must have | Fundamental function within the system |
| 13 | User friendly | Must have | Fundamental function within the system |
| 14 | Performance | Must have | Fundamental function within the system |
| 15 | Maintainability | Must have | Fundamental function within the system |
| 16 | View All Tattoos | Must have | Fundamental function within the system. |
| 17 | View artist booked schedules | Must have | Fundamental function within the system. |
| 18 | Deploy message about payment completion. | Must have | Fundamental function within the system. |
| 19 | Data Integrity | Must have | Fundamental function within the system |
| 20 | Scalability | Must have | Fundamental function within the system |
| 21 | Availability | Must have | Fundamental function within the system |

## Chapter 2.4.4: Hardware/Software Specification

Some of the hardware and software required for designing and developing of the project. The hardware and software required for Royal Tattoo Service are given below: -

Hardware Specifications:

Processor: CPU with 1.5MHz of higher.

Ram:2 GB or higher

Hard disk: 20 GB or higher

Display Type: Standard VGA

Software Specifications:

Operating System: Windows 7 or higher

Front - End: Bootstrap

Back - End: PHP, MYSQL, XAMPP.

## Chapter 2.5: Use Case Diagram

Use Case Diagram is a dynamic diagram which shows the interaction of the user with the system. It shows the interactions between actors, system and use cases. The main components that are used in this diagram are:

Actors: Actors are the users of the system and they are represented by stick figures.

Use Cases: It refers to the function of the system or role of actors and they are represented by oval.

Associations: It refers to the relation of actors with their related use cases.

System Boundary Boxes: It is represented by a rectangular box.

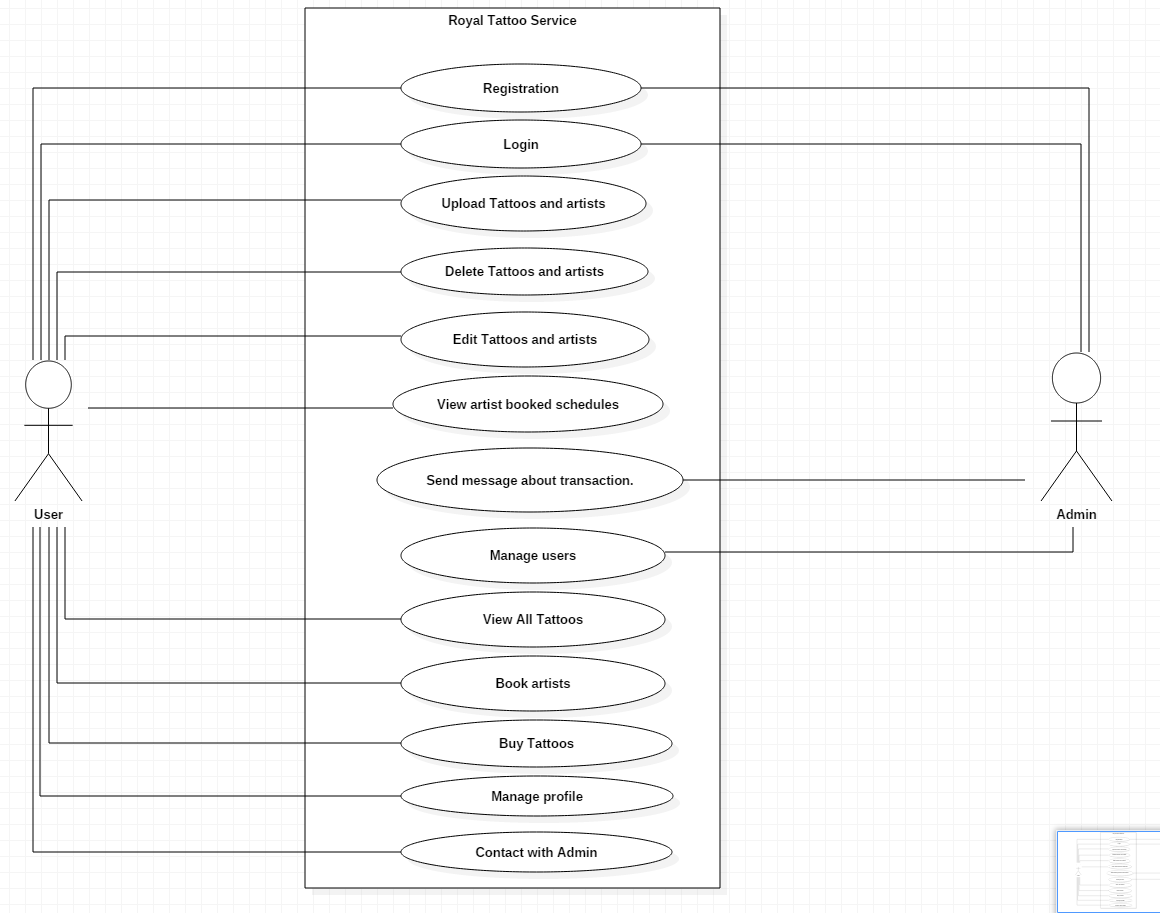


Figure Use Case Diagram

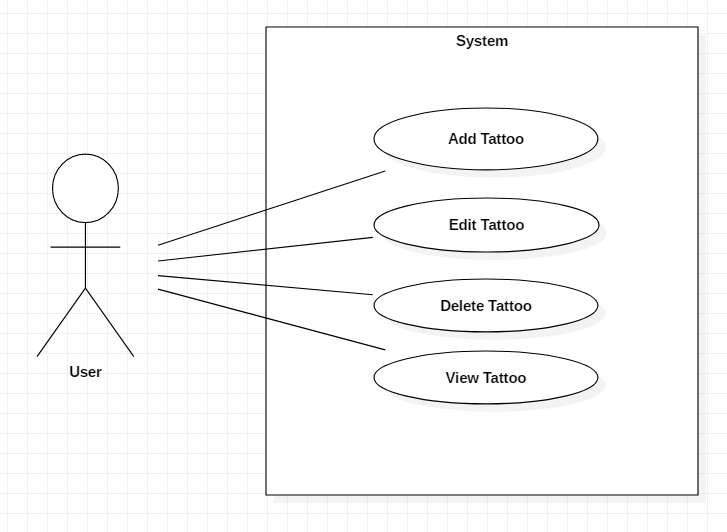


Figure Use Case Diagram CRUDE

Title: “Add Tattoo”

|  |  |
| --- | --- |
| ID | AT3 |
| Justification | It is an important use case which adds data into the system. |
| Primary Actor | User |
| Secondary Actor | NA |
| Primary Flow | 1. User logs into the system with valid credentials. 2. User navigates into the View tattoo page. 3. User then clicks on the corresponding button for adding of tattoo. 4. User inputs all the necessary data needed in each field and submits the form. 5. System then validates and verifies the valid input. 6. System adds the data of the form into the database. 7. System then redirects to a page displaying all the tattoos. |
| Alternative Flow | 5.1 User inputs invalid data into the field of the form.  5.1.1 System shows that the data is incorrect or invalid.  5.1.2 User inputs the values again into the form.  5.1.3 Repeating of process from 4.1 until data is valid and correct. |

Title: “Edit Tattoo”

|  |  |
| --- | --- |
| ID | AT4 |
| Justification | It is an important use case which updates the data within the database of the system. Used for correction of data or changing of data. |
| Primary Actor | User |
| Secondary Actor | NA |
| Primary Flow | 1. User logs into the system with valid credentials. 2. User navigates into the View tattoo page. 3. User then clicks on the corresponding button for updating of data of the tattoo. 4. System then redirects to an update form where admin can update data. 5. User edits the data that needs to be updated and submits the form. 6. System then validates and verifies the valid input. 7. System updates the data of the form into the database. 8. System then redirects to a page displaying all the tattoos. |
| Alternative Flow | 6.1 User inputs invalid data into the field of the form.  6.1.1 System shows that the data is incorrect or invalid.  6.1.2 User inputs the values again into the form.  6.1.3 Repeating of process from 4.1 until data is valid and correct. |

Title: “Delete Tattoo”

|  |  |
| --- | --- |
| ID | AT5 |
| Justification | It is an important use case which deletes data within the database of the system. Used for deleting huge and irrelevant data from the system. |
| Primary Actor | User |
| Secondary Actor | NA |
| Primary Flow | 1. User logs into the system with valid credentials. 2. User navigates into the View tattoo page. 3. User then clicks on the corresponding button for deleting of data from the system. 4. System shows a dialog box which shows if you want to delete it. 5. User confirms to delete the data. 6. System deletes the data form the database. 7. System then redirects to a page displaying all the tattoos. |
| Alternative Flow | 5.1 User declines to delete the data.  5.1.1 System then redirects to a page displaying all the tattoos. |

Title: “View Tattoo”

|  |  |
| --- | --- |
| ID | AT6 |
| Justification | It is an important use case which views all the data from the database. |
| Primary Actor | User |
| Secondary Actor | NA |
| Primary Flow | 1. User logs into the system with valid credentials. 2. User navigates into the View tattoo page. 3. System redirects to a page displaying all the tattoos. |
| Alternative Flow | NA |

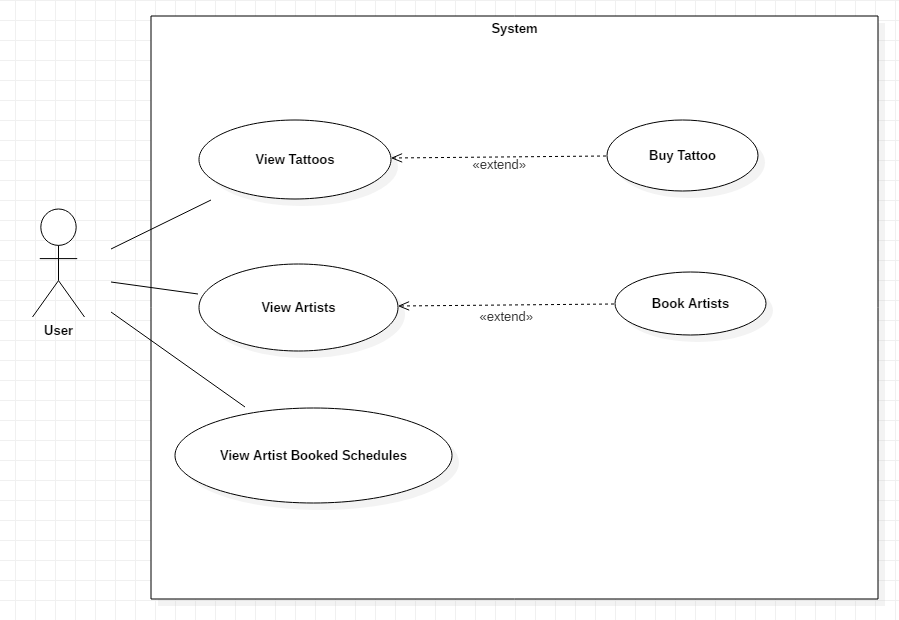


Figure Use Case Diagram Viewing Artist and Tattoos

Title: “View Artists”

|  |  |
| --- | --- |
| ID | AT7 |
| Justification | It is an important use case which views all the artists available in the system. Here user can book artists. |
| Primary Actor | User |
| Secondary Actor | NA |
| Primary Flow | 1. User logs into the system with valid credentials. 2. User navigates into the View Artists page. 3. System shows all the current Artists in the system. 4. User then clicks on the corresponding button for booking of artist. 5. System shows a dialog box which shows if you want to book it. 6. User confirms to book the artist. 7. System books the artist and updates the data into the database. 8. System then redirects to a page displaying all the artists. |
| Alternative Flow | 5.1 User declines to book the data.  5.1.1 System then redirects to a page displaying all the artists. |

Title: “View Booked Artists”

|  |  |
| --- | --- |
| ID | AT8 |
| Justification | It is an important use case which shows all the booked artists within the system. Used for seeing if that artist is available or not. |
| Primary Actor | User |
| Secondary Actor | NA |
| Primary Flow | 1. User logs into the system with valid credentials. 2. User navigates into the View artists page. 3. System shows all the current Artists in the system. 4. User then clicks on the corresponding button for viewing booked artists. 5. System redirects to a page displaying all the booked artists. |
| Alternative Flow | NA |

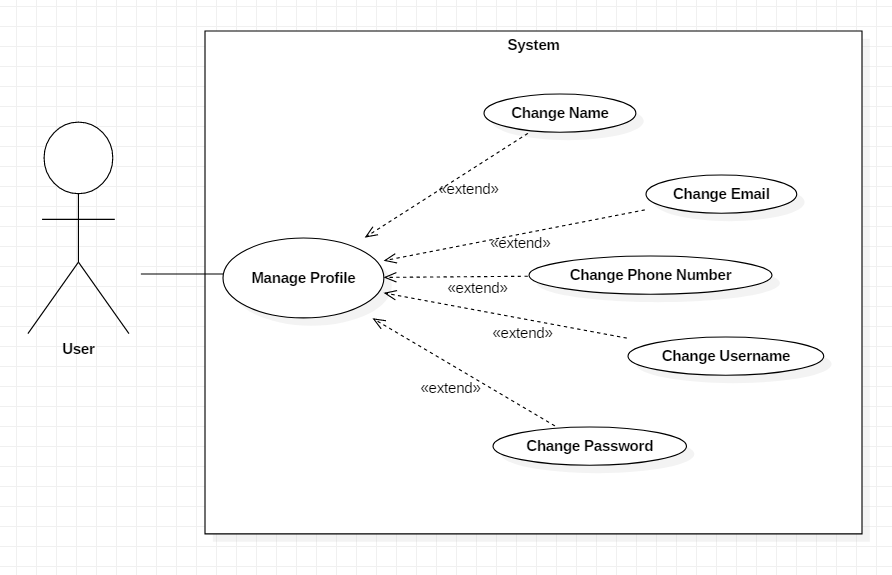


Figure Use Case Manage Profiles

Title: “Manage Profiles”

|  |  |
| --- | --- |
| ID | AT9 |
| Justification | It is an important use case which manages the profiles of the users. |
| Primary Actor | Users |
| Secondary Actor | NA |
| Primary Flow | 1. User logs into the system with valid credentials. 2. User navigates into the Profile page. 3. User then clicks on the corresponding button for editing of information. 4. User inputs all the necessary data needed in each field and submits the form. 5. System then validates and verifies the valid input. 6. System updates the data of the form into the database. 7. System then redirects to the home page. |
| Alternative Flow | 5.1 Admin inputs invalid data into the field of the form.  5.1.1 System shows that the data is incorrect or invalid.  5.1.2 Admin inputs the values again into the form.  5.1.3 Repeating of process from 4.1 until data is valid and correct. |



Figure Use Case Manage Users

Title: “Manage Users”

|  |  |
| --- | --- |
| ID | AT10 |
| Justification | It is an important use case which manages user according to their behavior in the system. |
| Primary Actor | Admin |
| Secondary Actor | NA |
| Primary Flow | 1. Admin logs into the system with valid credentials. 2. Admin navigates into the View Users page. 3. System shows logs of Users. 4. Admin sees the logs and behaviors of users. 5. Admin then manage user according to that log. |
| Alternative Flow | * 1. Admin finds out bad behavior with low impact      1. Admin temporarily bans the user.      2. System Deactivates the user for a certain time.      3. System Redirects to the View Users Page.   5.2. Admin finds out bad behavior with high impact.   * + 1. Admin permanently bans the user.     2. System Deactivates the user for a certain time.   5.2.3 System Redirects to the View Users Page. |

## Chapter 2.6: Initial Class Diagram (NLA)

NLA is defined as association of human language usable as language where computers can understand, interpret and manipulate it. There are three important things they are nouns as classes, verbs as functions and adjectives as attributes.

|  |  |  |
| --- | --- | --- |
| Nouns (Classes) | Verb (Function) | Adjective (Attributes) |
| Admin | Signup, Login, View order, Add Tattoos | Admin Id, Admin Name |
| User | Book Artists, View Tattoos | User Id, Name, Username, Email |
| Tattoo | Add Tattoo, Update Tattoo, Delete Tattoo, View Tattoo | Tattoo id, Tattoo Name, Details, Price, Manufactured Date |
| Artists | Book Artists, View Booked Schedules | Date, Time, Username |
| Ordered Items | Create Order, Update Order, Cancel Order | Username, Tattoo id, Ordered Date |

## Chapter 2.7: Class Diagram

A Class Diagram is a static diagram which shows a static representation of a system. It involves structures like Classes, Attributes and Operations. It also shows the relationship between all the classes within the system. Through the help of NLA this is the class diagram of my system that I have created:

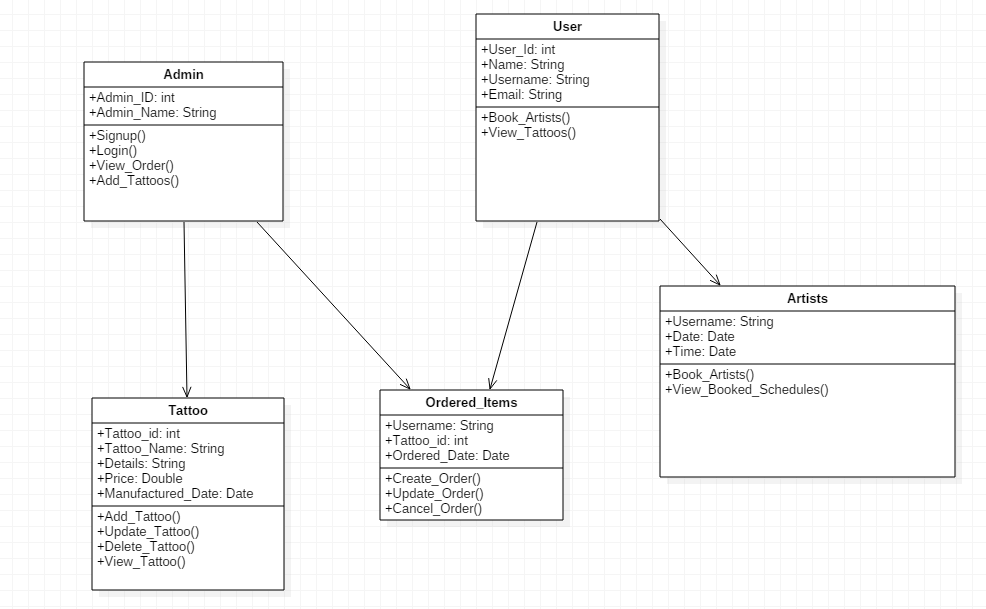


Figure Initial Class Diagram

# Chapter 3: Design

## Chapter 3.1: Introduction to Design

Design is an important phase of development process in which developers develop the system based on the specific requirements and needs of their customers.

## Chapter 3.2: Structural Design/Model

Structural design which is mostly based on mathematical model, statistical model as well as the computer algorithms.

## Chapter 3.2.1: Final Class Diagram

A Class Diagram is a static diagram which shows a static representation of a system. It involves structures like Classes, Attributes and Operations. It also shows the relationship between all the classes within the system. It shows all the connections between the views, controllers as well as the models of the system. It shows which view is connected to which controller and which controller is connected to which model for instance User view is connected to User Controller and User Controller is connected to the User Model.

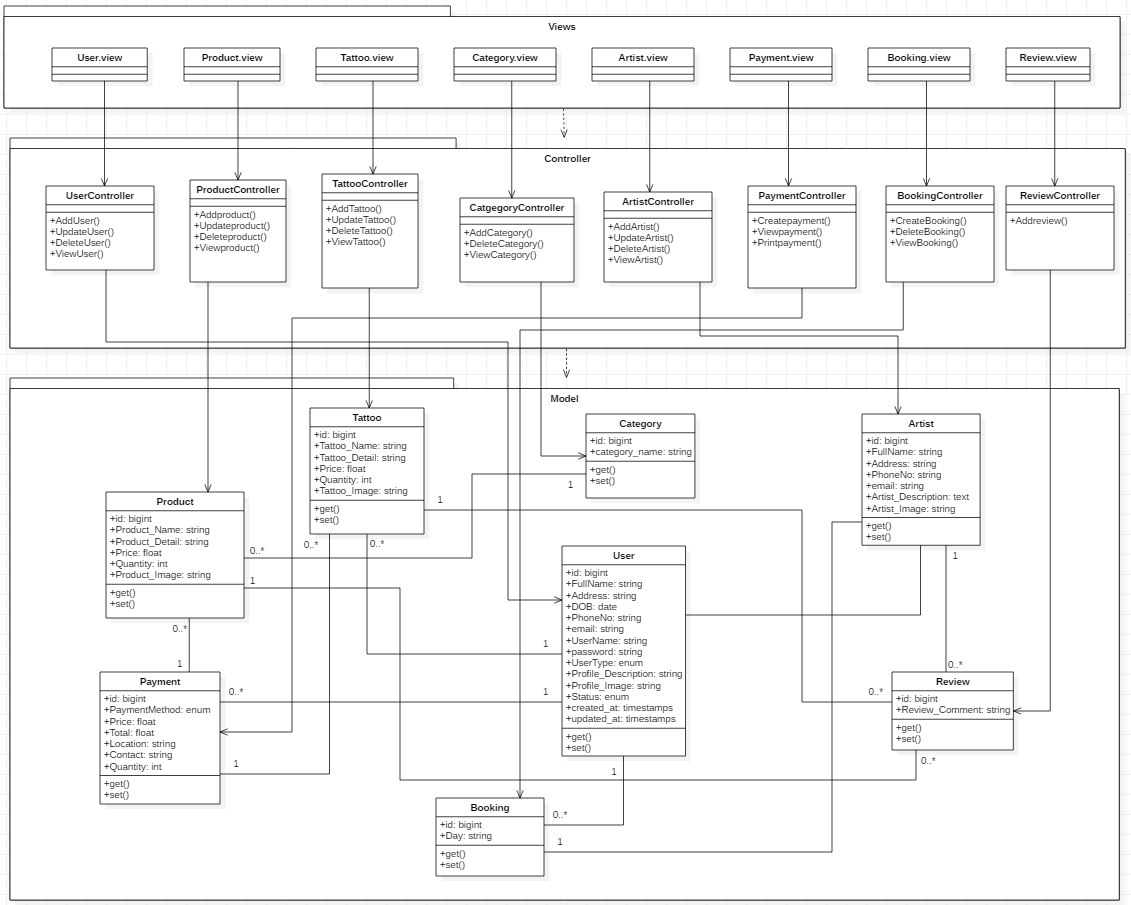


Figure Final Class Diagram

## Chapter 3.3: Behavioral Model

Behavioral Model are basically a dynamic model which shows how the system responds according to the actions performed from the environment or users.

## Chapter 3.3.1: Activity Diagram

Activity Diagram is an important behavioral diagram which narrates the dynamic aspects of the system. It is an advanced version of flow chart which is used to represent the flow from one activity to the other.

|  |  |  |
| --- | --- | --- |
| Notation Used | Notation | Description |
| Action |  | Shows what activity is done |
| Initial |  | Shows the starting of the activity |
| Final |  | Shows the ending of the activity |
| Fork |  | Splits one flow into multiple flow |
| Join |  | Joins two actions into one |
| Decision |  | Decides the condition |
| Control Flow |  | Shows the flow of activity |
| Send Signal |  | Sends message through the system |
| Accept Signal |  | Receive message sent from the system |
| Swimlane |  | Group related activities into one column vertical or horizontal |
| Activity Interrupt |  | Interrupts the flow of the activity with a lightning bolt |
| Accept Time Event |  | Event which stops the flow of an activity for a certain specific time |
| Interruptible Activity Region |  | Activity terminated if interruption occurs. |

Here are all of my activity diagrams:-

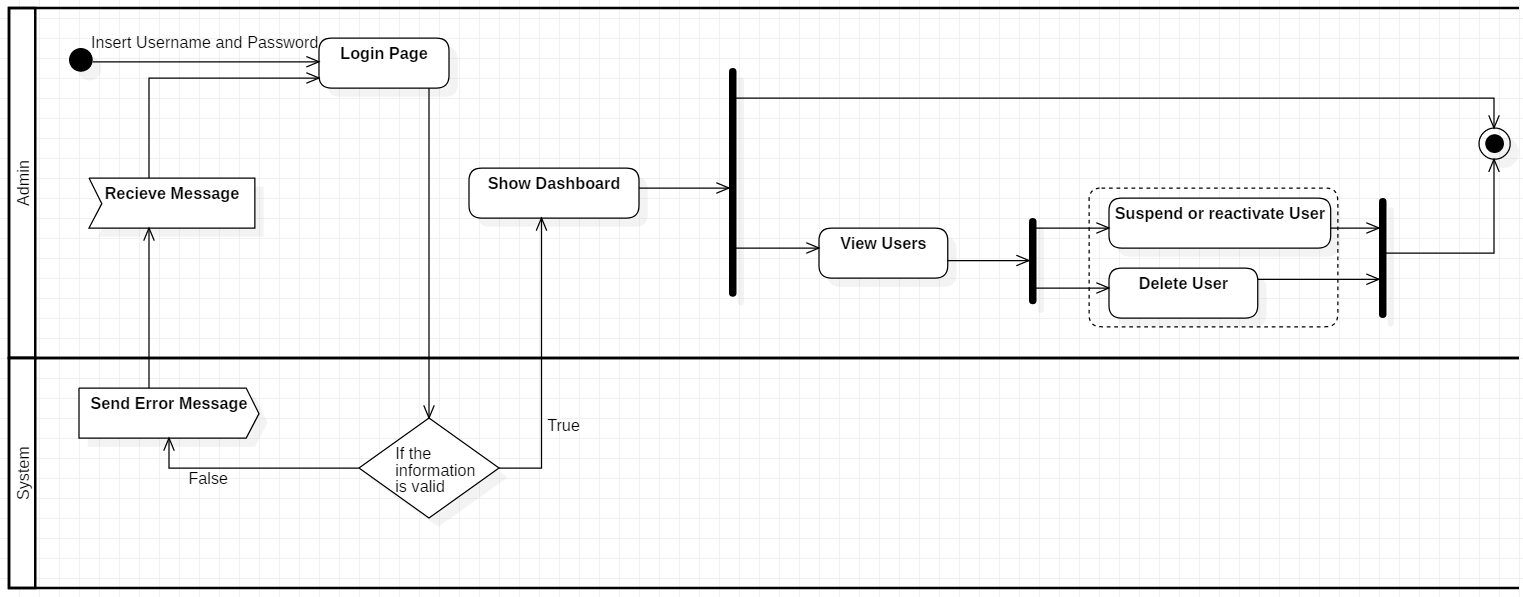


Figure Admin Login Activity Diagram

The above activity diagram is about how an admin of the system works and can operate its functions. First admin opens up a login page where he/she provides the required information i.e username and password, if the provided information is incorrect the system sends a error message to the login page(Looping is done if information is not correct). If provided information is correct the admin is redirected to the admin dashboard where he can view users and can suspend,delete as well as reactivate users.

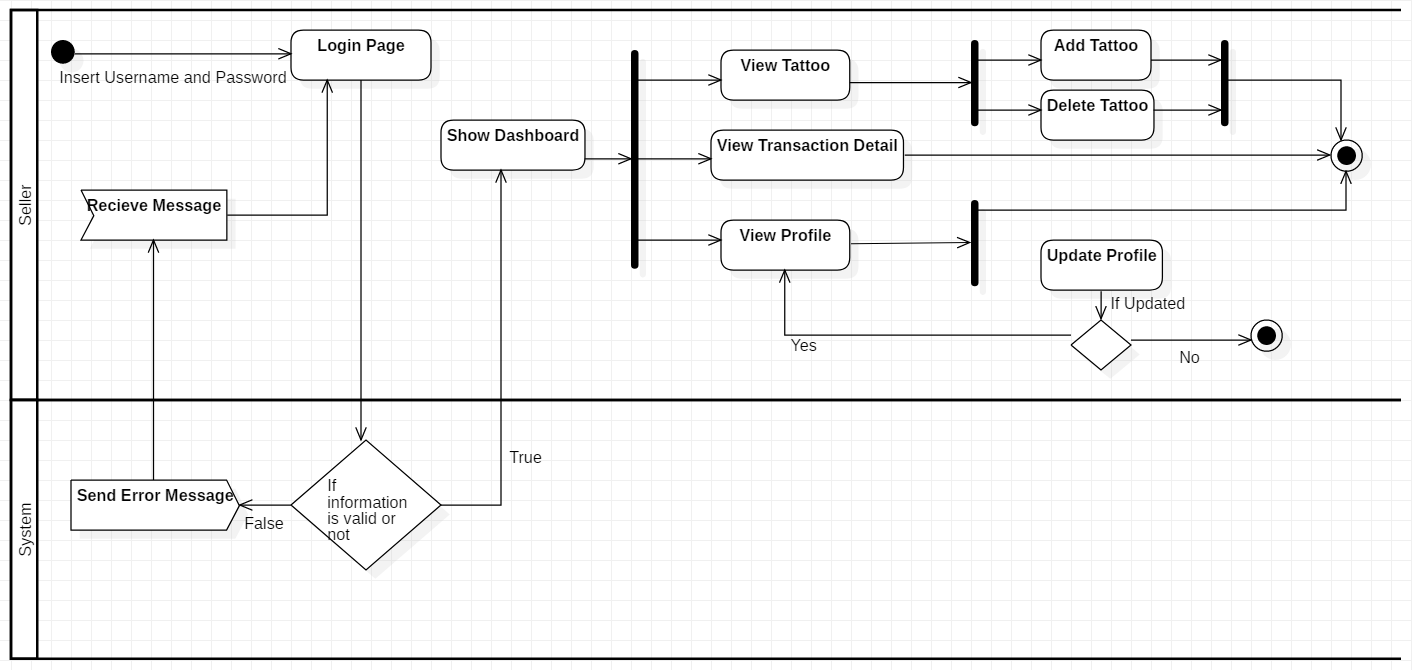


Figure Seller Login Activity Diagram

The above activity diagram is abot how an seller (which is a artist) of the system works and can operate its functions. First seller opens up a login page where he/she provides the required information i.e username and password, if the provided information is incorrect the system sends a error message to the login page(Looping is done if information is not correct). If the provided information are correct the artist is redirected to the artist dashboard where he/she can view tattoos, view transaction details and view profiles. In tattoos artist can delete as well as add tattoos and artist and update their profile as well as upload a photo.

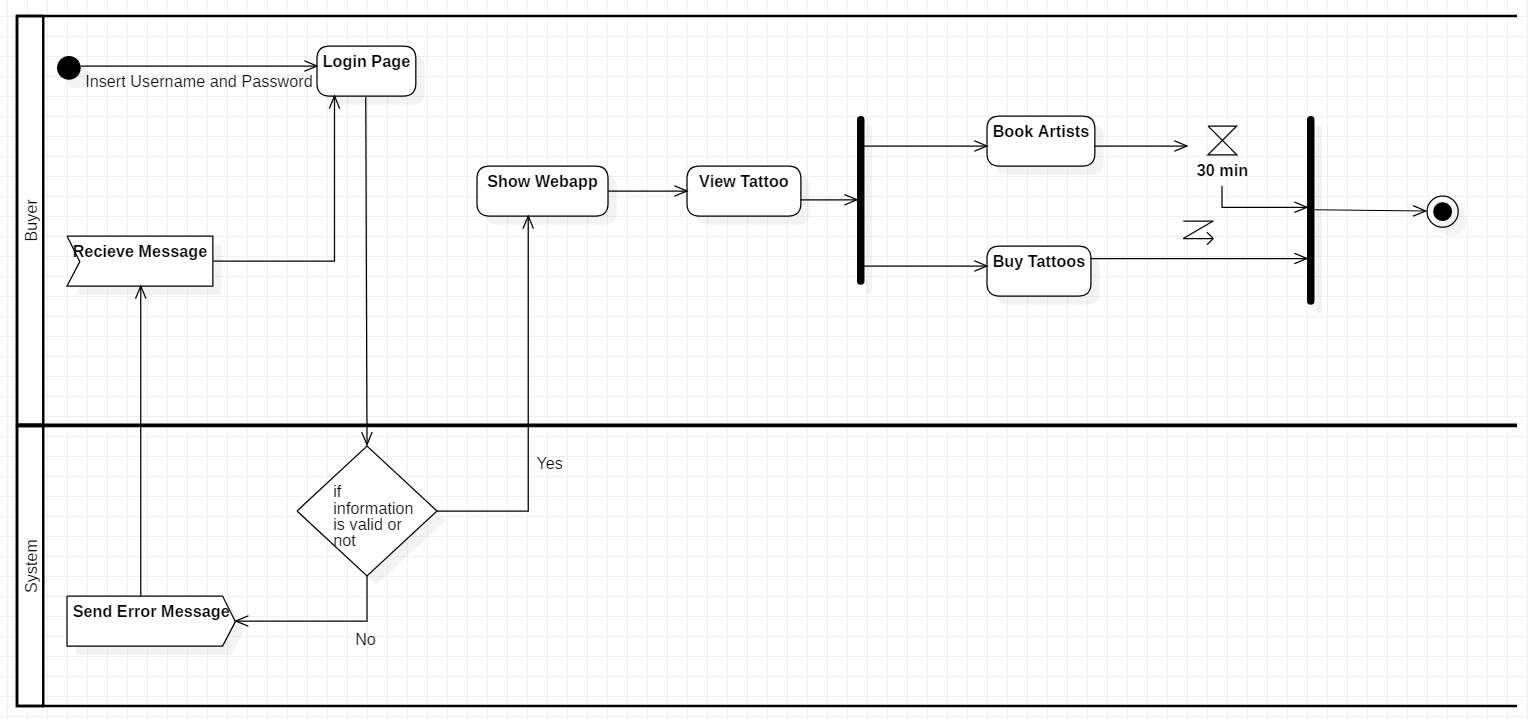


Figure User Login Activity Diagram

The above activity diagram is abot how an Buyer (which is a normal user) of the system works and can operate its functions. First buyer opens up a login page where he/she provides the required information i.e username and password, if the provided information is incorrect the system sends a error message to the login page(Looping is done if information is not correct). If the provided information are correct the User is redirected to the webapp where he/she can view tattoo and buy it as well, users can also book artists in the artist section.

## Chapter 3.3.2: Sequence Diagram

Sequence Diagram is a diagram which shows how actors interact with objects in a sequential order. It shows how and what order the objects in the system function.

|  |  |  |
| --- | --- | --- |
| Notation Used | Notation | Description |
| Actor |  | It is an entity which interacts with the objects of the diagram. |
| Lifeline |  | It interacts with other objects during the sequence. |
| Message Arrow |  | Shows the flow of the diagram. |
| Alternate Frame |  | It is if else statement. |
| Loop Frame |  | It is a loop statement used for repetition of a part. |

Here are my sequence diagrams: -

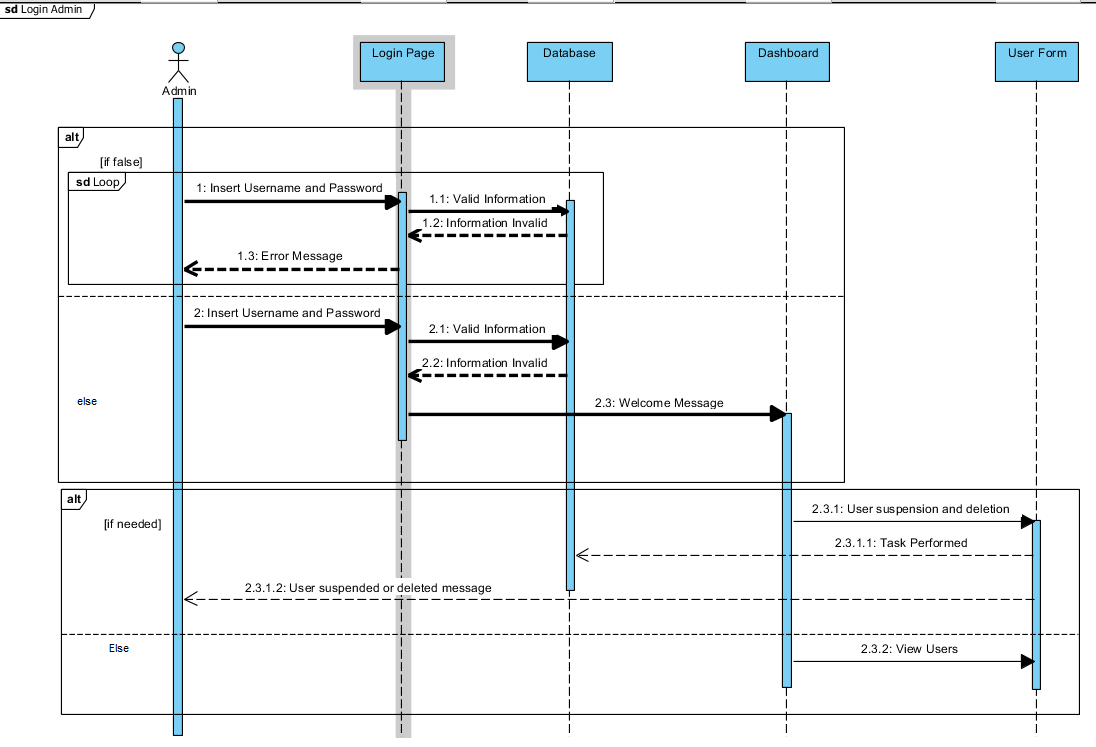


Figure Admin Login Sequence

The above sequence diagram is about how a admin logs into the system. First the admin enters the login page where he/she provides the required information. If correct then admin is redirected into the admin dashboard where admin can suspend, reactivate as well as delete users.

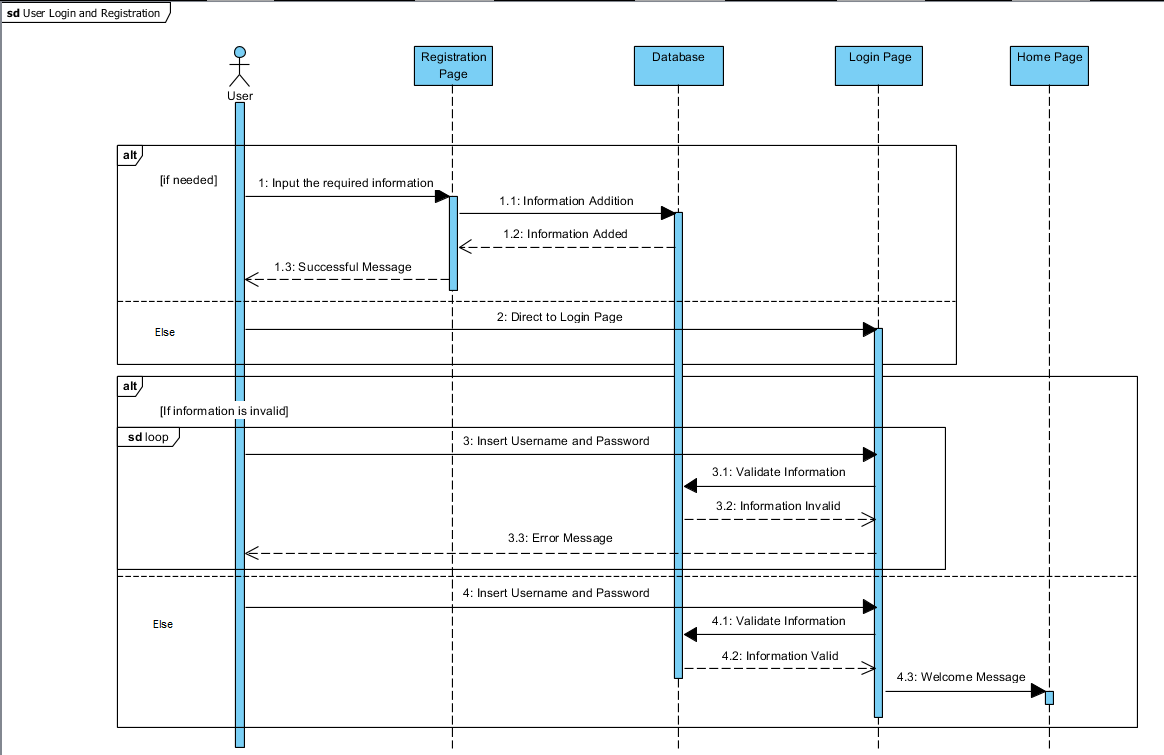


Figure User Register and Login Sequence

The above sequence diagram is about how a User registers itself and logs into the system. First the user enters the registration page where he/she provides the required information which is then stored into the database. Then user is redirected to the login page where he/she provides the required information. If correct then user is redirected into the home page of the webapp where user can view tattoos and artists as well as buy tattoos and book artists.

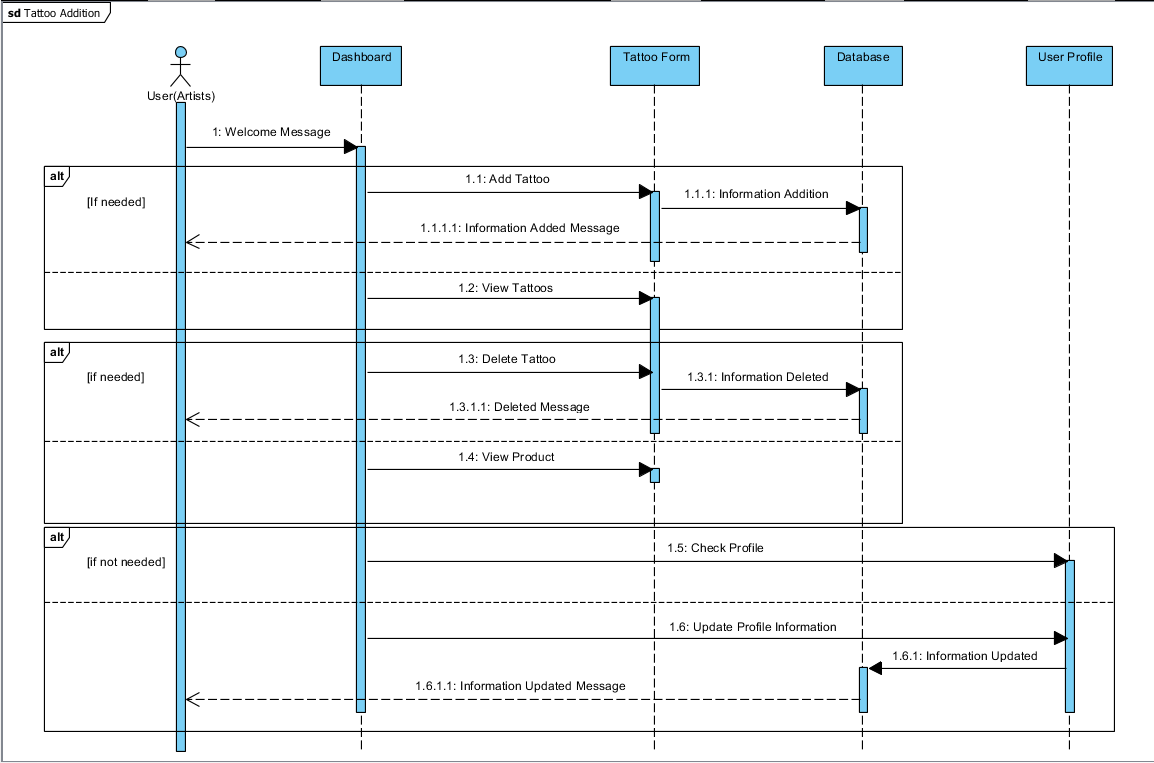


Figure Tattoo Sequence

The above sequence diagram is about how a Artist adds tattoos into the system. First the artist enters the artist dashboard from there artist enters the view tattoo page where artist can find a tattoo form from which the artists can add tattoos which is stored in the database, artists can delete their tattoos as well but cannot delete other artists tattos. Here artists can check their profile and add information about them or update details or even add photos.

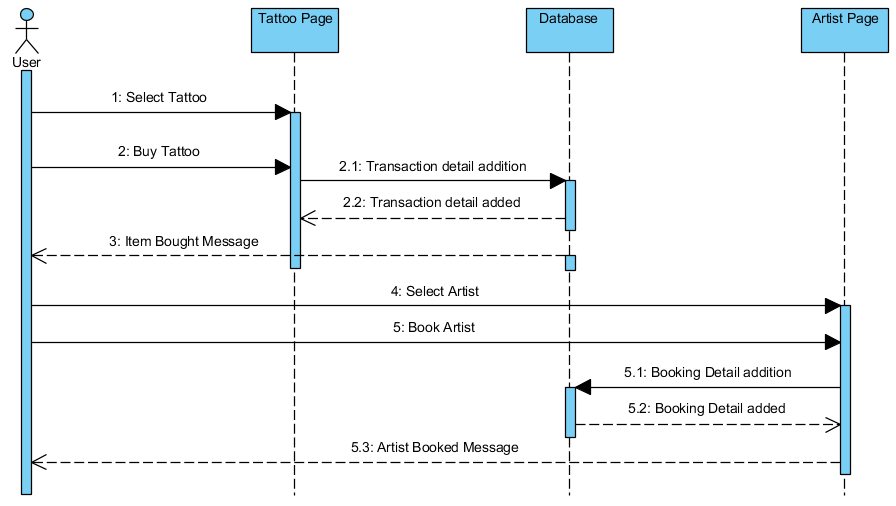


Figure Buying Sequence

The above sequence diagram is about how a user can buy tattoos and book artists from the system. For buying tattoo first the user enters the tattoo pageers where all the available tattoos are shown from their user can select one and buy it. From there the transaction details are stored and bought message is sent to the user. Now for booking artist first the user enters the artist page where all the available artists are shown from their user can select one and book it. From there the booking details are stored and booked message is sent to the user.

## Chapter 3.4: Database Model

It is a logical structure which is used to determine how data in the system is stored, manipulated, processed and organized.

### Chapter 3.4.1: Data Dictionary

Data dictionary is simply a repository of metadata which is a set of files which stores all the collection of tables with metadata. It also contains relationship to other data.

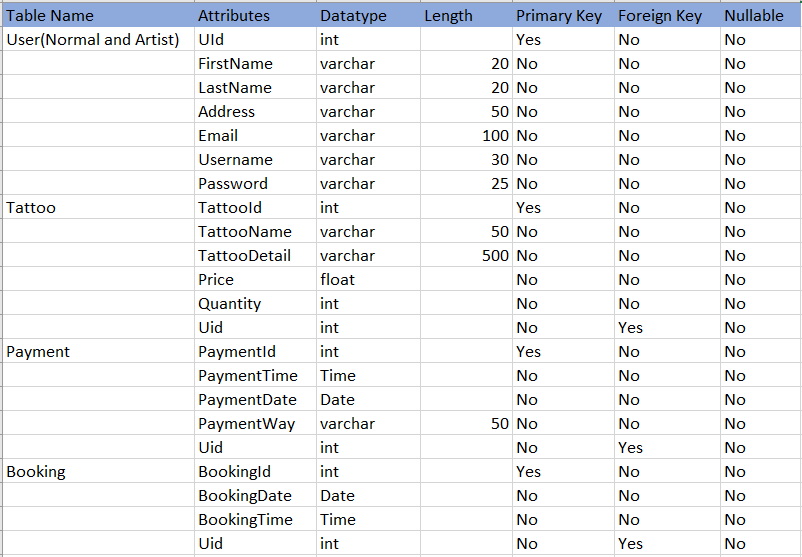


Figure Data Dictionary

### Chapter 3.4.2: ER Diagram

ER diagram is a diagram which shows the relationships between all the entities within the system. An ERD consists of entities with its attributes and relationship between the entities.

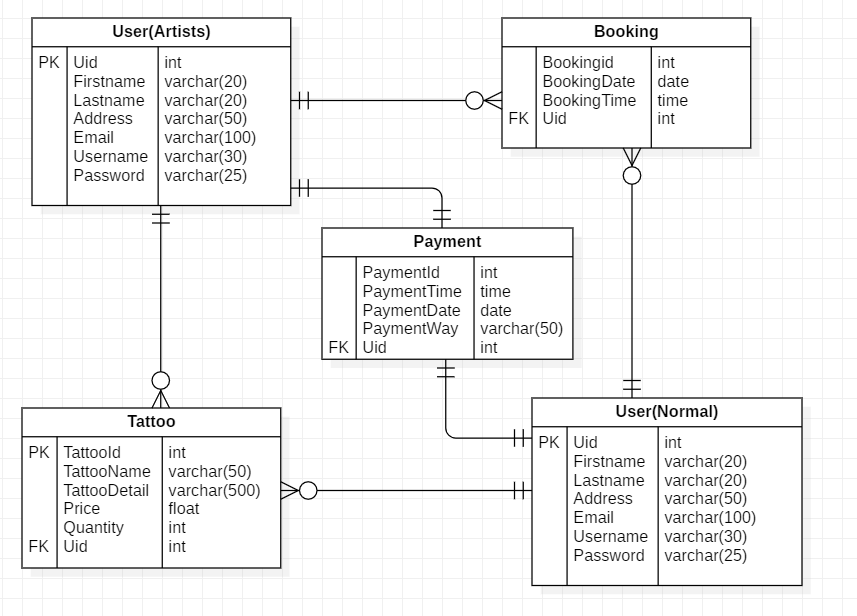


Figure ER Diagram

## Chapter 3.5: Architectural Model

Architectural model is mainly known as the blueprint of the system as it contains both the elements of structural model as well as behavioral model. For this system I have follow MVC pattern and it follows a 3-tier architecture such as:

1. Model – This connects to the database to perform certain tasks such as insert, delete, etc.
2. View – This is the user interface of the system where the clients interact with the system.
3. Controller – This acts as a layer which connects both the view and model of the system. Here validations, insertions and many more methods are performed.

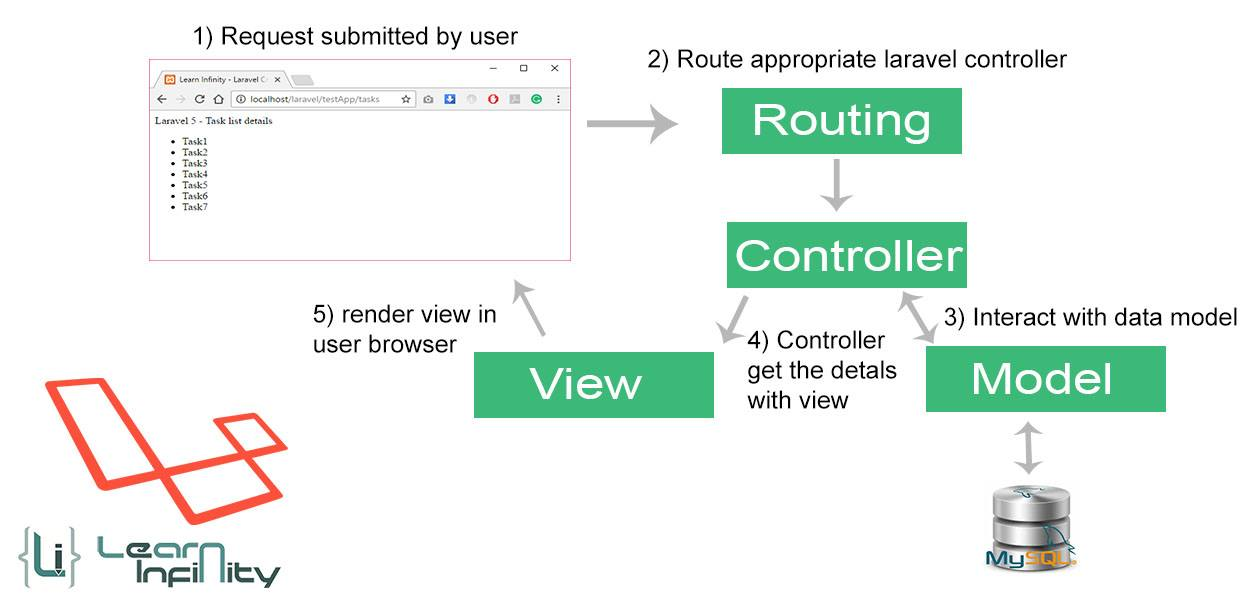


Figure Architectural Model

## Chapter 3.6: UI Modeling

User interface modeling is a development technique used by computer application developers.

## Chapter 3.6.1: Prototyping

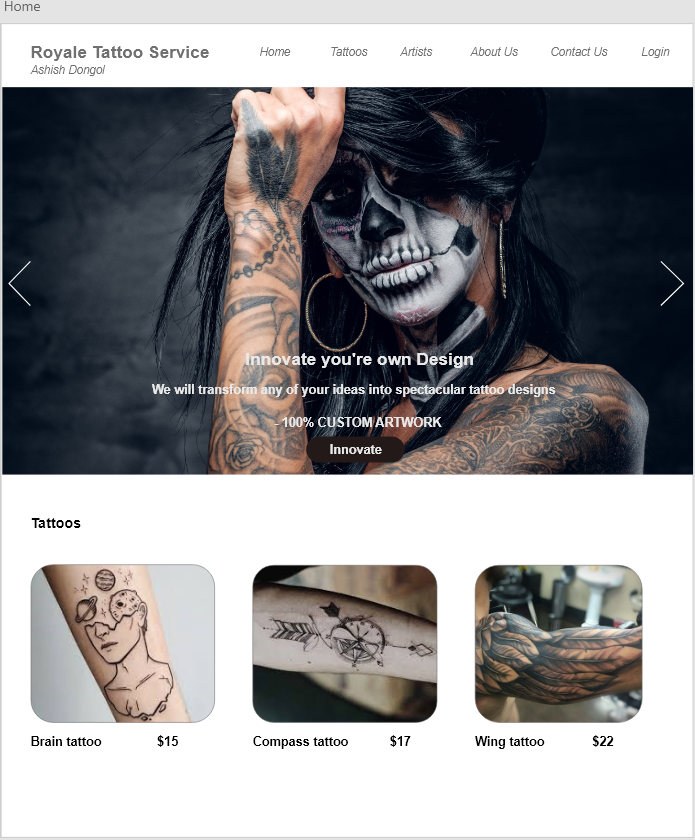
Prototype is a rough sketch which shows how the project will look in the future and It will be used as a basis to create that software. There are many types or categories of prototyping but the most commonly used is paper prototyping as it can be done very easily. Here is my prototype of the web app: -

Figure Home Page

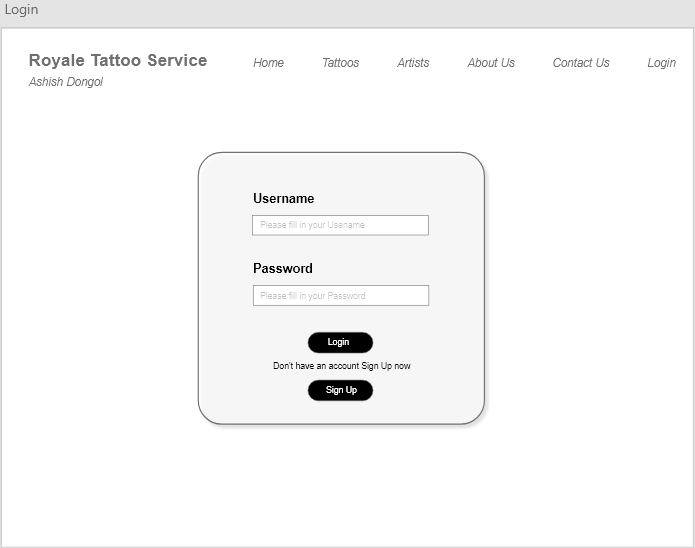


Figure Login Page

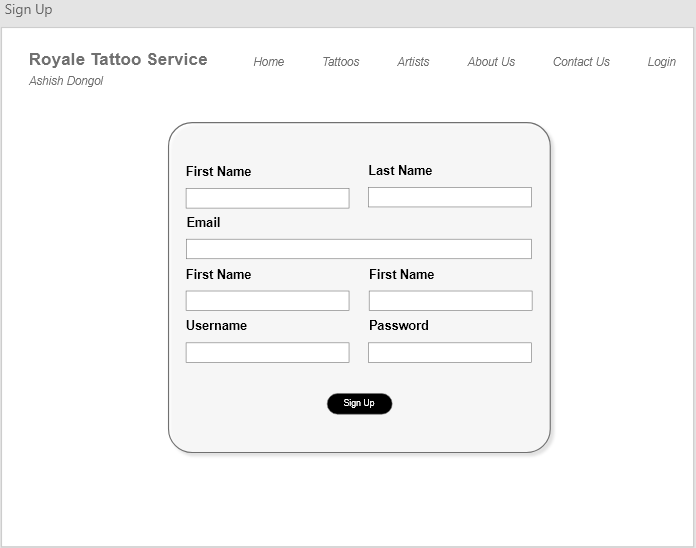


Figure Registration Page



Figure Contact Us Page

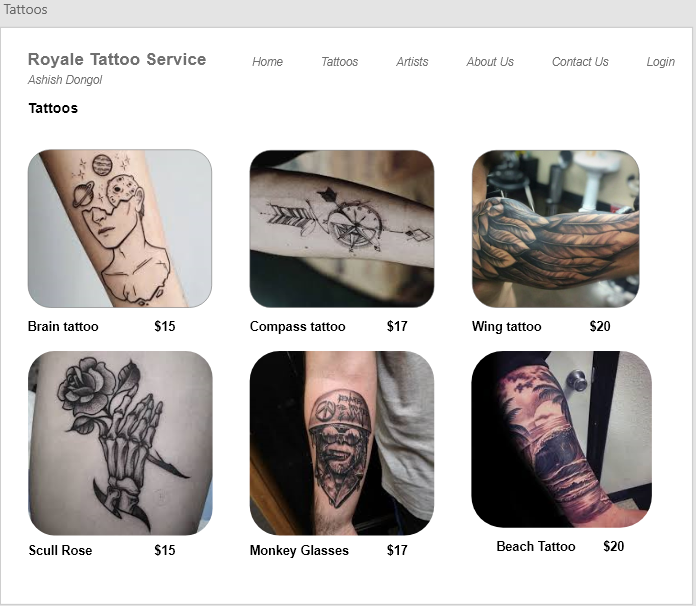


Figure Tattoos Page

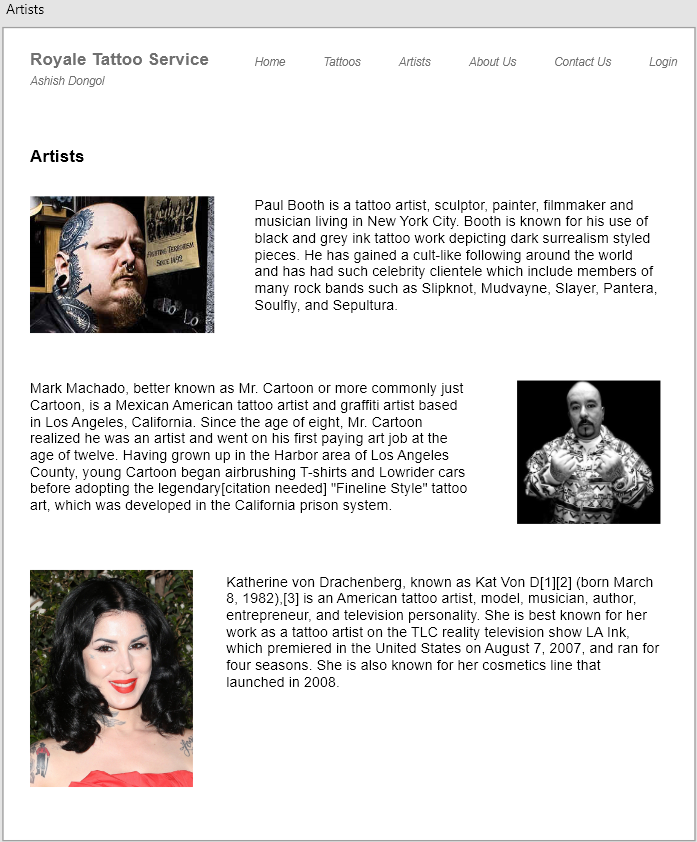


Figure Artists Page

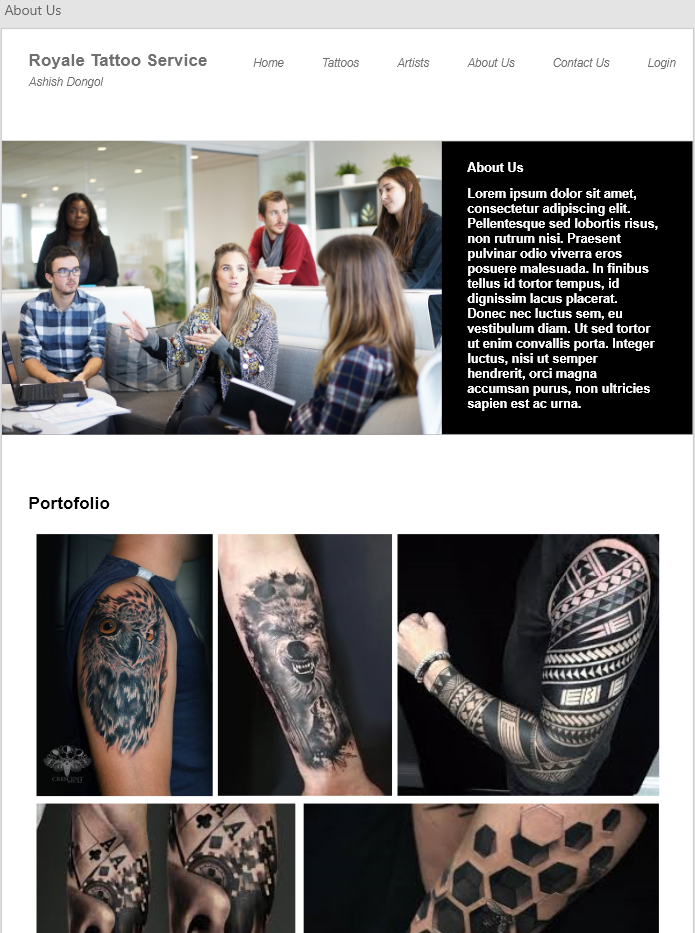


Figure About Us Page



Figure Admin Dashboard

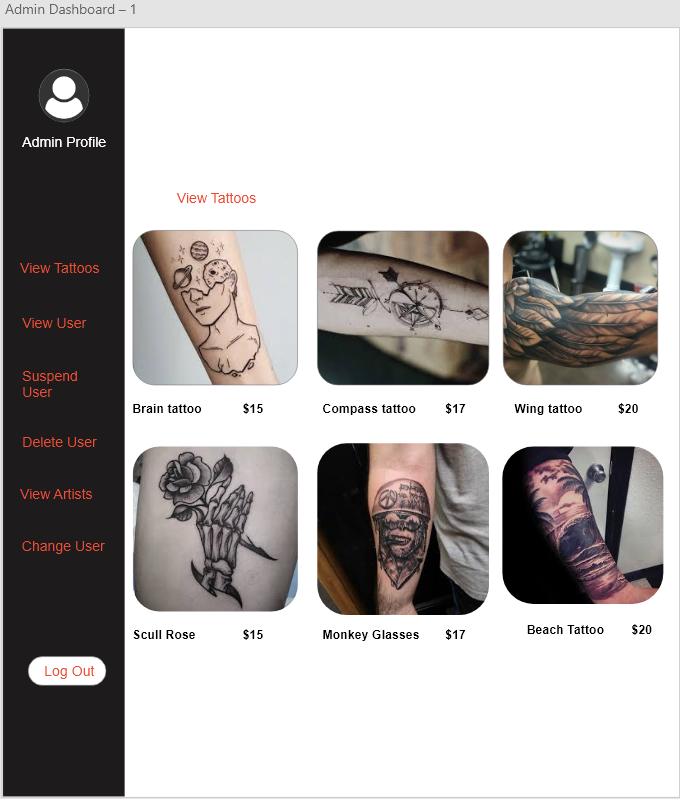


Figure Admin View Tattoos

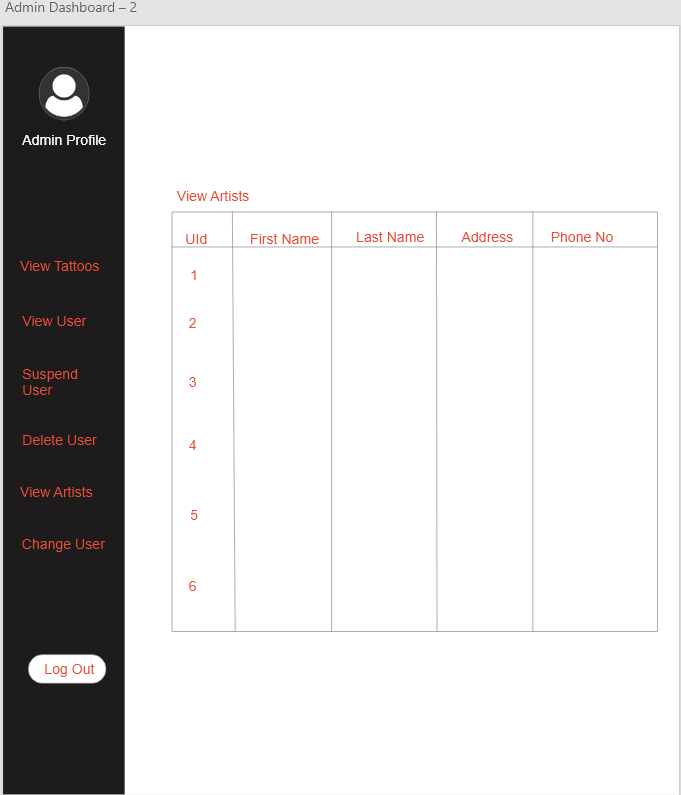


Figure Admin View Artists



Figure User Delete Form

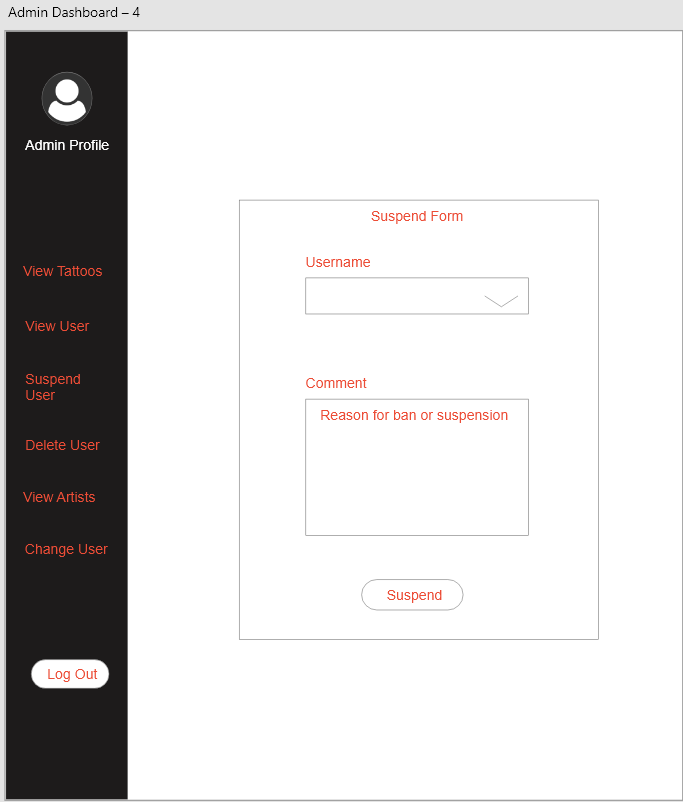


Figure User Suspend Form

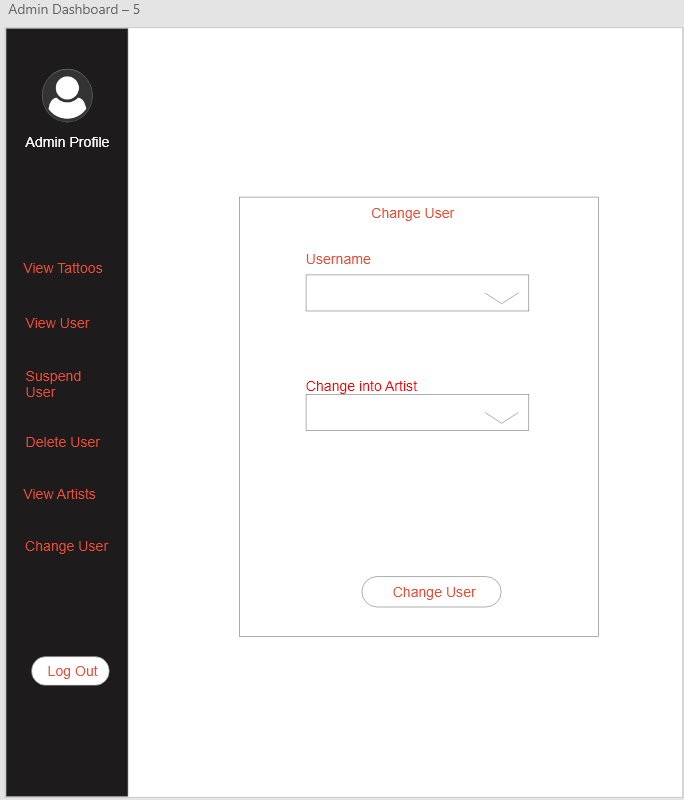


Figure Change User Form

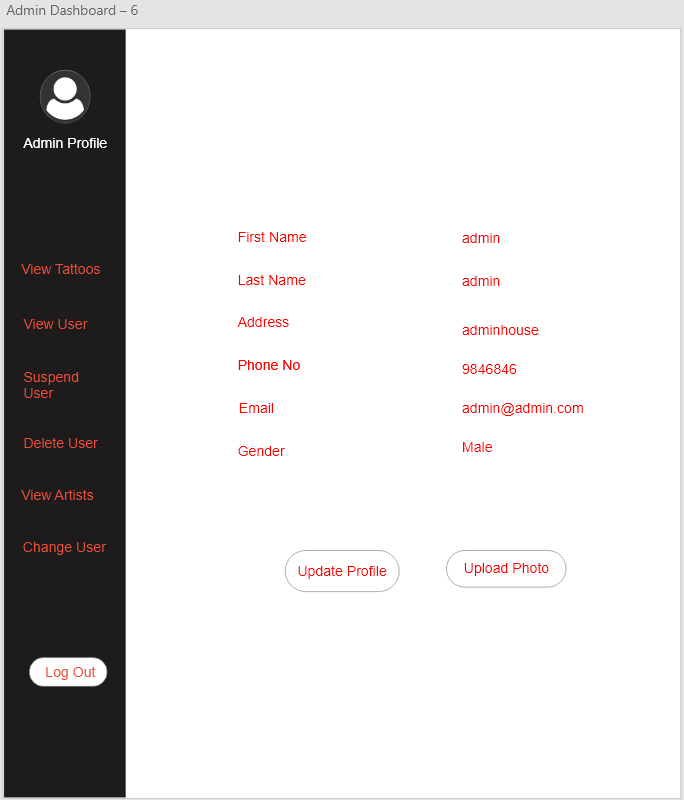


Figure Admin Profile

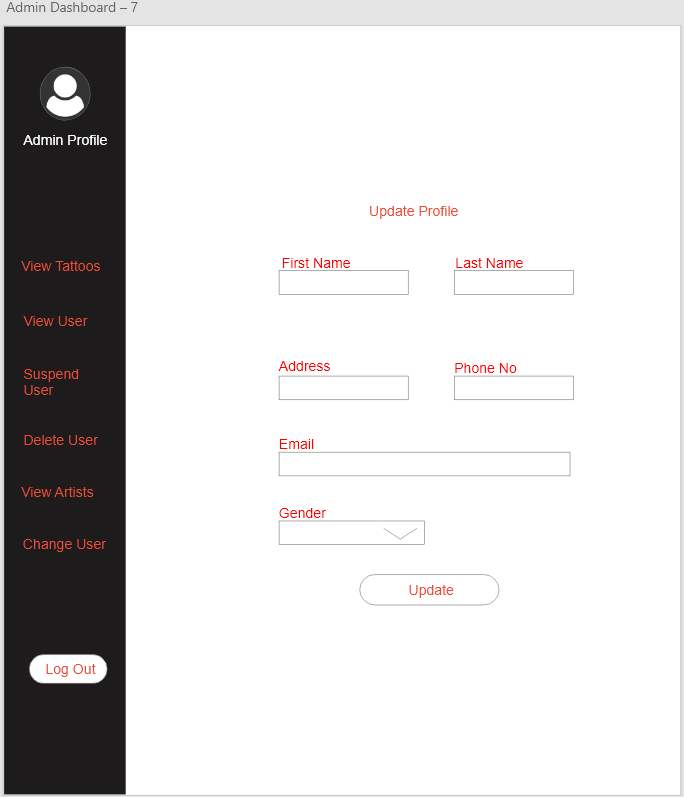


Figure Admin Update Profile

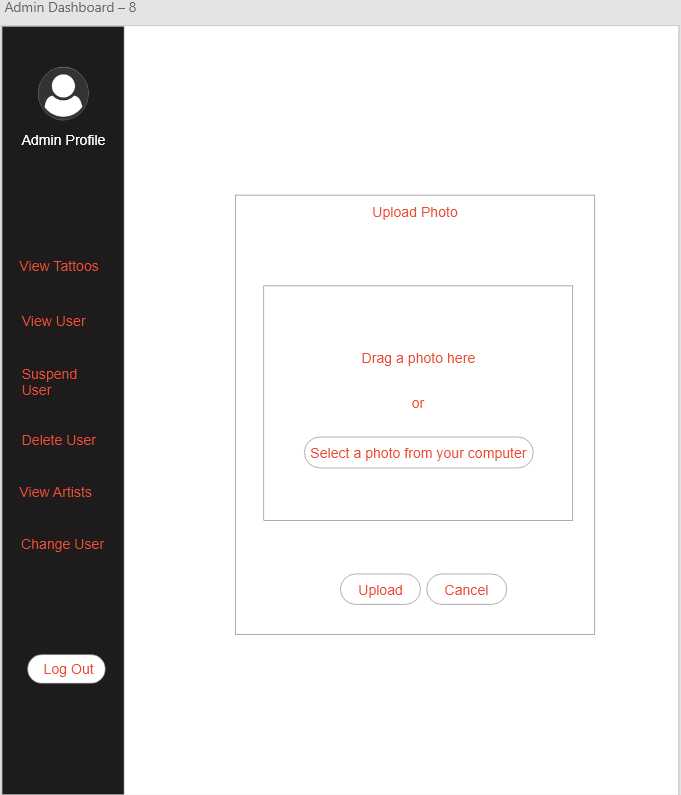


Figure Upload Photo Form

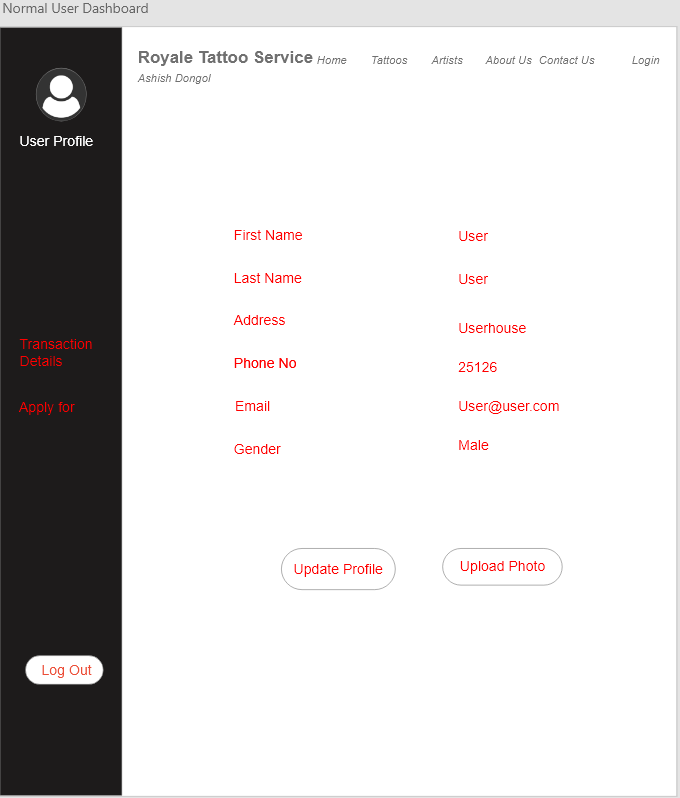


Figure User Dashboard

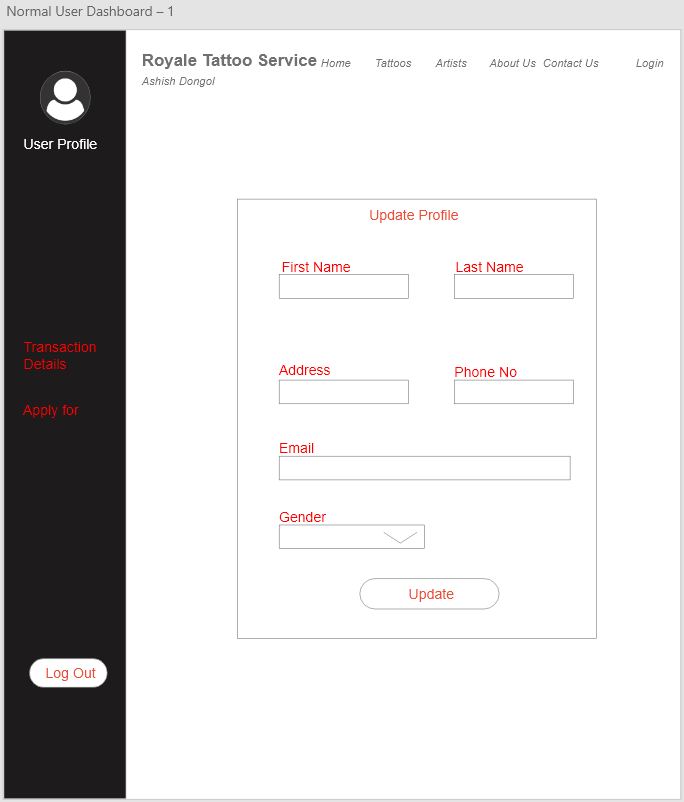


Figure User Update Profile Form

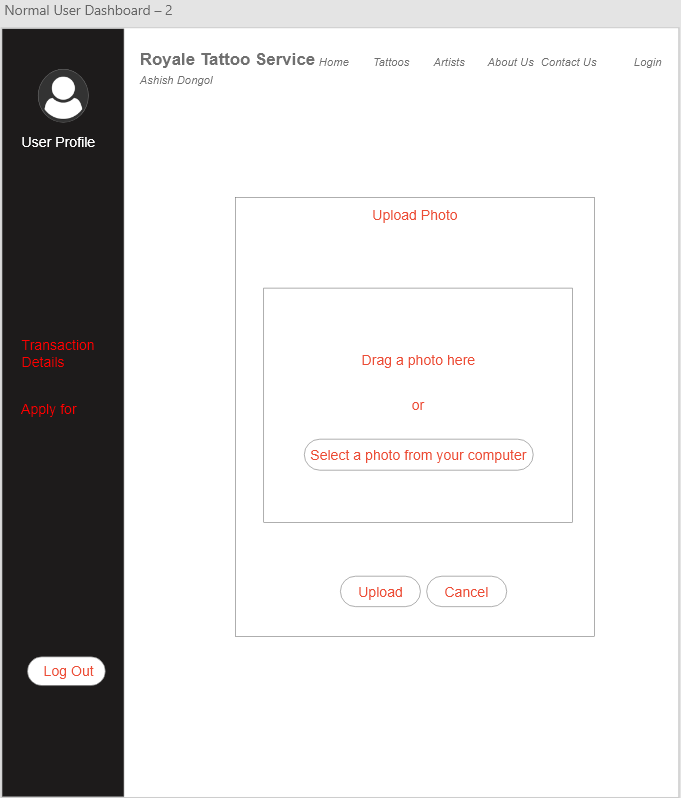


Figure User Upload Photo

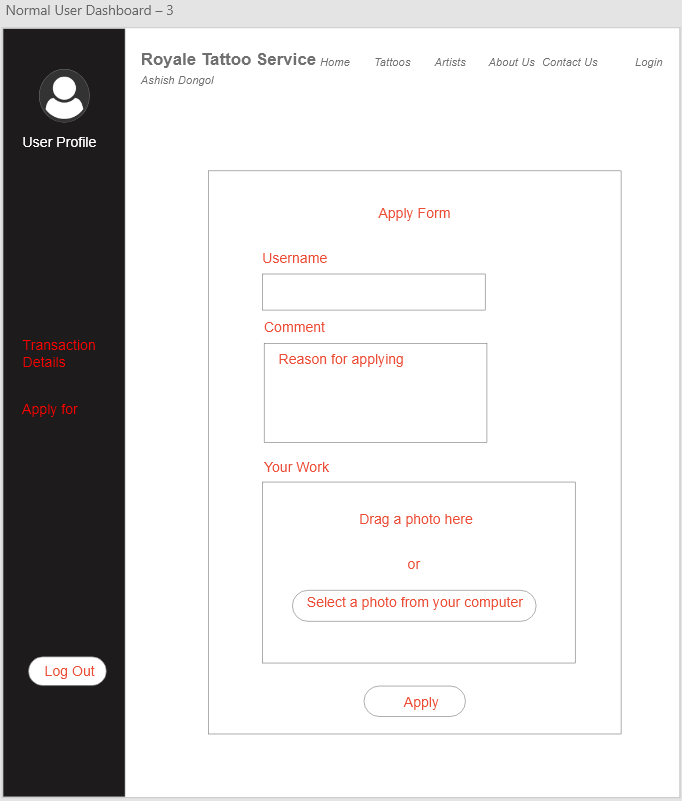


Figure Artist Apply Form

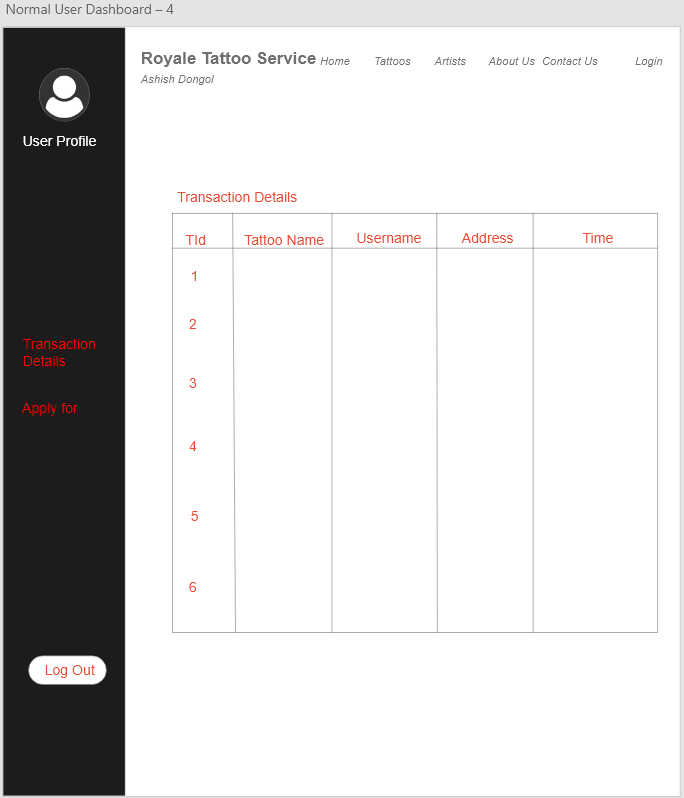


Figure User Transaction Details

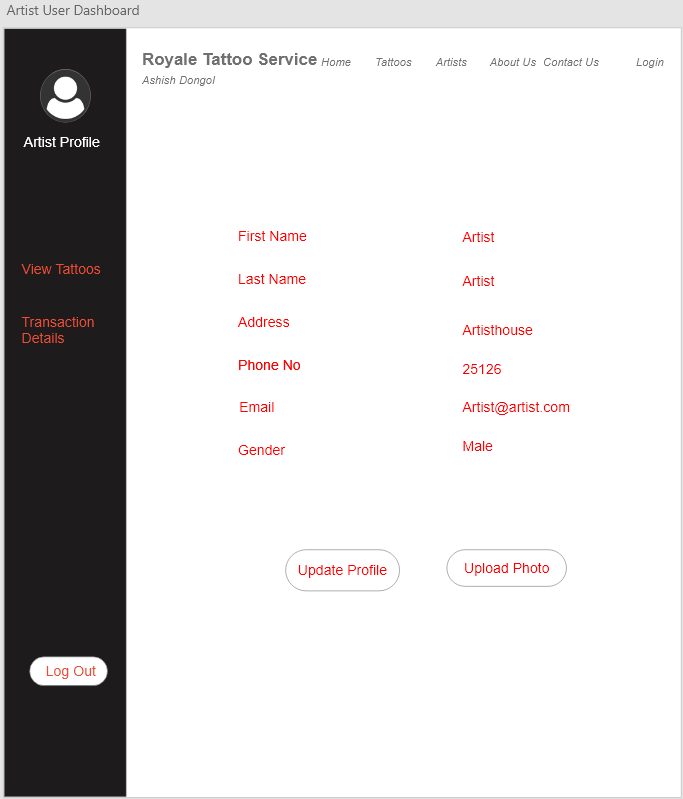


Figure Artist Dashboard

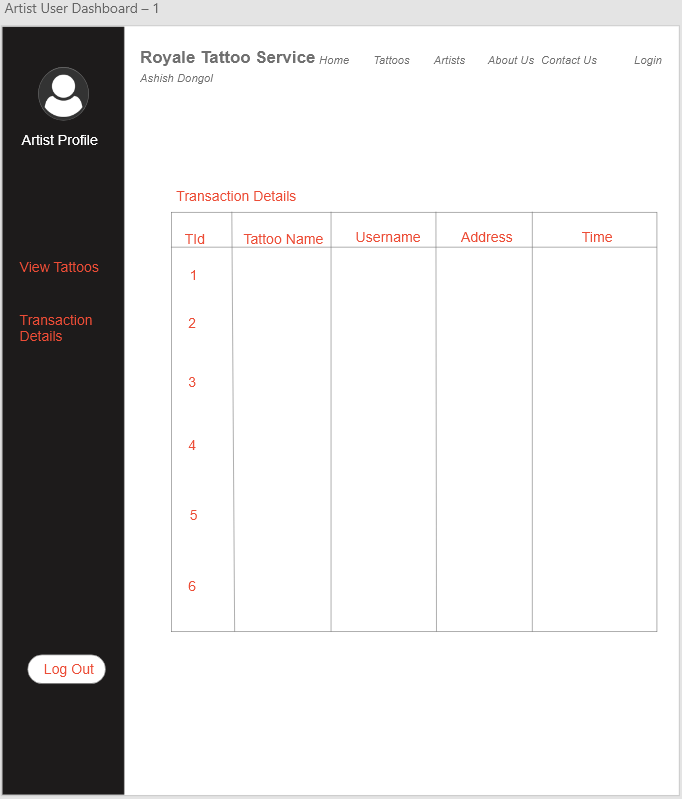


Figure Artist Transaction Details

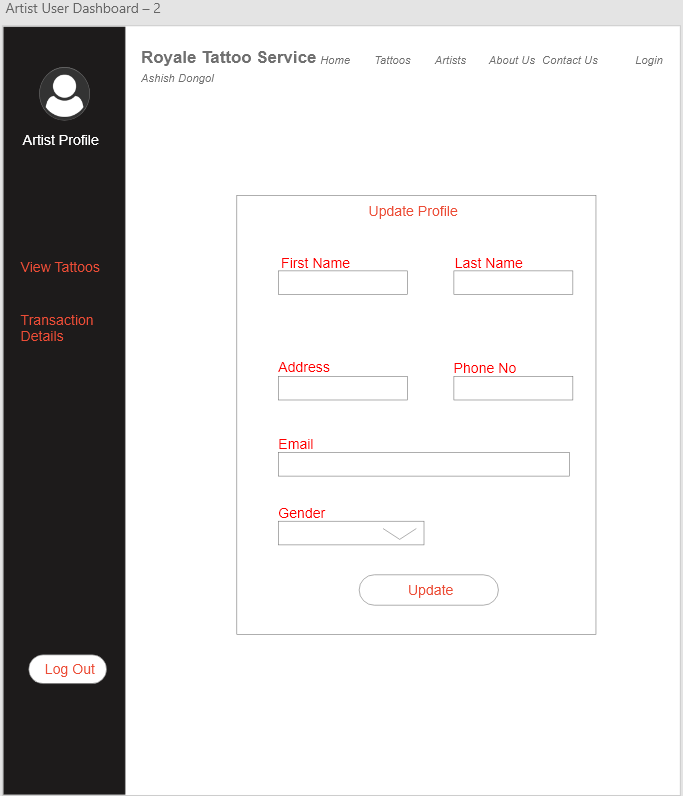


Figure Artist Update Profile

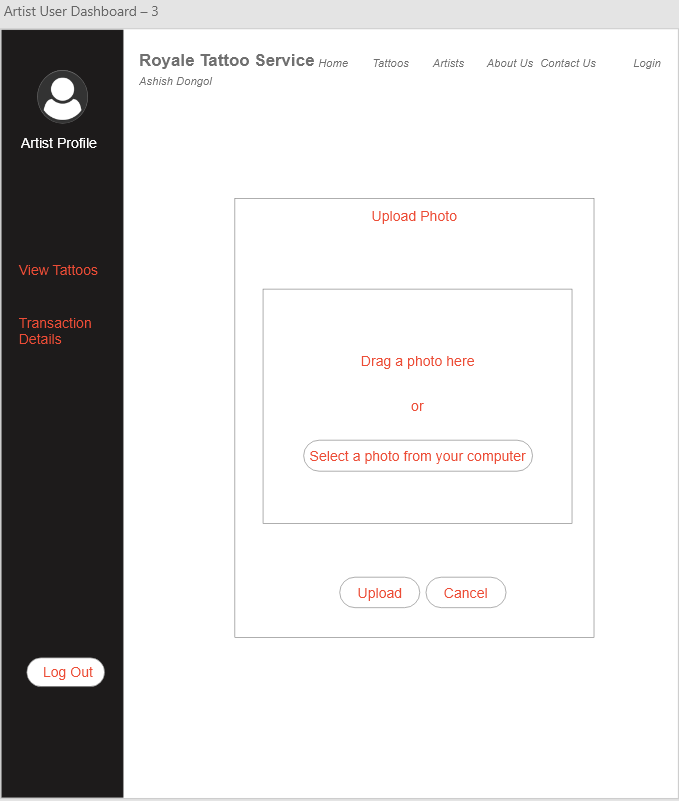


Figure Artist Upload Photo

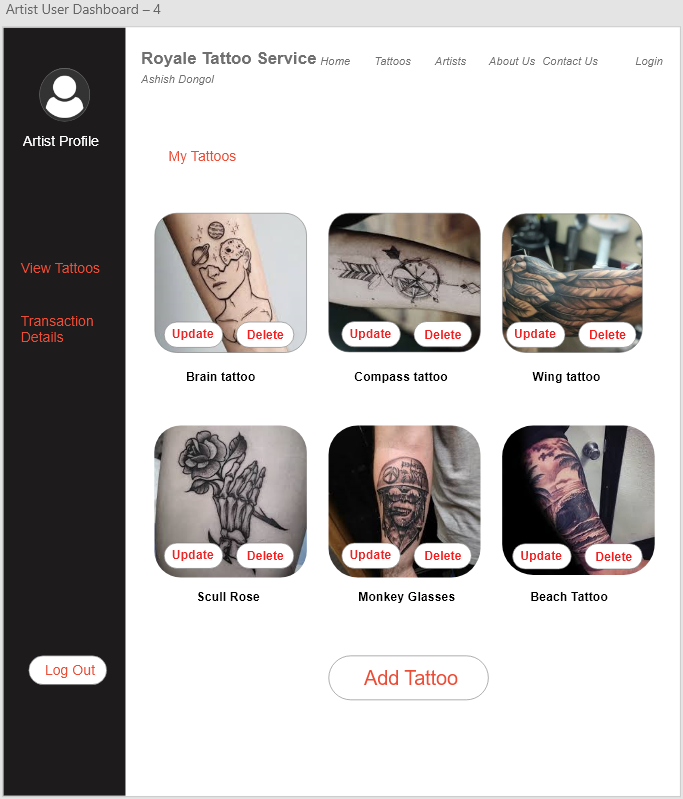


Figure Artist View Tattoos

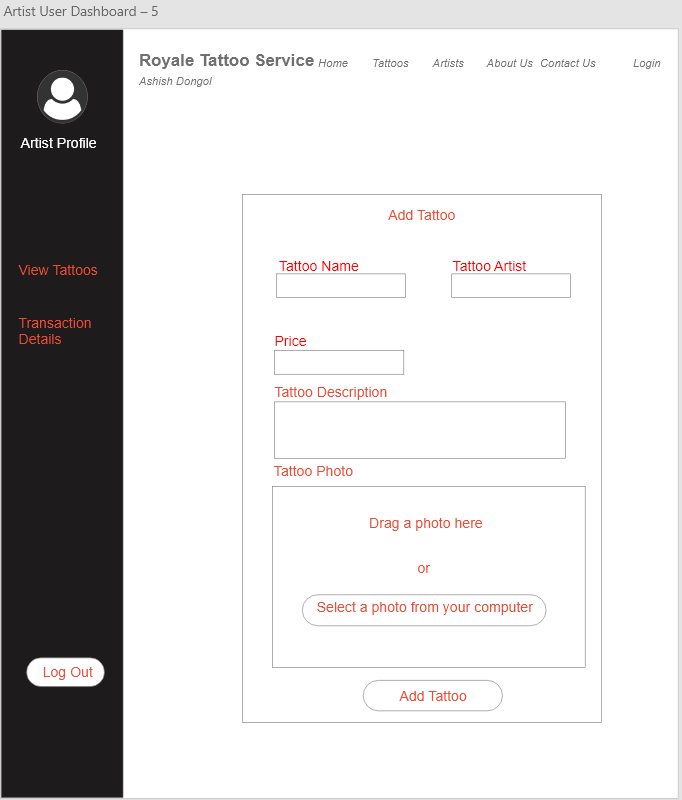


Figure Add Tattoo

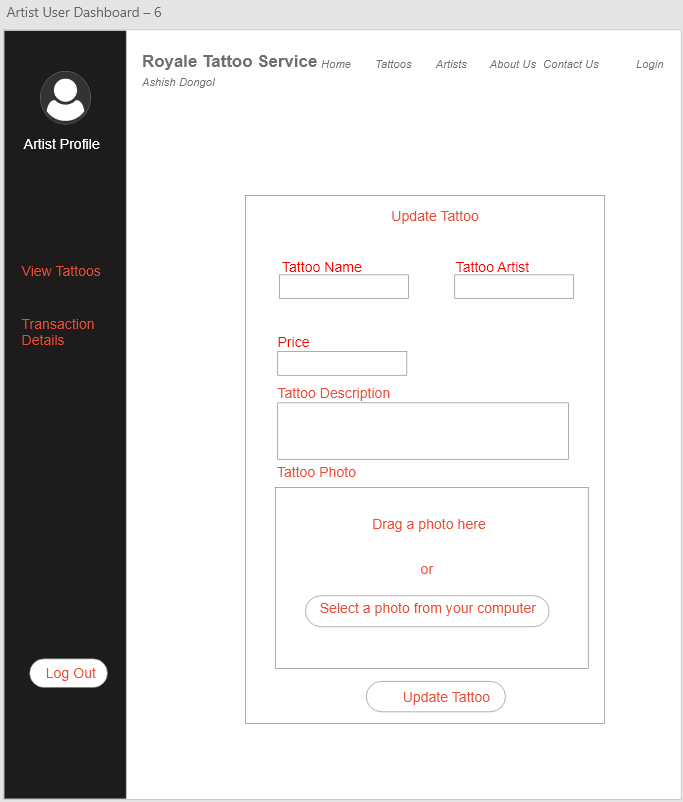


Figure Update Tattoo

# Chapter 4: Implementation

In this phase the system is developed with the chosen development model and for me that is the waterfall model. According to this model and design diagrams the coding is done. For implementation of the system I have used HTML, CSS, JavaScript, Bootstrap as well as jQuery. For these implementations I have used Laravel a php framework and have used the standards MVC pattern. Here the HTML are the front end of the system while the Laravel’s model is the back end of the system. To run this system a local server is made with XAMPP. For these implementations I have used PHP storm. Hence all the snap shots of the coding are given in the appendix section of the documentation.

# Chapter-5 Testing

Testing is a process, which is used to evaluate the functionality of an application with an intent to find whether the developed software meets the specified requirements or not and also used to identify the problems or defects within the system.

## Chapter 5.1: Unit Testing

Unittesting is a level of testing where individual units/ components of an application are tested. The purpose is to validate that each unit of the application performs as designed.

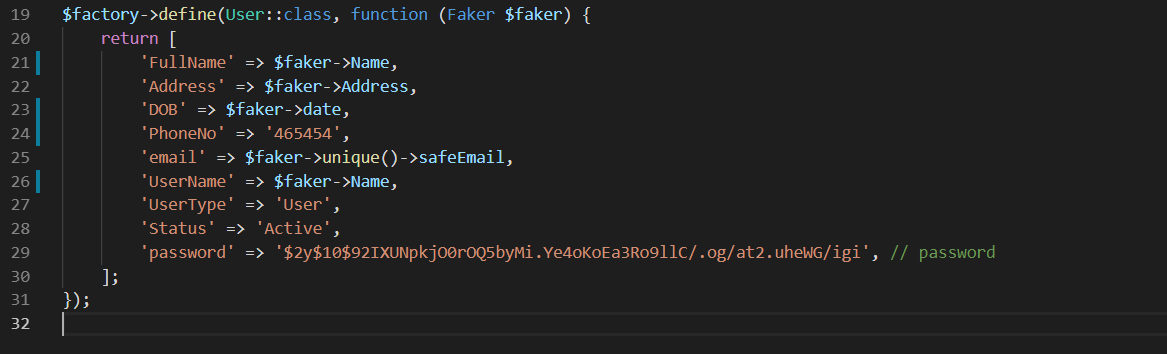


Figure Fake Data for register



Figure Testing of data

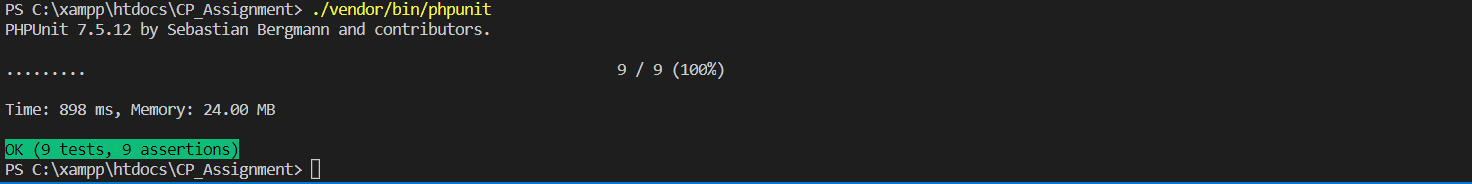


Figure Test Passed

|  |  |
| --- | --- |
| Test Id | 1 |
| Test Class | User |
| Expected Outcome | User Register Successful |
| Actual Outcome | User Registration was successful |

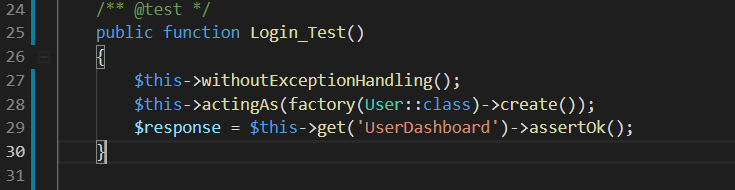


Figure Login Test

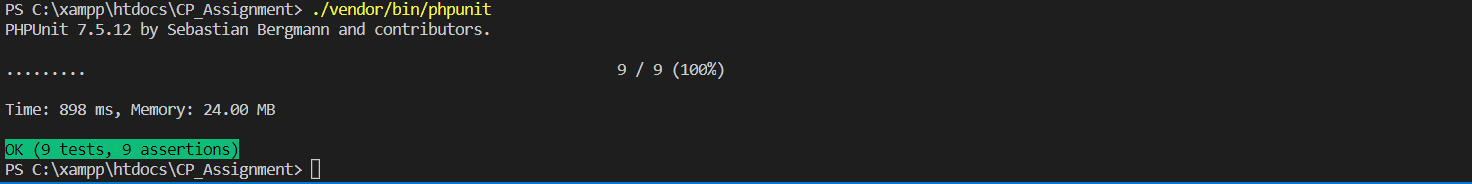


Figure Test Pass

|  |  |
| --- | --- |
| Test Id | 2 |
| Test Class | User |
| Expected Outcome | Login Successful |
| Actual Outcome | Login was successful |

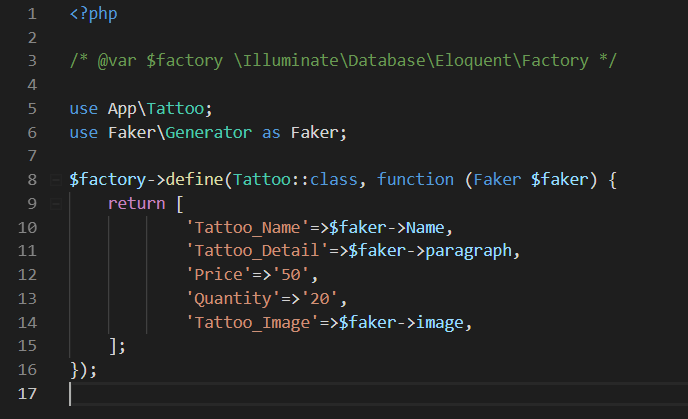


Figure Fake Tattoo Data

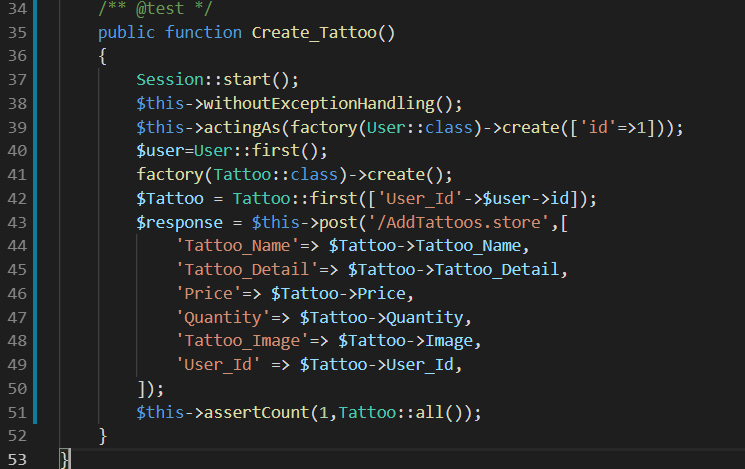


Figure Create Tattoo Test

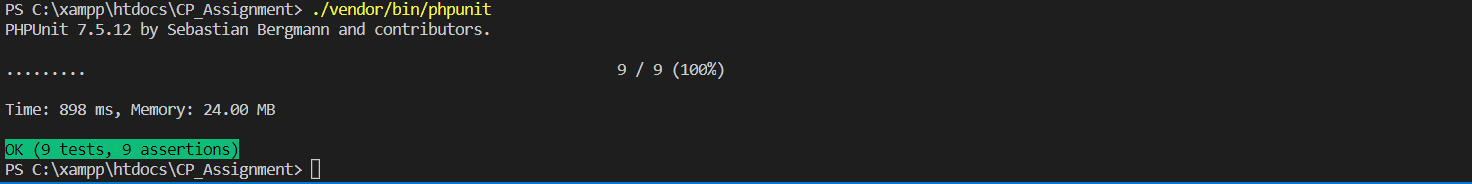


Figure Test Pass

|  |  |
| --- | --- |
| Test Id | 3 |
| Test Class | Tattoo |
| Expected Outcome | Tattoo successfully created |
| Actual Outcome | Tattoo was successfully created |

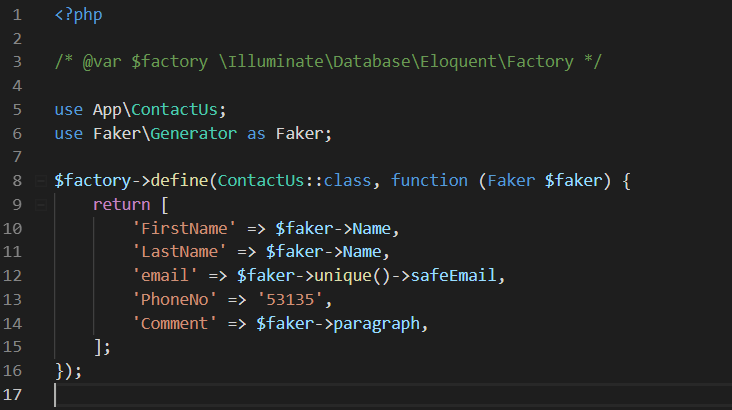


Figure Fake Contact Us Data



Figure Create Contact US Test

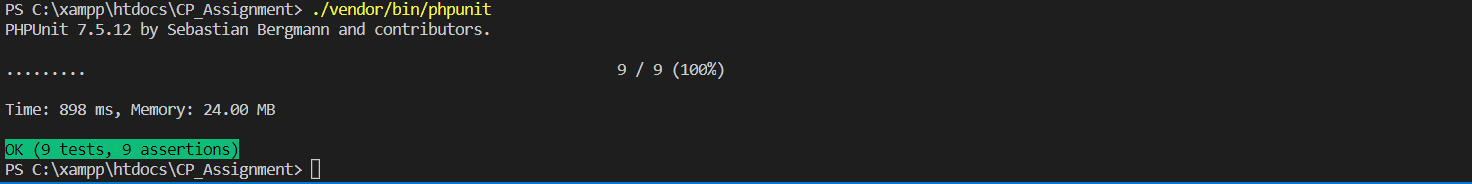


Figure Test Pass

|  |  |
| --- | --- |
| Test Id | 4 |
| Test Class | Contact Us |
| Expected Outcome | Contact Us successfully created |
| Actual Outcome | Contact Us was successfully created |

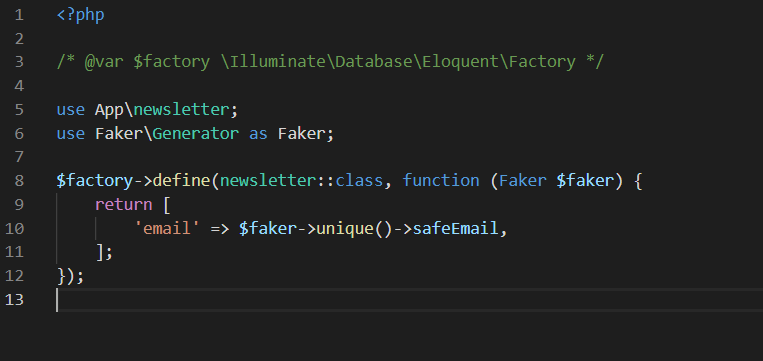


Figure Fake Newsletter Data

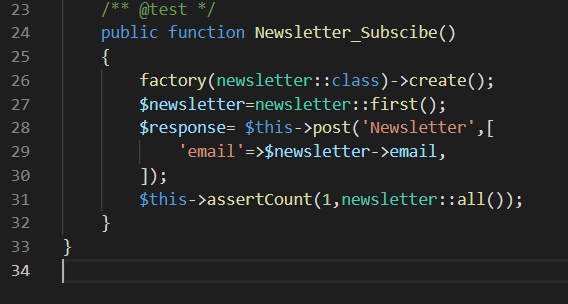


Figure Newsletter Test

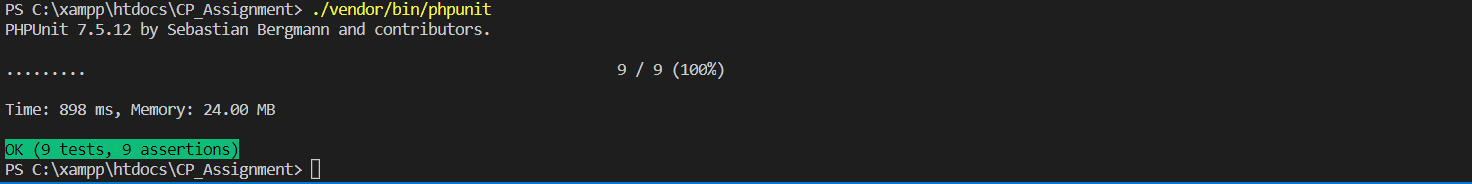


Figure Test Pass

|  |  |
| --- | --- |
| Test Id | 5 |
| Test Class | newsletter |
| Expected Outcome | Newsletter successfully subscribed |
| Actual Outcome | Newsletter was successfully subscribed |

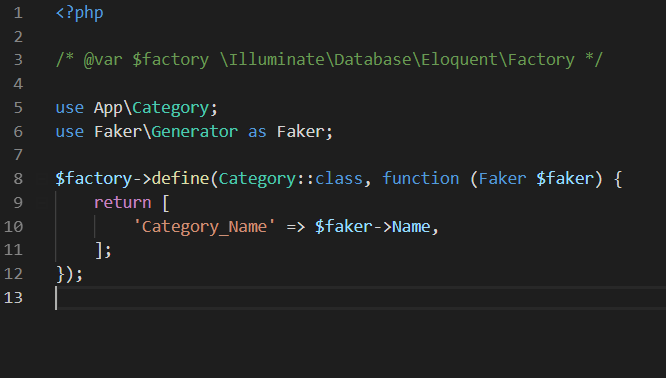


Figure Fake Category Data

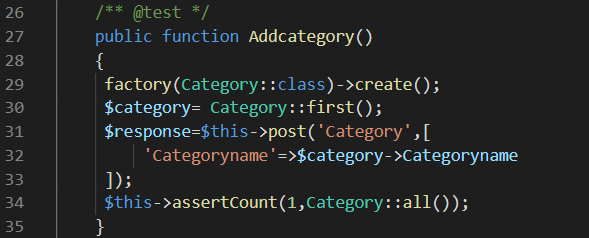
`

Figure Add Category Test

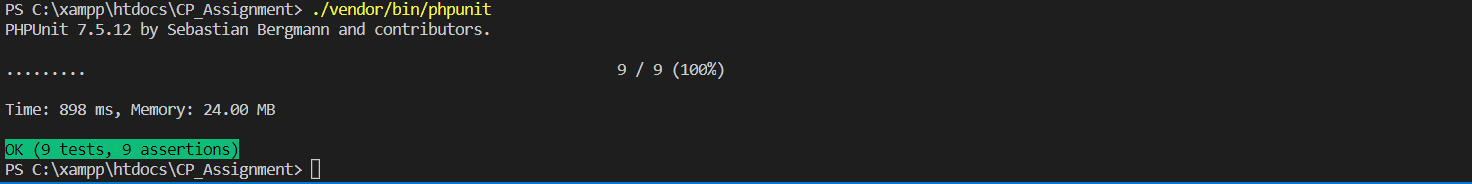


Figure Test Pass

|  |  |
| --- | --- |
| Test Id | 6 |
| Test Class | Category |
| Expected Outcome | Category successfully created |
| Actual Outcome | Category was successfully created |



Figure Buying Tattoo Test

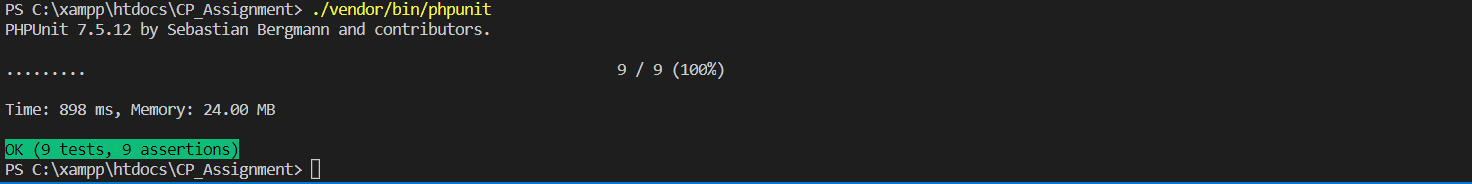


Figure Test Pass

|  |  |
| --- | --- |
| Test Id | 7 |
| Test Class | buys |
| Test Data | Payment Method = “Esewa”, Price =”2000”, Quantity =”1”, Total =”2000”, Location =” asdasd”, Contact =” asdasd” |
| Expected Outcome | Tattoo successfully bought |
| Actual Outcome | Artist was successfully bought |

## Chapter 5.2: Black Box Testing

Black-box testing is a method of application testing that examines the functionality of an application which does not concern about the internal codes or structures it mainly focuses on the user interface of an application. It concerns on whether the functionality works or not.

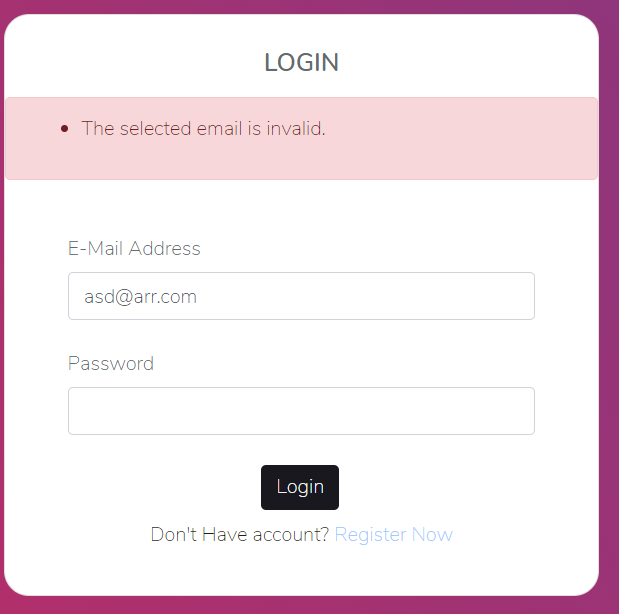


Figure Login Fail with wrong email

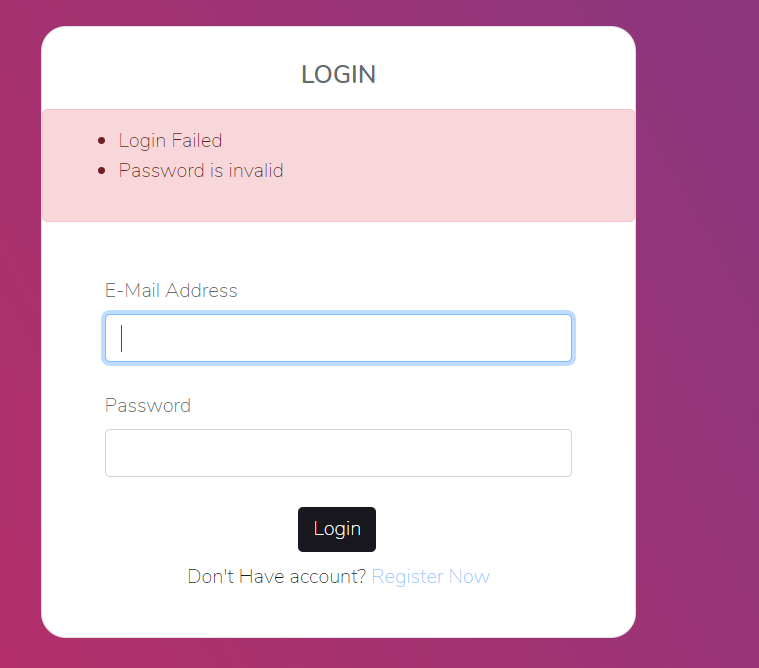


Figure Login Failed with wrong password

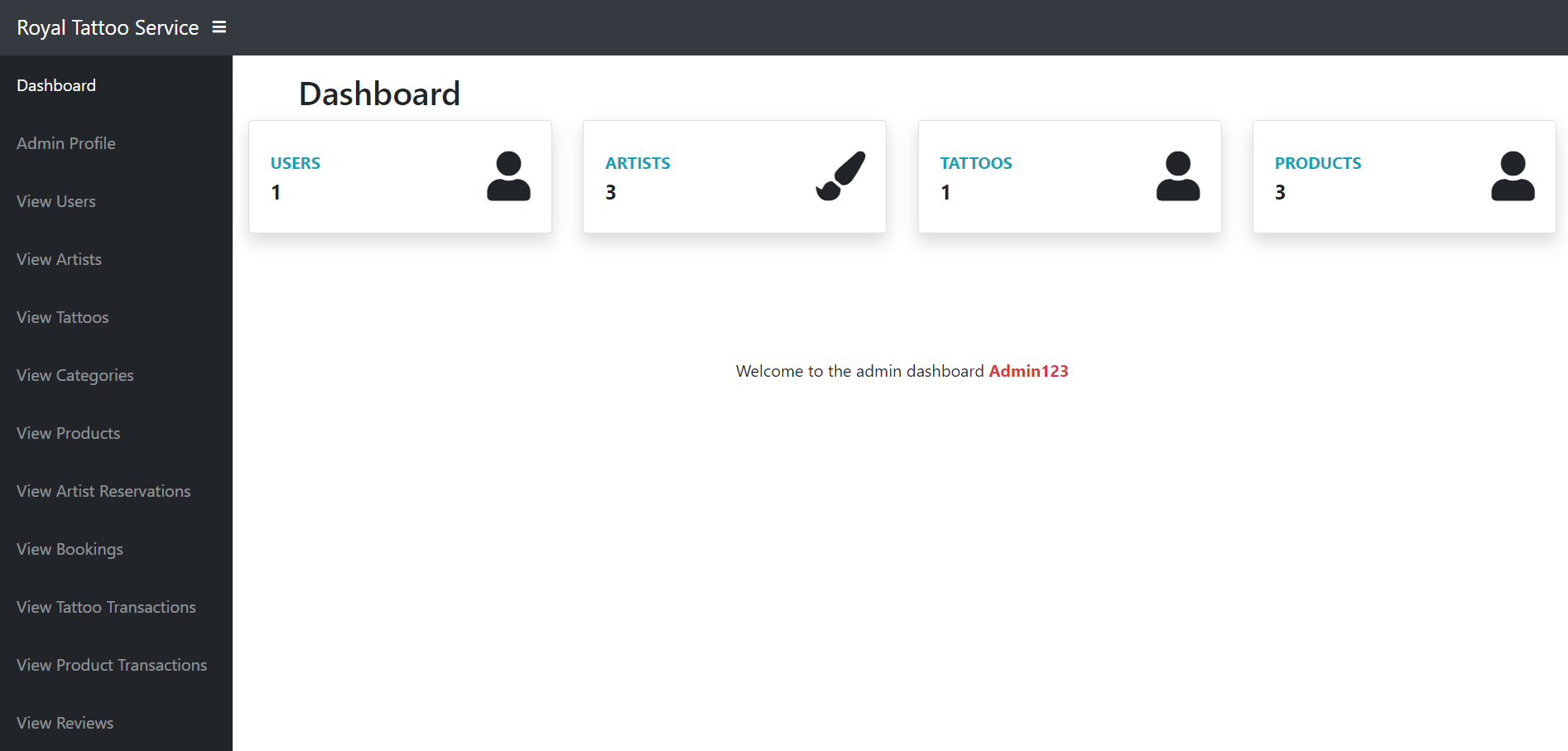


Figure Dashboard After Login

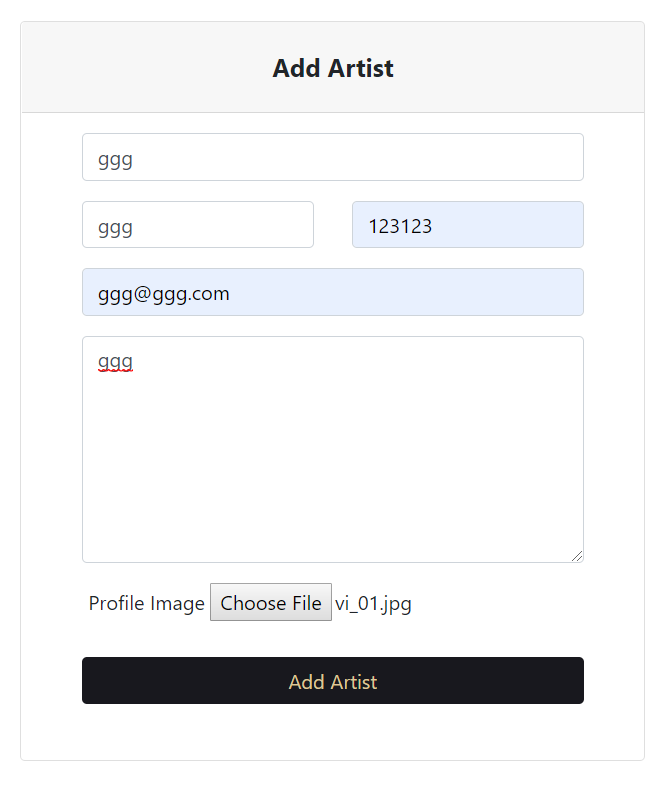


Figure Add Artist Form

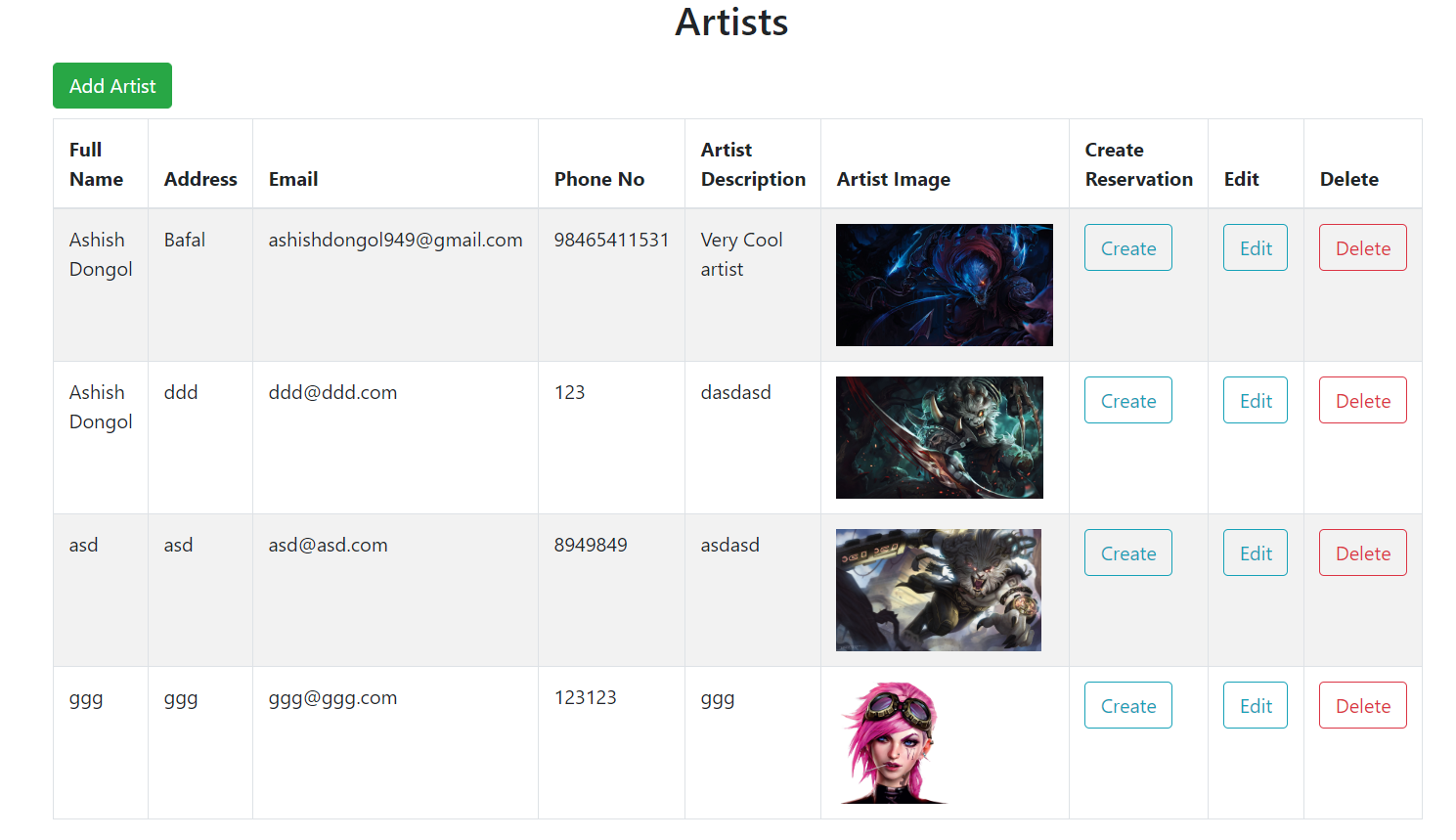


Figure After Addition

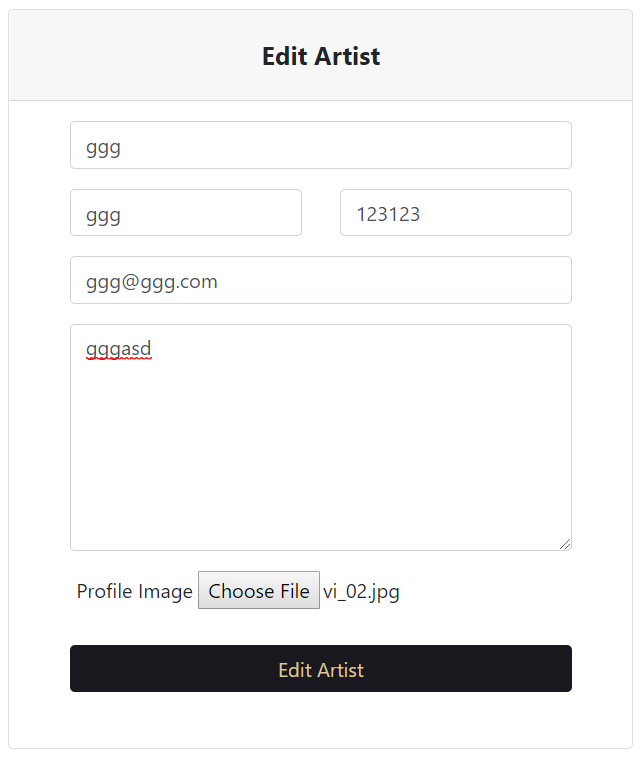


Figure Edit Artist Form

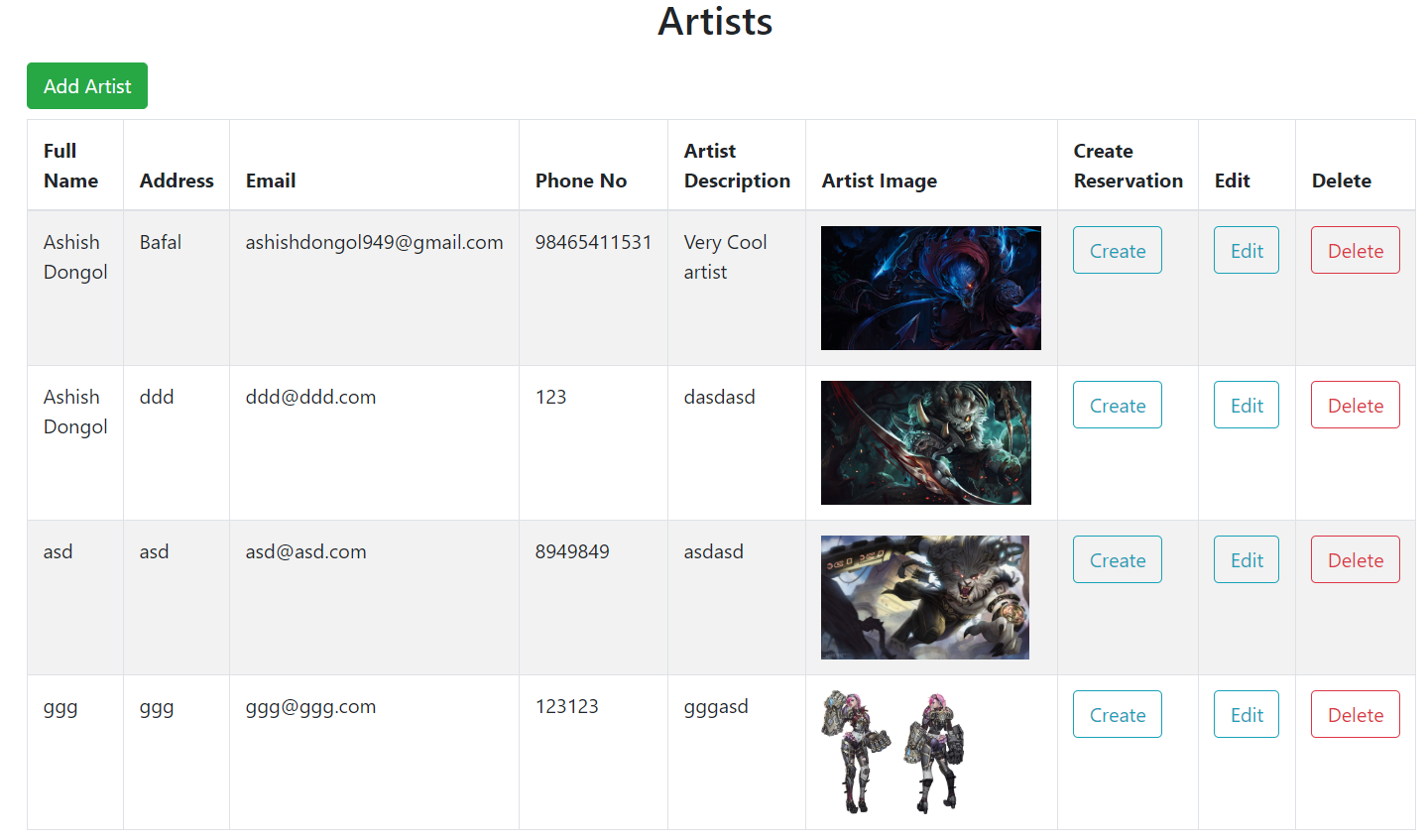


Figure After Edit

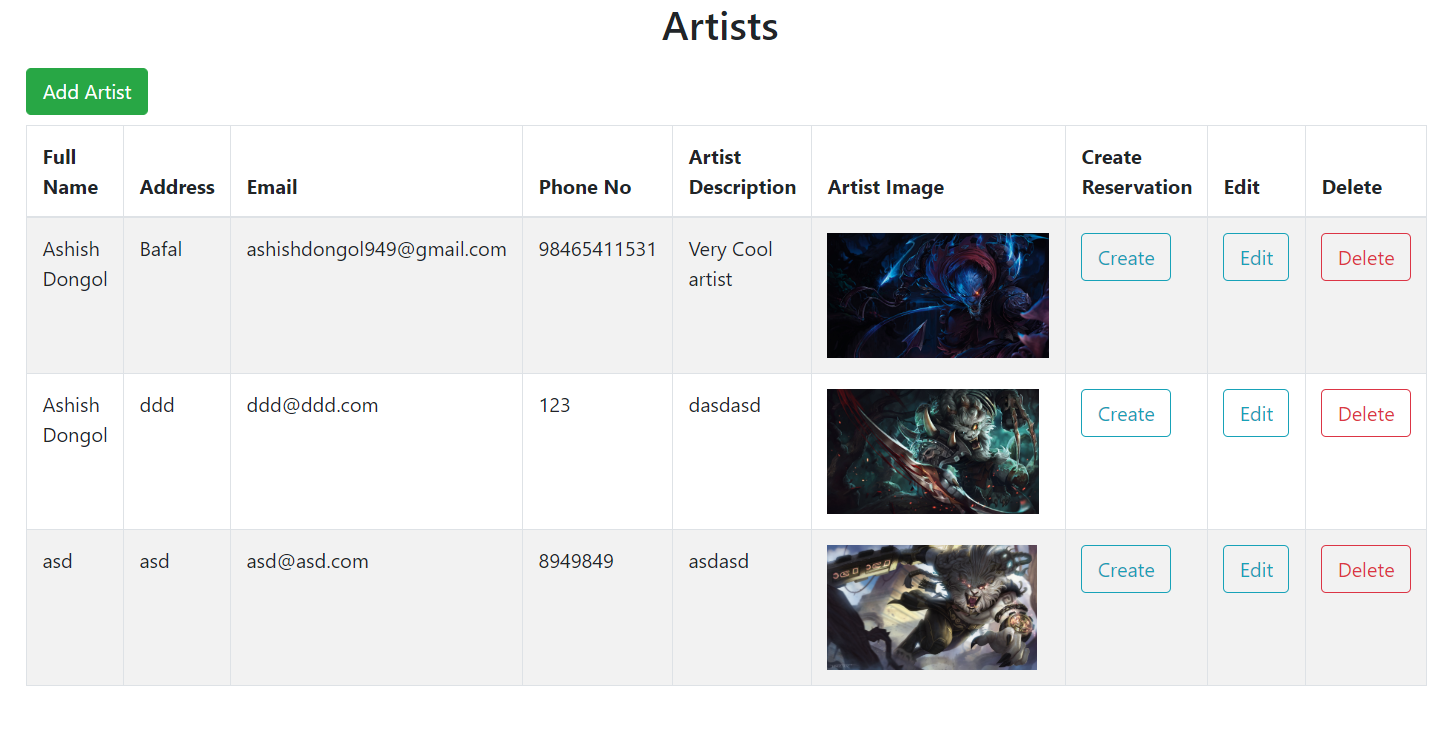


Figure After Deletion

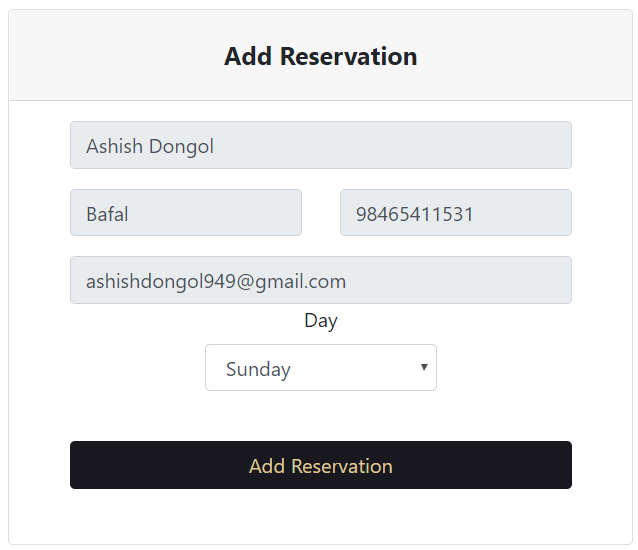


Figure Reservation Addition Form



Figure After Addition

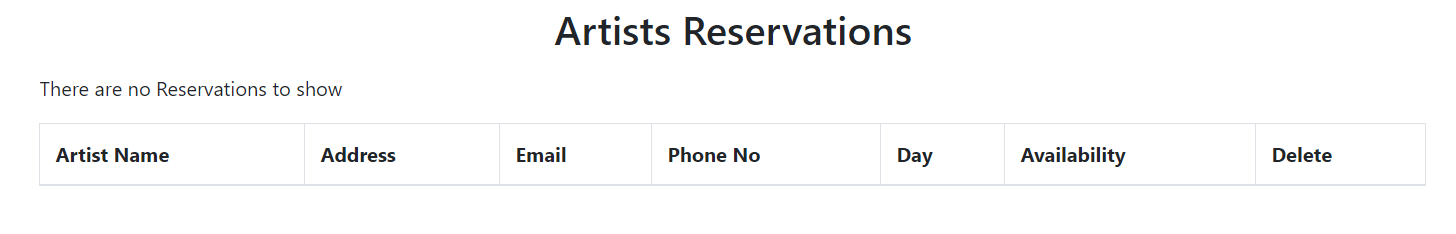


Figure After Deletion

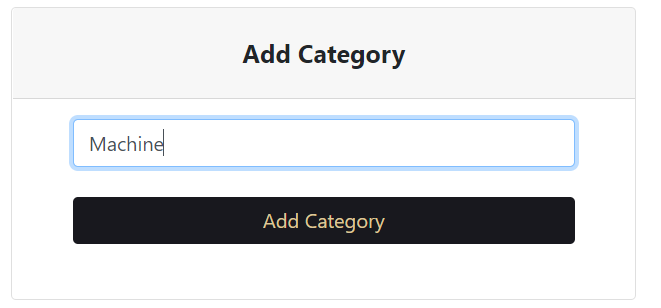


Figure Add Category Form

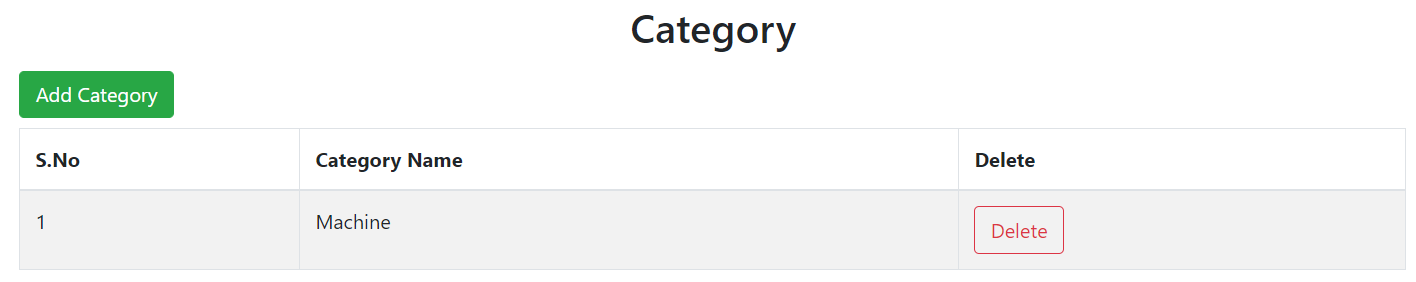


Figure After Addition

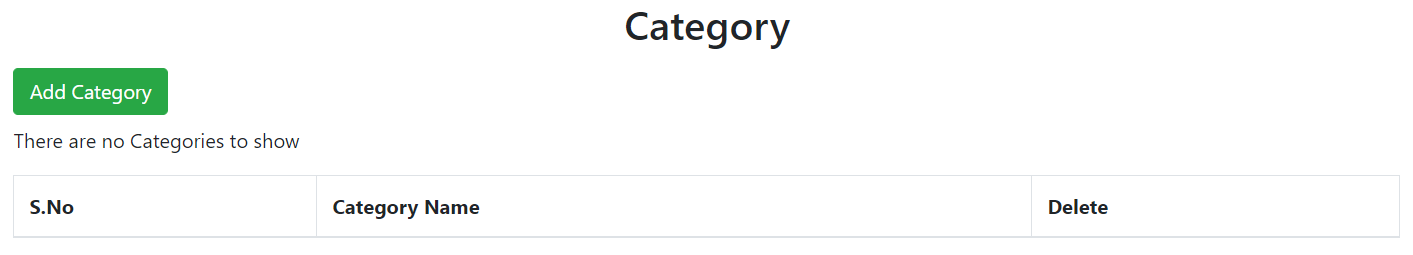


Figure After Deletion

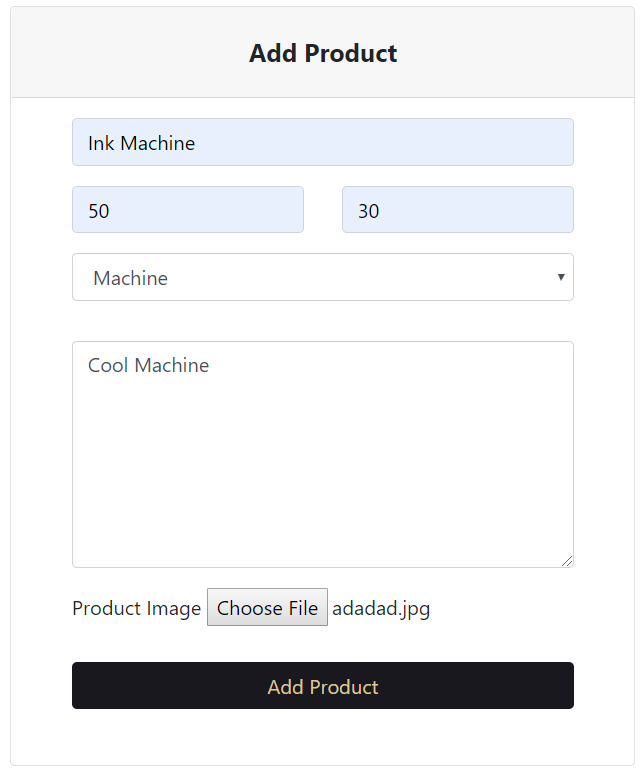


Figure Add Product Form

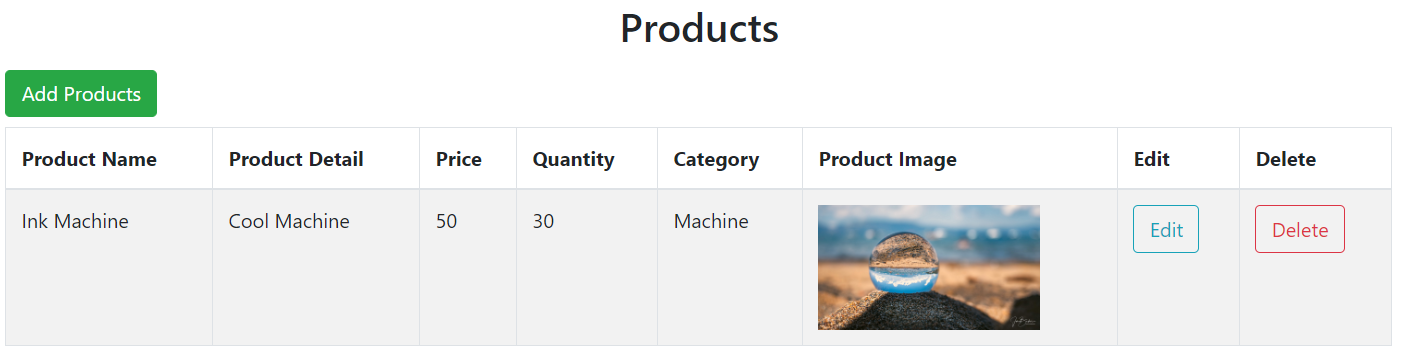


Figure After Addition

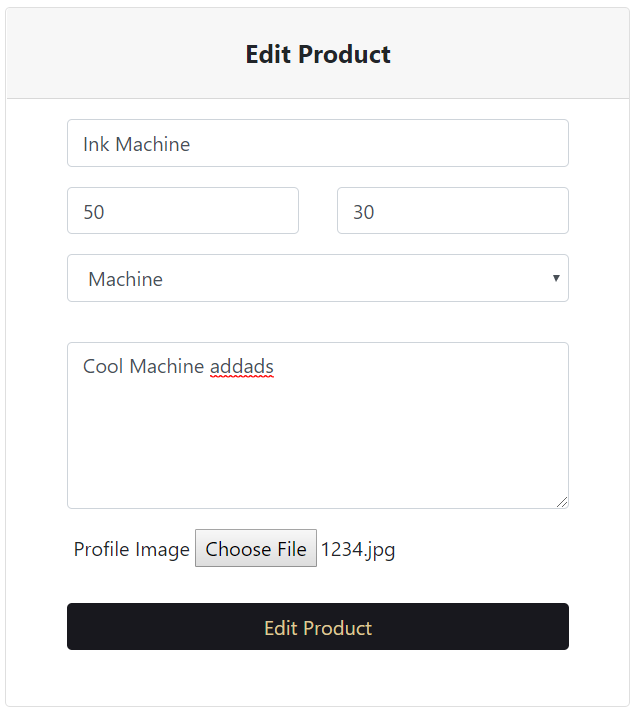


Figure Product Edit Form

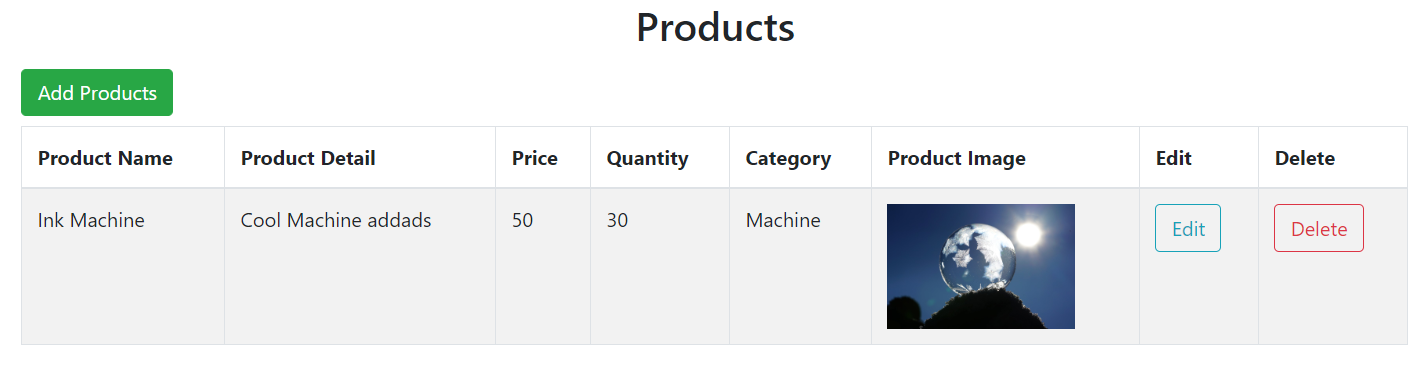


Figure After Edit of Product

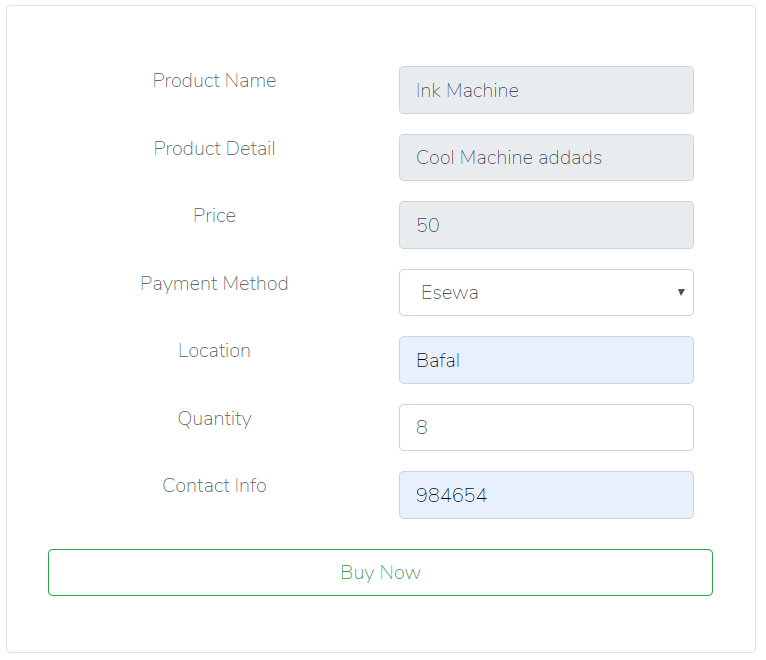


Figure Product Buy Form

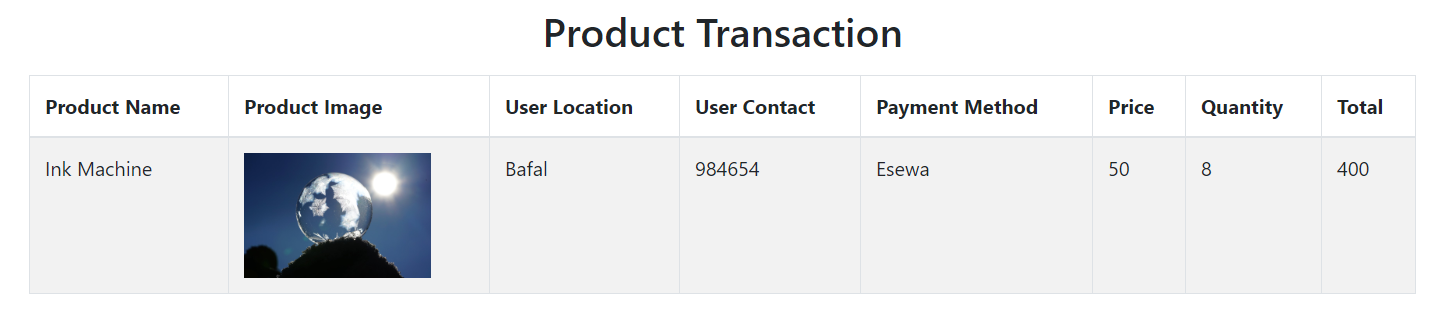


Figure After Buying

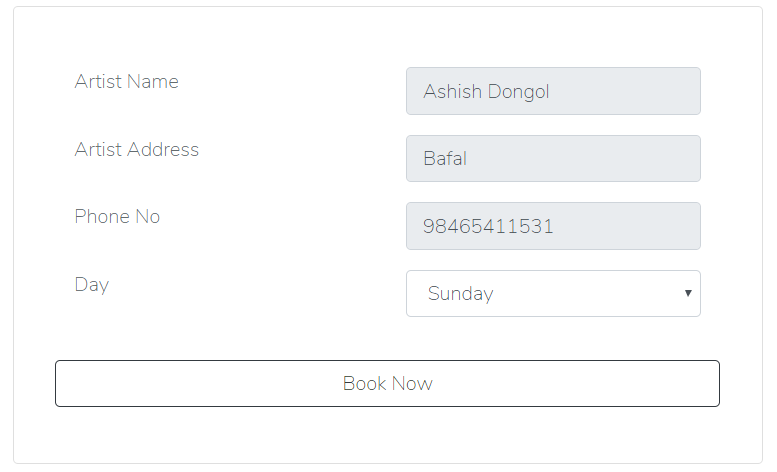


Figure Artist Booking Form

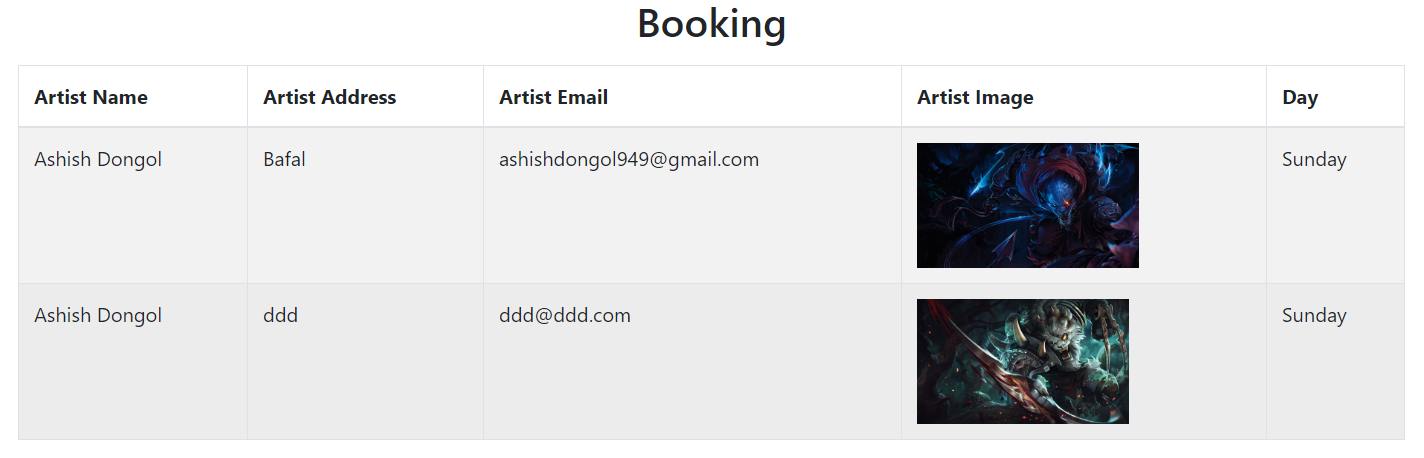


Figure After Booking

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Description** | **Expected Result** | **Test Data** | **Actual Result** | **Status** | **Tester** | **Date/Time** | **Comments** |
| Admin Login | Login Successful | Email=admin@admin.com, Password=dasd | Login Fail | Fail | Ashish Dongol | 2019/06/30 | Wrong Password |
| Admin Login | Login Successful | Email=admin@admin.com, password=admin123 | Login Success | Pass | Ashish Dongol | 2019/06/30 |  |
| Artist Addition | Artist Successfully Added | Name=gg, address=ggg, phone=123123, [email=ggg@ggg.com](mailto:email=ggg@ggg.com), comment=ggg | Addition Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Artist Update | Artist Successfully Updated | Name=gg, address=ggg, phone=123123, [email=ggg@ggg.com](mailto:email=ggg@ggg.com), comment=ggasd | Update Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Artist Deletion | Artist Successfully Deleted | Id=4 | Deletion Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Reservation Addition | Reservation Successfully Added | Day = Sunday | Addition Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Reservation Deletion | Reservation Successfully Deleted | Id=1 | Deletion Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Category Add | Category Successful Added | Category name = Machine | Category Successfully Added | Pass | Ashish Dongol | 2019/06/30 |  |
| Category Delete | Category Successful Deleted | Id=1 | Category Delete Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Product Add | Product Successful Added | Ink Machine, 50, 30, Cool Machine | Addition Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Product Update | Product Successful Updated | Ink Machine, 50, 30, Cool Machine addasd | Update Successful | Pass | Ashish Dongol | 2019/06/30 |  |
| Buy Product | Product Successful Bought | Esewa, Bafal, 8, 984654 | Successfully bought | Pass | Ashish Dongol | 2019/06/30 |  |
| Book Artist | Artist Successful Booked | Day = Sunday | Successfully booked | Pass | AshishDongol | 2019/06/30 |  |

# Chapter 6: Other Project Issues

## Chapter 6.1: Limitation of the project

There are few limitations in my project they are: -

* Payment system of the application is not fully functional
* There is no searching of artist in the website.
* There is no searching of tattoos or products in the website.

## Chapter 6.2: Future Works

I have thought of many features that are left for future works. Some of them are: -

* Chatting system with other users or artists.
* Fully fledged notification panel for msg, transaction notification.
* Involving of cart system for the website.
* Innovation of tattoos by contacting with artists.

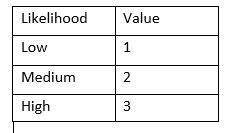
## Chapter 6.3: Risk Management

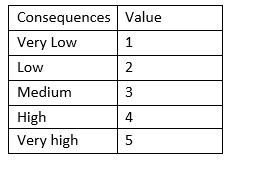
For a system risks are simply losing of data/information of clients, staffs etc. These risks can also damage the functionality of the system as well as reduce the quality of the system. For preventing all of these risks we use risk management while the system is being developed. Here risk management is simply a process which identifies a risk and takes certain measures to minimize that risk from the system.

There are few steps risk management uses. They are: -

* Identifying of risk
* Analyzing of risk
* Evaluating of the risk
* Curing of Risk
* Monitoring and reviewing of risk

For calculating how much of risk can impact my system I have used the following table: -





By taking of the given values we can thoroughly calculate the impact of risk in the system. Here is a table which shows the following ways of how risk is managed in the system.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk Type | Risk | Likelihood | Consequences | Impact | Action type | Action |
| Non-Technical | Out of budget project. | 2 | 3 | 6 | Avoidance | Performing of a meeting where thorough planning is done about all the requirements and features of the system. |
| Occurrence of Natural Disaster | 2 | 4 | 8 | Contingency | Backing up of important data at a regular basis. |
| Longer deadline than planned | 2 | 4 | 8 | Contingency | Keeping of limited time zones on each phase for completion before deadline. |
| Technical | Degrade of performance due to hardware. | 2 | 3 | 6 | Avoidance | Upgrading of hardware for better performance. |
| Hard disk Crash | 3 | 5 | 15 | Contingency | Setting up a cloud backup before starting of project. |
| Software Bugs | 3 | 5 | 9 | Avoidance | Releasing of patches frequently to solve bugs. |

## Chapter 6.4: Configuration Management

For this system I have chosen to use the waterfall model so I have configured a directory according to that model. This directory is created in the personal computer as well as uploaded in GitHub with the repository name of Royale-Tattoo-Service as per the link is <https://github.com/Ashish911/Royale-Tattoo-Service> and for the coding of the system a different repository is made and the link is <https://github.com/Ashish911/Code_Implementation>.

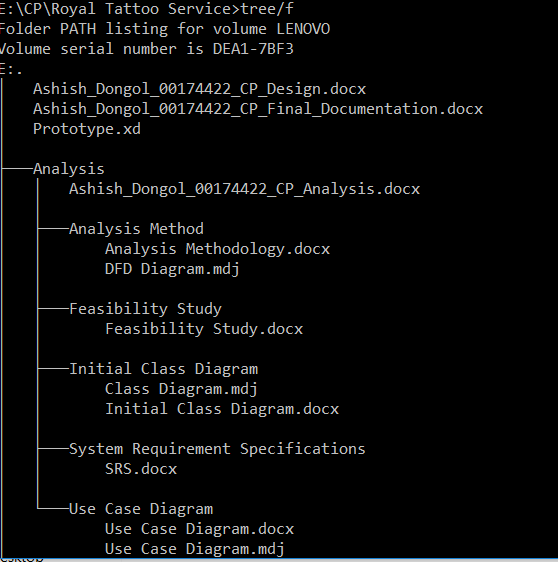
Here is the directory as per the model: - 

Figure Directory 1

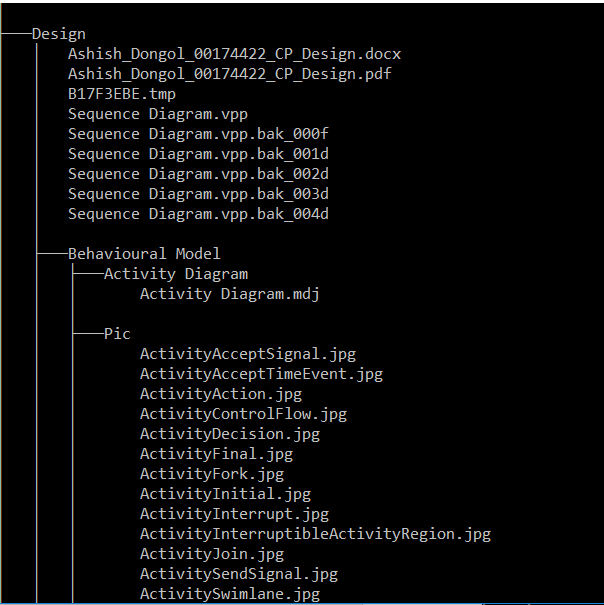


Figure Directory 2

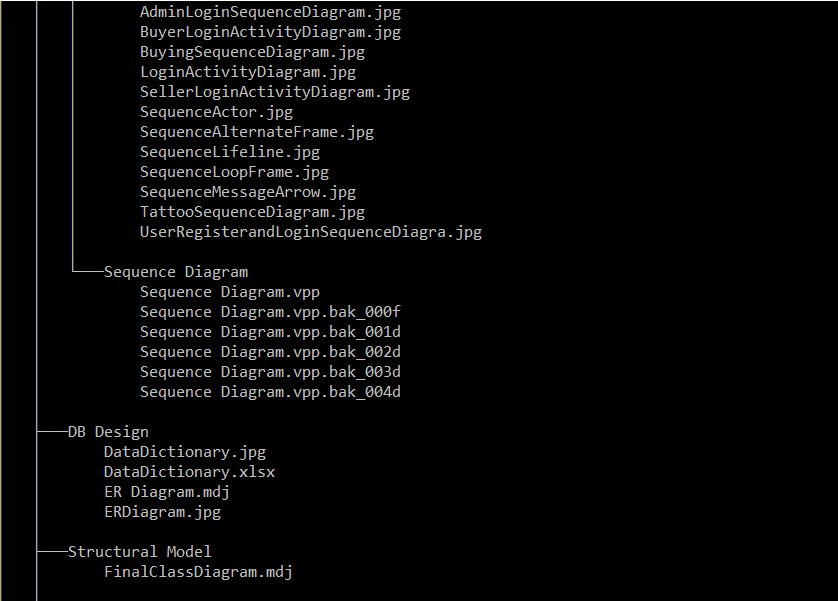


Figure Directory 3

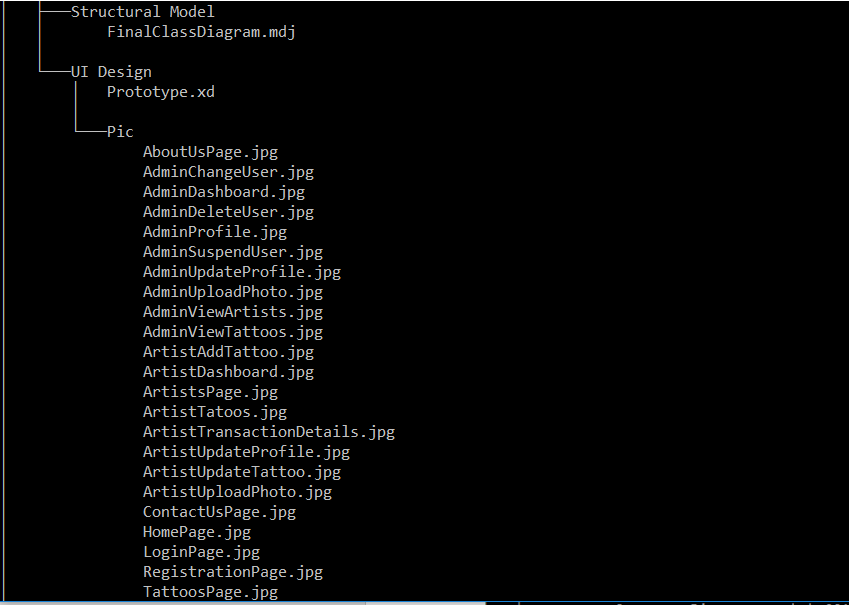


Figure Directory 4

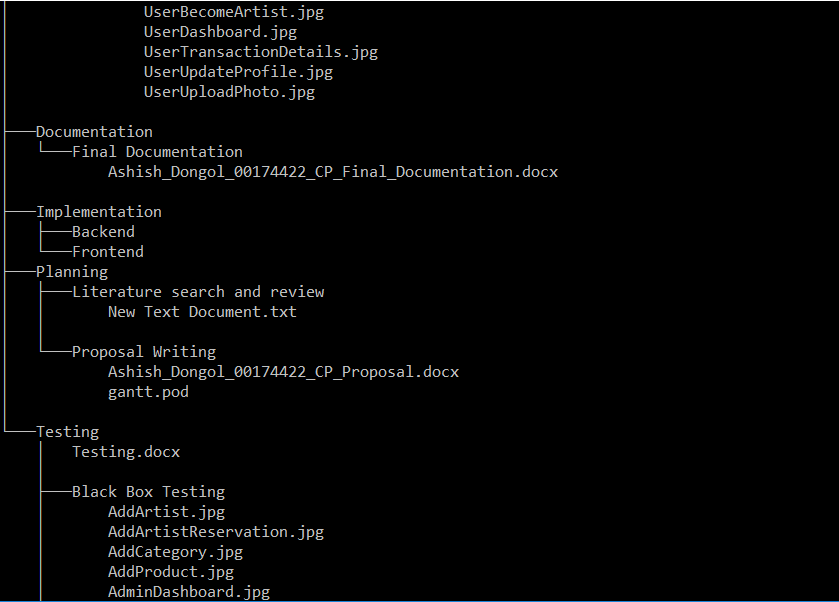


Figure Directory 5

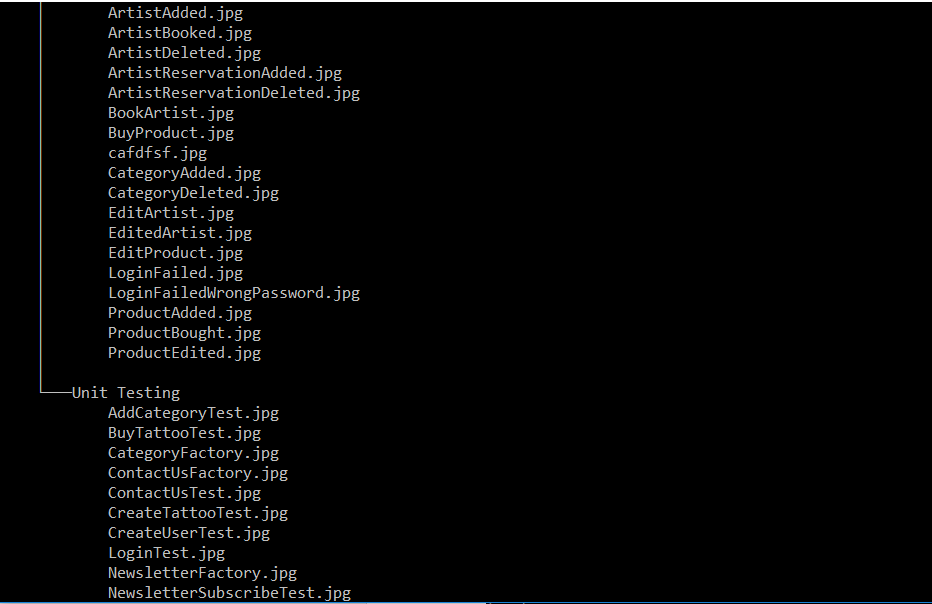


Figure Directory 6

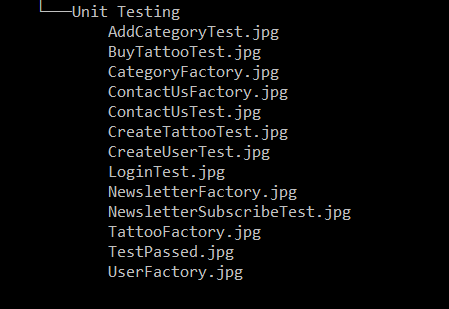
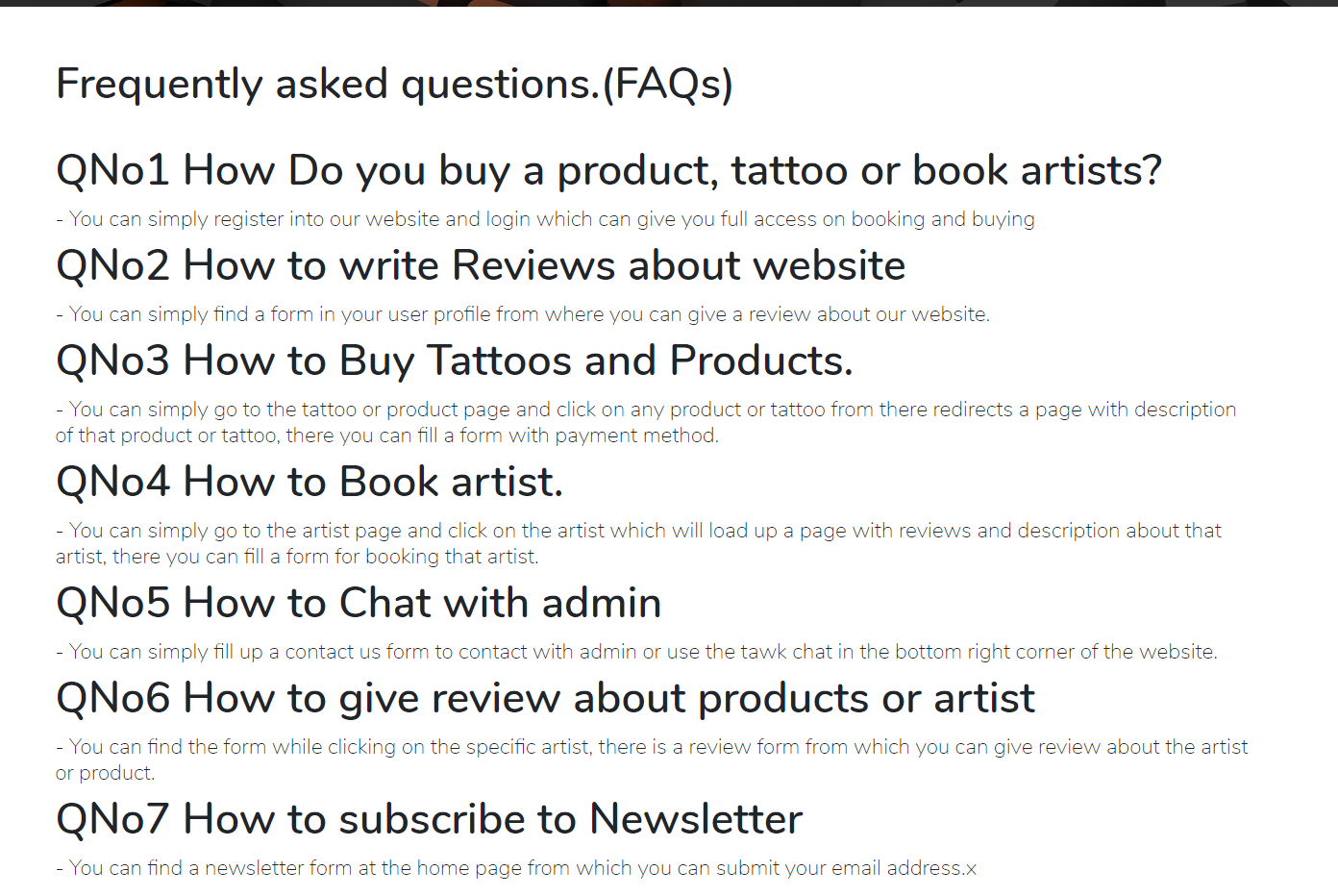


Figure Directory 7

## Chapter 6.5: User Manual

User Manual is just a guide book or a helper for users on how to use the website properly. For this system I have made a separate web page where a manual is given on how to use it and it is shown below: -



# Conclusion

By completing of all the required task, I have successfully completed in developing of the tattoo service system. By analyzing of all the required functionalities, the software is then design on the basis of that by making of crucial diagrams for developing such as class diagram, etc. On the basis of these diagrams the system is developed and later tested to see if it works properly and after that the project is documented.