# Analysis

## 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 every thing 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.

## 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.

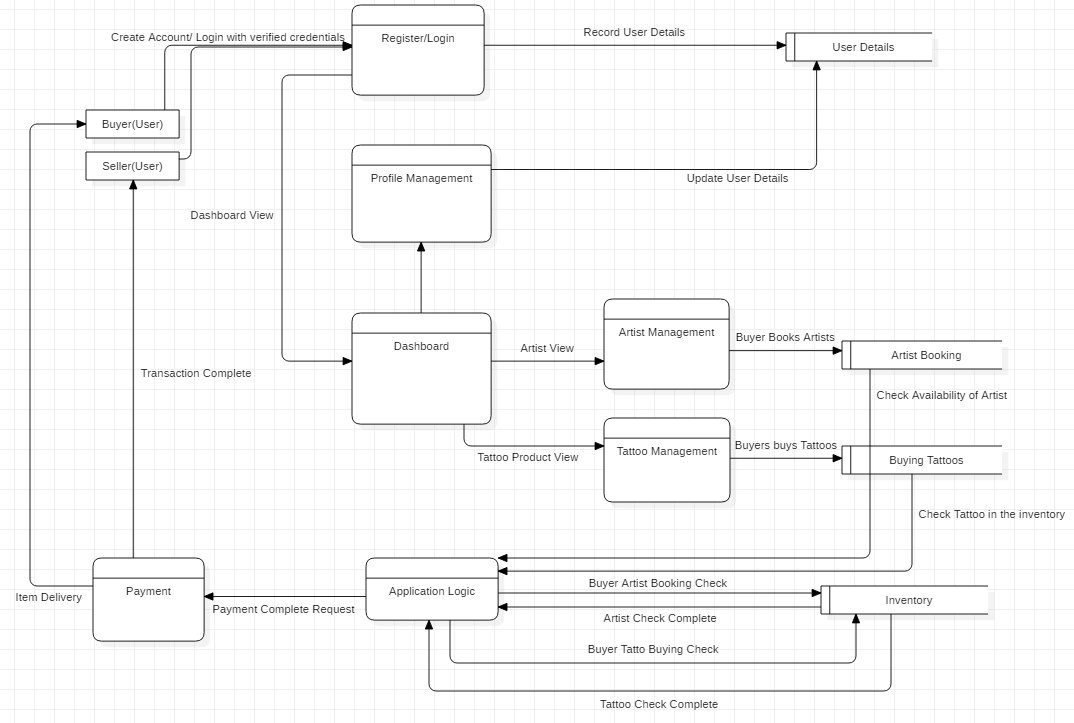


Figure 1 Data Flow Diagram

## 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.

## 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.

### 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 |

### 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. |

### 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 |

### 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.

## 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.

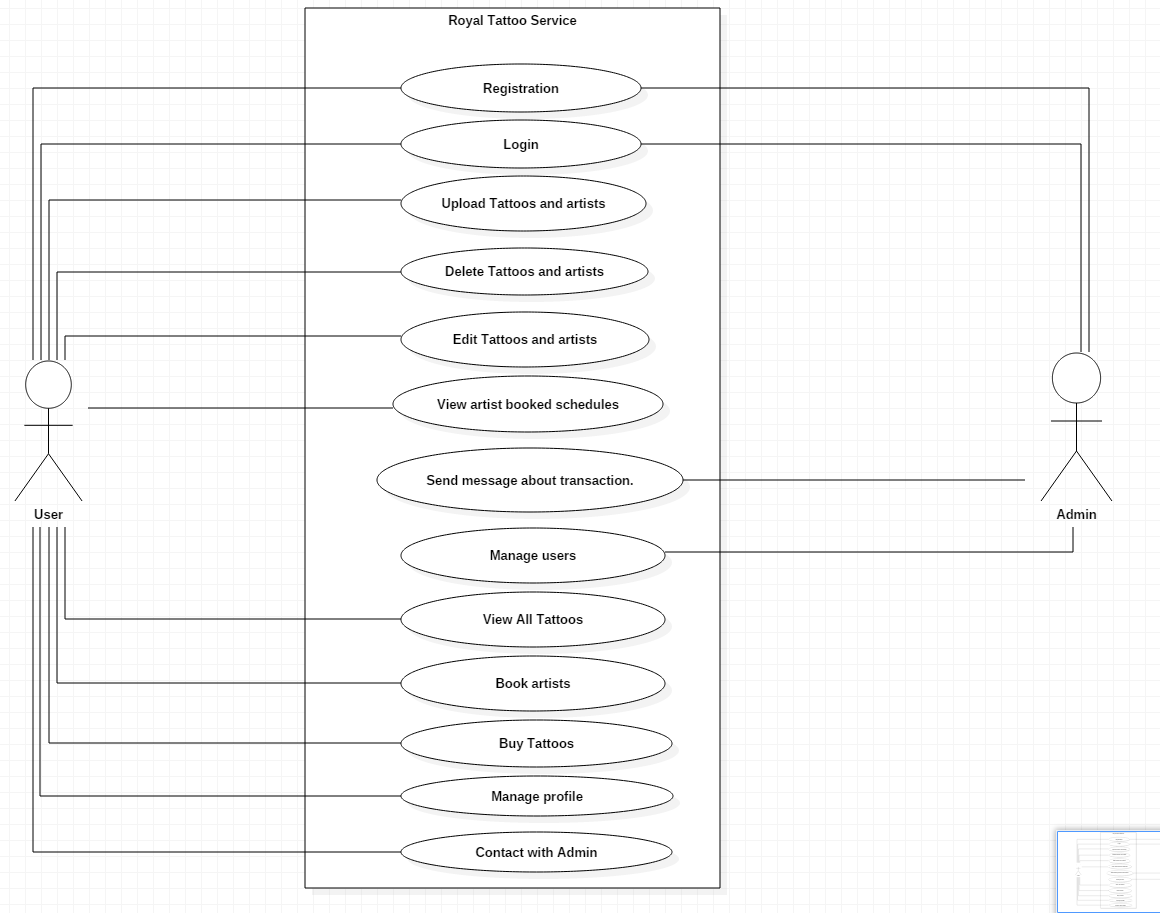


Figure 2 Use Case Diagram

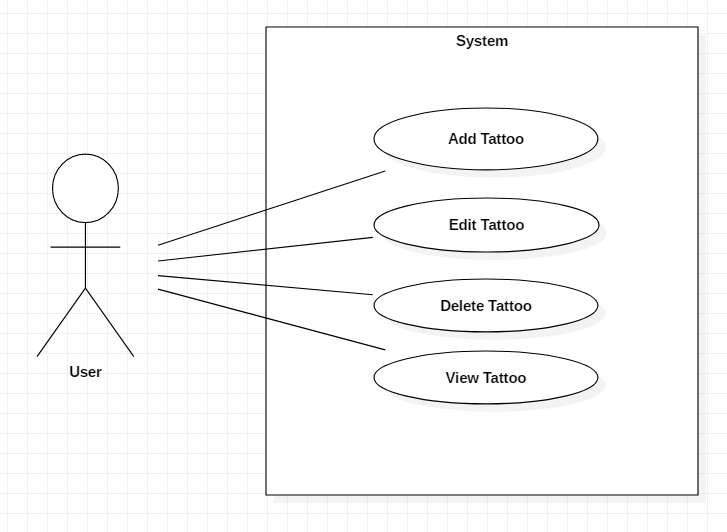


Figure 3 Use Case Diagram CRUDE

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 |

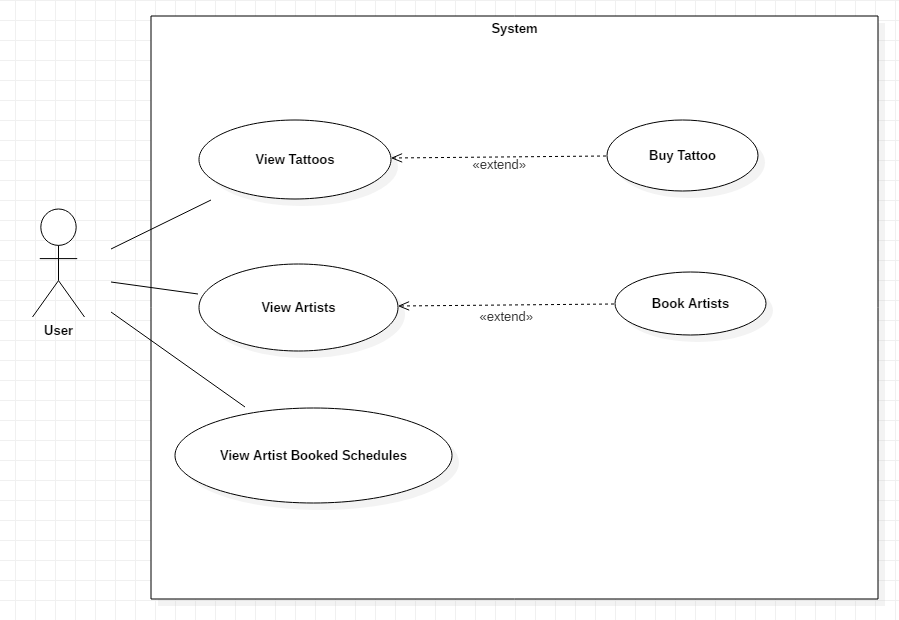


Figure 4 Use Case Diagram Viewing Artist and Tattoos

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 |

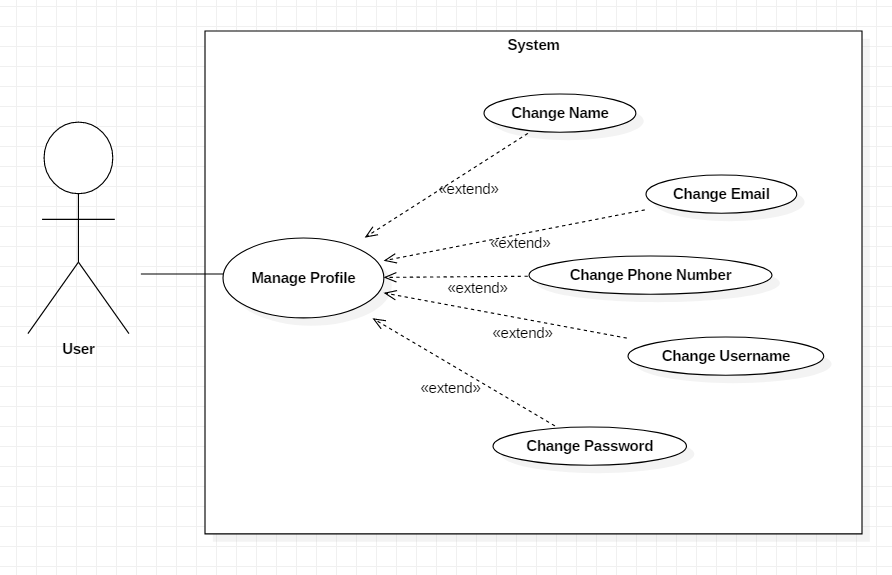


Figure 5 Use Case Manage Profiles

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. |

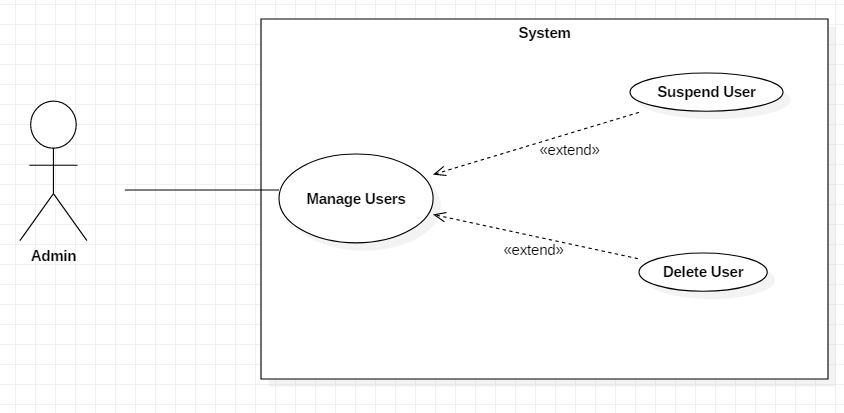


Figure 6 Use Case Manage Users

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. |

## 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 |

## 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:

