# Chapter 2: Analysis

# 2.1 Introduction

The initial stage of building a software development process is done through analysis. It can be also considered as the one of the key stages of software development process. Better the analysis better will be the product. It has main role in project completion. In analysis part we gathered information from sources i.e. staff of an organization. It gives developer an overview of what user wants from a system. The requirements thus gathered will help in identification of data need and information about current system of organization which will help developer to build a correct system with correct features. It consists of two parts.

**Requirement Analysis**

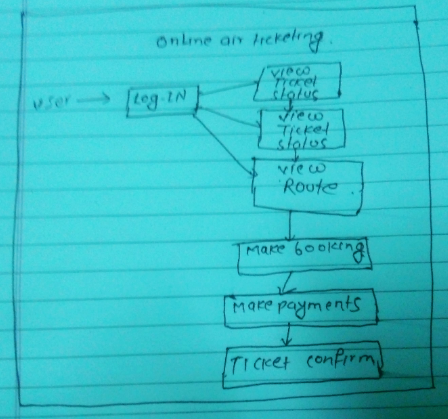
Requirement analysis is the first stage of analysis. It helps analyst to gather all the information regarding the system from users. It will give the outlook of what the user wants from a system. It is also gives feedback of user for a current system and their level of satisfaction towards the current system. There are different way of gathering information e.g. observation, interviews, focused group, historical data etc.

**Requirement Specification**.

Requirement specification is the method of collecting several information that are gathered from requirement analysis. It is then documented as SRS (System Requirement Specification) which will include all the user's needs form system. It can be considered as legal document and helps to solve issues in future.

**Rich Pictures**

Rich pictures are symbolic representations of natural or social frameworks. They can place compound circumstances and differentiate unseen issues by comprising every significant constituent and partners of a framework.



Figure; Rich Picture showing main functions inside system

# 2.2 Information Gathering Techniques

Information gathering techniques refers to the gathering of information through different means or sources. I have used two dissimilar methods for the collection of information which are defined below.

**Interviews**

It refers to the conversation between two or more people for the collection of related information. It is more effective than others methods. We can easily interact with the users and take their requirements, views, ideas etc. It is most common method used everywhere and gives very useful information than others method.

**Focus Group**

It is similar to interviews but it has limited number of selected people i.e. 6 to 10 people. It plays vital role in gathering information like opinion, concern, perception and knowledge of selected people regarding the given topic. It is an effective method to gather client experience. . Focus group is usually applied marketing tools that allows relations to create the product and services that are obstructed.

# 2.3 Feasibility Study

Feasibility study can be defined as the research or investigation that are carried out for the proper completion of project. Any project can be finished effectively by gathering all the requirements through the help of feasibility study. It is guidance to complete project and how a business can run for long term. Feasibility study of different activities like technical feasibility, legal feasibility, economical feasibility etc. are performed to ensure that that no problems occur during the implementation part and to successfully run any organization. It is carried out in the analysis part of the project.

**Types of Feasibility Study**

1. Economic Feasibility

It is done to ensure that the project which is going to develop is within the budget limitation or not. It gives us details of all the expenses of the project. It also helps in the budget identification for different sector within the project. It is analysis that is done to know the profit against the investment made on the project or product.

1. Technical feasibility

It examines whether the existing technology i.e. hardware and software is sufficient or not for the completion of proposed project. If any technical items are lagging then what technical items should be modified and what new things should be bought i.e. hardware and software.

1. Operational Feasibility

It is type of feasibility which tells us that clients/users will find the product easy to use or not. It also tell us if any kind of training should be provided to users for operating any task. It also tells what things should be needed if training is necessary.

1. Schedule Feasibility

It is the allocation of time for the evolvement of new system. Division of time for all the category is also included. It gives us clear vision about the impact if the product is not delivered in time.

1. Legal Feasibility

Legal feasibility talks about the legal and ethical matters of the project. It ensure that proposed system will meet all the legal requirements and ethical requirements.

# 2.4 Analysis methodology

In this particular project, I have used object-oriented methodology. I have selected this methodology because it simple to understand and also reduces the complexity of software object models. It allows changes and structure are also clear. It enables code reusability as well as software components.

# 2.5 System Requirement Specification (SRS)

The System Requirement Specification is an official announcement of the functional and non-functional requirements of the system. It acts as agreement between developer and client for whom the framework is being manufactured. Its developer duty to provide all the features that are included in the specification. The client agrees to find the item fulfilled in case it gives the restrictions verified in the SRS.

1. **Functional Requirements**

Functional requirements are those requirements which defines what are the main functions inside the system. It points out the main activities of the system like calculation, data manipulation, data processing, login, registration etc.

Functional requirements of my system are:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FID | Functions | Data | Rational | Dependency | Remarks |
| F01 | Registration | User details | For users | F01 | Getting access inside system |
| F02 | Login | Username and Password | For users security | F02 | Welcomes to dashboard |
| F03 | View Flight Detail | Time place date of flight | Information for users | F02 | All details of flight |
| F04 | View Ticket Status |  | Price for particular flight | F02 | Prices of all flight |
| F05 | Book Flight | Provide all personal details | Booking flight | F02 | Flight booked |
| F06 | Flight Availability | ticket not booked | flights available for users | F02, F03 | which flights are booked and what are remaining |
| F07 | Booking Confirmation | Booking confirmed | Ensure your flight is booked | F04 | Insurance for your booking |
| F08 | Reschedule  Ticket | Edit flight information | Making new updates | F04, F07 | Editing all flight information for new booking |
| F09 | Ticket Cancelation | Cancel ticket booking | Due to some reasons | F04, F07 | Cancelation of booked ticket |
| F10 | Phone Access | Phone Number | For customers | F02 | Direct contact to head office |
| F11 | Update Users Profile | User information | For users to edit information | F01, F02 | Allow users to update their information |
| F12 | Admin Login | Admin Details | Admin | F12 | Admin can login |
| F13 | Admin Profile update | Admin Details | Admin | F12 | Admin can update profile |

Figure: Functional requirements

1. **Non- Functional Requirement**

Non-Functional requirements refer to the products excellence product. It is generally used to justice the systems operation. They are basically categorized into two different sectors.

1. Execution qualities: Usability
2. Evolution qualities: Testability, Maintainability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nid** | **Functions** | **Data** | **Rational** | **Dependency** | **Remarks** |
| N01 | Responsive | - | Gives different solutions |  | Fits in screen of different size. |
| N02 | Usability | - | Easy to use, User friendly |  | Users must find it easy to use |
| N03 | Maintainability | - | Change Adaptability |  | Should be maintained easily in case of errors |
| N04 | Reliable | - | Accurate output |  | Must give correct output of given input |
| N05 | Robust | - | Run on any platform |  | Must be able to run on several platforms |
| N06 | Multi-Browser support |  | Run on multiple browsers |  | Supports multiple browsers |
| N07 | Scalability |  | Workload handled easily |  | Data flow handled easily |

Figure: Non-Functional requirements

1. **Prioritization**

Prioritization means giving focus to those functions that are most to the system. It ensures that important things are managed accordingly. It acts as a rule to project team for ranking items. It also helps in allocating more time to those which are in top priority.

In this particular project I have used **MoSoCoW** prioritization

• It guides vital and expenditure plans.

• It inspires people to review on their requirements.

• It impulses people to plan and achieve improved work.

**Moscow (Must, Should, Could, Wont)** it is basically originates form DSDM.

**Must have:** It enlists the requirements that are most to the system. It plays vital role in project success because wrong requirements can lead to failure. It is like basic needs to the system and are not negotiable.

**Should have:** It enlists the requirements that are important but not the vital. They can be consider important but shouldn’t be included always.

**Could have:** It enlists the things that seems necessary but important to the system. Its implementation depends upon the time remaining to complete the project.

**Won't have:** These requirements are not necessary but can be implemented in future at development phase if needed. They does not play any important role in the project success.

**Moscow Prioritization**

|  |  |  |
| --- | --- | --- |
| **ID** | **Functionality Requirement** | **Moscow** |
| F01 | Registration | Should have |
| F02 | Login | Must have |
| F03 | View Ticket Status | Must have |
| F04 | View Flight Details | Must have |
| F05 | Book Flight | Must have |
| F06 | Flight Availability | Should have |
| F07 | Booking Confirmation | Must have |
| F08 | Reschedule Ticket | Should have |
| F09 | Ticket Cancellation | Should have |
| F10 | Phone Access | Should have |
| F11 | Update Users Profile | Could have |
| F12 | Admin Login | Must have |
| F13 | Admin Profile Update | Could have |
|  | | |
| **ID** | **Non-Functional Requirements** | **Moscow** |
| N01 | Responsive | Must have |
| N02 | Usability | Must have |
| N03 | Maintainability | Must have |
| N04 | Reliable | Should have |
| N05 | Robust | Must have |
| N06 | Multi-Browser Support | Must have |
| N07 | Scalability | Won't have |

Figure: Moscow prioritization

# 2.6 System Architecture

System architecture is theoretical model that characterizes the structure, conduct and perspective on framework. It portrays the portrayal of entire framework. For this venture, I have use 3-level structure. System engineering is applied model that characterizes the structure, conduct and perspective on framework. It depicts the portrayal of entire framework. For this task, I have use 3-level structure.

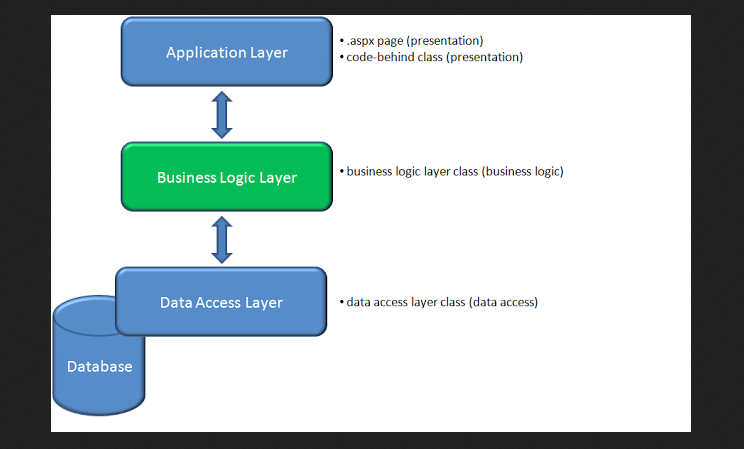


Figure: Three tier structure

**Model**

The model handles all the data related logic that user work with. This represent data that is transfer between view and controller.

**View**

The view handles all UI logic of application.

**Controller**

Controller acts as intermediary between model and view to process incoming requests and logic

# 2.7 Use Case Diagram

Use case diagram is the diagramed representation of what functions does the actor i.e. admin, user do in the respective system. It gives the maximum functionality of any system. It allocates the function assigned to different all the actors involving inside the system. They are used to collect the system requirements containing internal and external impacted.

The purpose of Use Case diagram is given below:

* It gets the dynamic piece of the system.
* It shows how on-screen character cooperate with the system.
* The framework design is approved.
* Determined content of system.

The symbols used in the use case diagram are given below. They help us to show interaction between system and actor.

**System:** By using rectangle shape draw our boundaries of the system which contains the use cases. Actor should be placed outside the system boundaries.



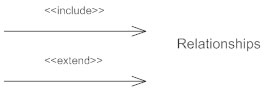
**Actors:** Actors are something that interact with our system .it is denoted by the symbol.



**Use Case:**  Use case is oval in shaped which represent various uses that an actor might have. It is denoted by symbol the symbol.



**Relationship:** Relationship can be labeled by arrow either uses or extends among use cases. To perform a task on use case is needed by another is indicates by uses relationship where extends indicates optional under a certain use case.



Use Case Diagram of My System

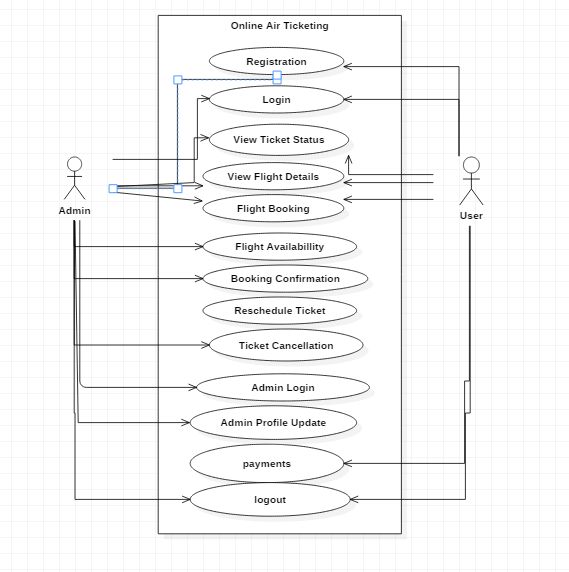


Figure: Use case diagram for admin and user of system

# 2.8 NLA (Natural Language Analysis)

Natural Language analysis is a method of finding nouns and verbs which will be possible classes and methods in a system. It is one of the best ways of finding the classes and behaviors. It will minimize the redundancy and the efficiency of system is also increased.

**Scenario**

In the present situation people has become very modern and technology-dependent. They want fast and easy method for communication as well as in working platforms. They don’t want to waste their valuable time in unnecessary things like standing on queue for a simple work. It is like wasting time for them. They want something which they can access from their home by using their mobiles phones, computers or by using internet services. They are using technology in their day to day activities like purchasing products, ordering goods, foods, air ticket booking, booking movie ticket etc.

Online Air Ticketing is a web application where people can book ticket for specific date as well as they can cancel ticket if emergency occurs. It also includes price of the ticket with ticket class like normal, business, economy etc. You can also see the time of flights of many airways’ companies. It also shows the flights are available or not. As you travel by air ticket, you can also search for the hotel around local areas using this web portal. You can easily book ticket a month before travelling. It will have facility of phone access and rescheduling ticket. You can get cheapest flight for several places of country using this application. All you need to do is register first in application and you can book air ticket simply by browsing. Tickets are available for all the airports inside Nepal and it has 24 hours services. Users can also cancel the ticket if emergency occur. Admin can login and edit their profile inside the system.

**Identifying Nouns and Verbs**

|  |  |
| --- | --- |
| **Nouns** | **Verbs** |
| People, Platform, Queue, Mobile Phones, Computers, Internet, Flight, Products, Goods Ticket Booking, Movie Ticket, Online, Application, Registration, Class, Business, Economy, Companies, Available, Admin, Nepal, Login, Profile, | Fast, Communication, Waste, Access, Purchasing, Ordering, Booked, Includes, View details, Cancel, Standing, Occurs, Search, Rescheduling, Travelling, Browsing, Registered, Edit, Make payments, |

Figure: Nouns and Verbs identified from NLA

After removing all the synonyms from the identified nouns and verbs and ambiguous word. Also, the noun out of scope and technical words are removed. Finally, we are able to find the given classes and behaviors inside the system.

|  |  |
| --- | --- |
| Classes | Functionality |
| Admin, login, Registration, Booking, Confirmation | Users Login, Users Registration, View Flights Details, View Ticket Status, Reschedule Ticket, Cancelation, Profile Update, Make Payments, Availability |

Figure: Classes and Functionality from NLA

# 2.9 Initial Class Diagram

System static view is representing by class diagram. It shows the classes within a model. By consuming the outcomes of the crc procedure initial class diagram is established. It can signify all these things fairly effortlessly. Icon of the class is just rectangles which is separated into the three components.

* The uppermost compartments specify the name of the class.
* The middle section specifies a list of attributes.
* The bottom section specifies a list of processes (Methods).

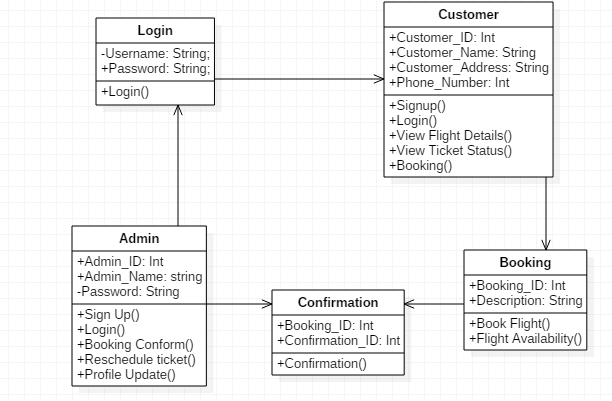


Figure: Initial class diagram with attributes and operation

# Conclusion

In the analysis period, I have done several things. All the requirements were identified. Then the functional and non-functional requirements were identified. After that Moscow prioritization plays an important role in identifying the important requirement for the system. Design was marked by the help of MVC. I have utilized use case diagram that shows the interaction of actor with the system and at last initial class diagram was established.

# 3 Designing

# 3.1 Introduction

Designing is supposed to be an important part in the development process of software. Design Specification portrays how system can perform the necessities which is sketched in functional requirements. Depending upon the frame. This can include guidelines on testing particular requirements, arrangement settings, or audit of capacities or code. Here in the functional specification all outlined requirements must be addressed.

# 3.2 Tools

In this particular project, I have used different tools or software for the construction of ER diagram, Class diagram, Database, Activity diagram etc.

* IDE (Integrated Development Environment) = Sublime
* Database = MySQL
* Class Diagram, Use case diagram, Activity diagram, and sequence diagram = Star UML
* Entity Relationship Diagram = Visual Paradigm

# 3.3 Structure Model

It is the static structure of the framework which demonstrates the communication, relationship of the object. It comprises of the Entity Relationship and class diagram which is given beneath. Structural models of software show the association of a system as far as the segments that make up that framework and their connections. Here we will center around class diagram for displaying the classes structure in a software system.

# 3.3.1 Class Diagram

The relationship between class and showing its methods or behavior and the attributes are called class diagram. It furthermore establishes the cardinality joining among or between the classes. It gives vision of an explanation of the model into the code. It is furthermore used for modeling that symbolizes the opinion responsibilities of the framework.

Final class diagram for online air ticketing

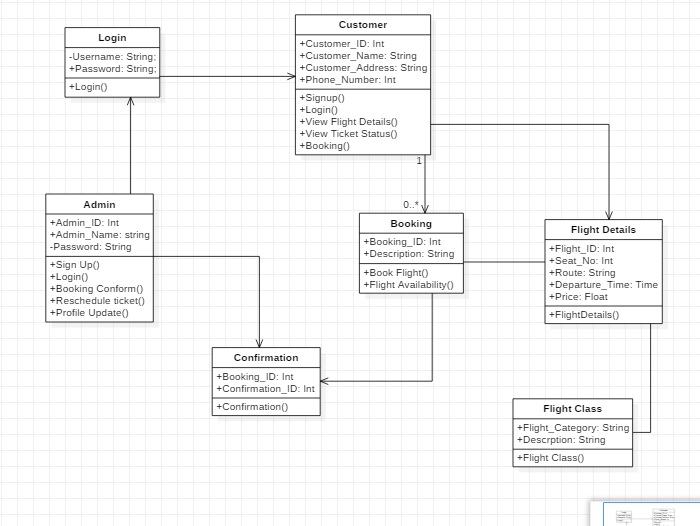


Figure: Final Class Diagram for Online Air Ticketing

Justification: Above class diagram includes all the classes involved in the online air Ticketing. Customer and Admin need to log in inside the system and booking are performed by customer and confirmation is dependent of admin. Booking also uses flight details and flight class.

# 3.3.2 Data Flow Diagram

Data Flow Diagram can be defined as the movement of data inside the system or framework. It consists of information, output, information sources etc. It also consists of sub process through which information passes.

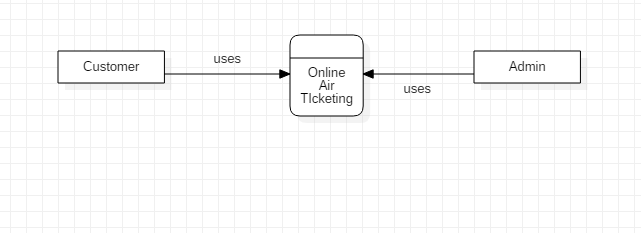


Figure: Level 0 Data Flow Diagram

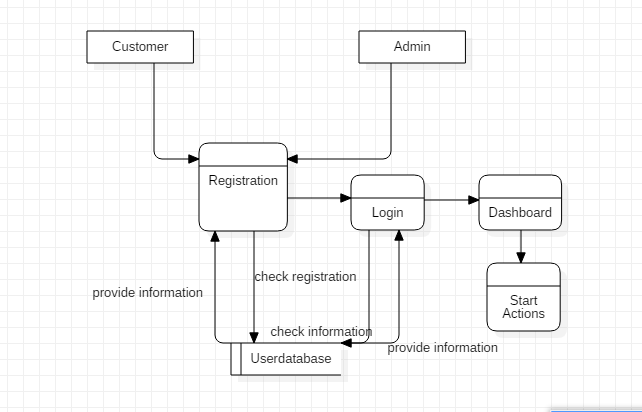


Figure: Level 1 Data Flow Diagram

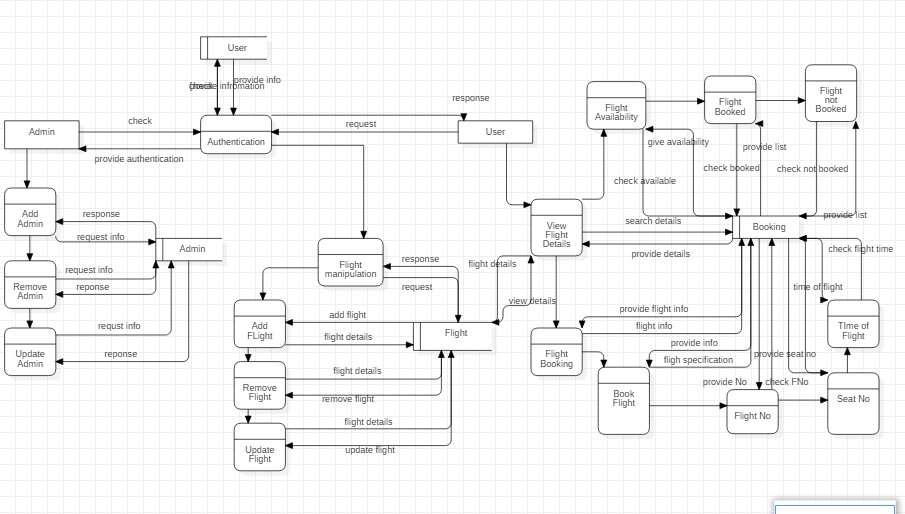


Figure: Level 2 Data Flow Diagram

# Justification

The above diagram shows the interaction inside the system. It includes entity and their behavior. Sequence diagram shows the data fetching from database. It is showing all the functions of all the entities.

# 3.4 Data Modelling

Data Modelling is a procedure used to characterize and inspect information requirements expected to help the business forms exclusive the extent of relating data frameworks in associations. In this manner, the procedure of information representative includes proficient information modelers working closely with business partners, just as potential clients of the data framework.

# 3.4.1 Entity Relationship Diagram

Entity Relationship Diagram can be defined as the graphical representation of entities and relationship between them. They can also be called as entity relationship model. It relies mainly on view of realistic world which consist of an object collection known as entities. It also includes attributes related with entities.

Entity Relationship Diagram for Online Air Ticketing

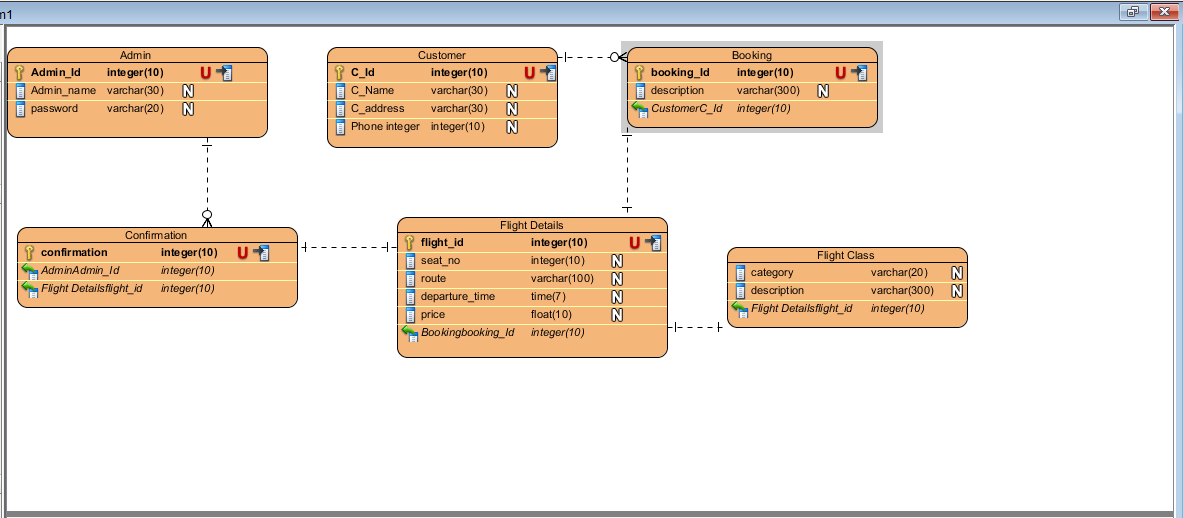
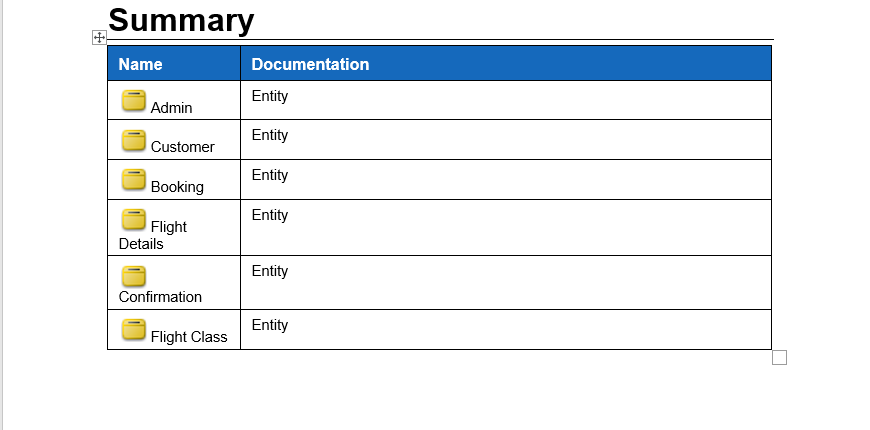


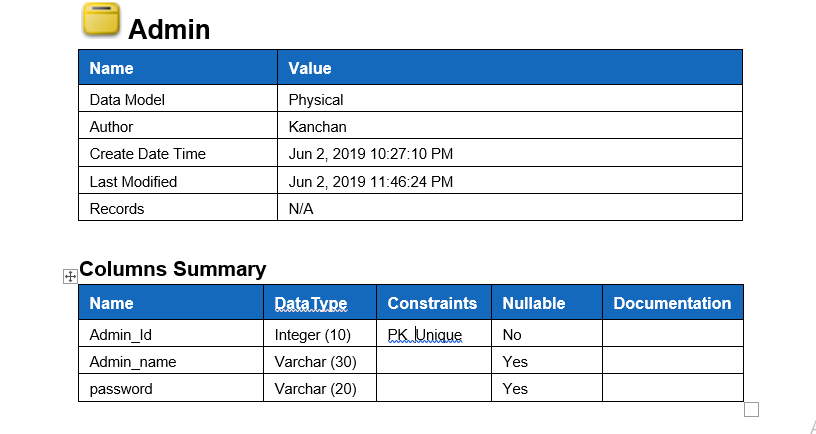
Figure: Entity Relationship Diagram of online air ticketing

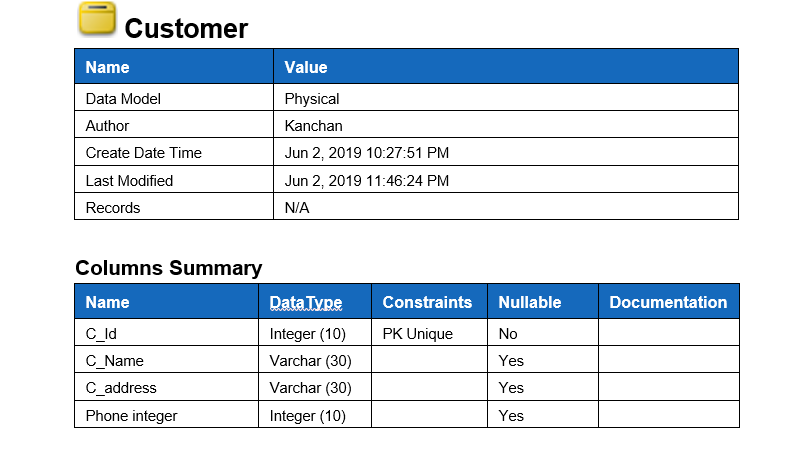
# Justification

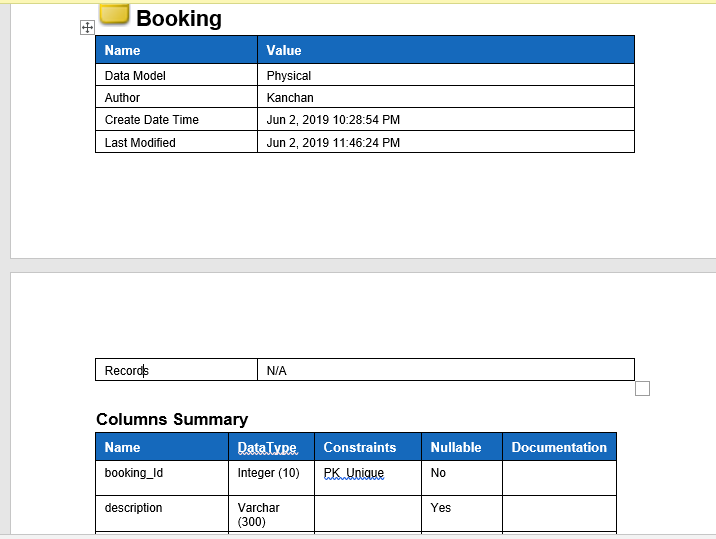
The above diagram shows the ER diagram which is showing entities like admin, customer, booking, confirmation, flight details, flight class. Customer and booking have one to many relations and other have one to one relationship.

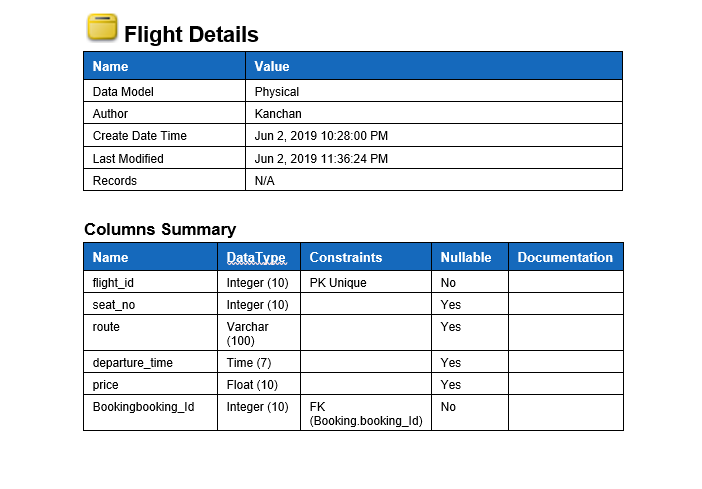
# Data Dictionary

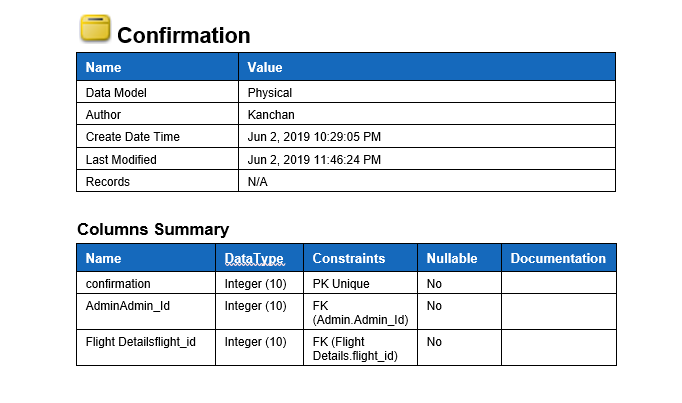












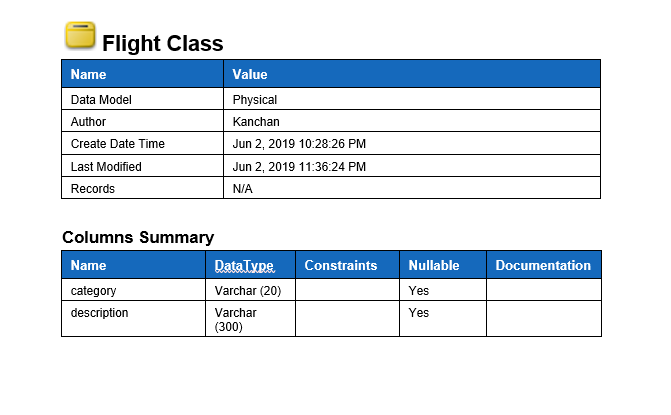


Figure: Data Dictionary for online air ticketing

# 3.5 Behavioral Model

Behavioral model is called dynamic model. It is time dependent so it can have changes while running. Some of the importance of behavioral model are.

* Demonstrate the controlled architecture of the platform
* Meeting the needs of framework
* Architecture of system are explainable

# 3.5.1 Activity Diagram

Activity Diagram can be defined as the flow of activity inside the system. It shows the systaematic order of flow of all the activities inside the system. Its a graphical representation of overall system functions.

Activity Diagram for Online Air Ticketing

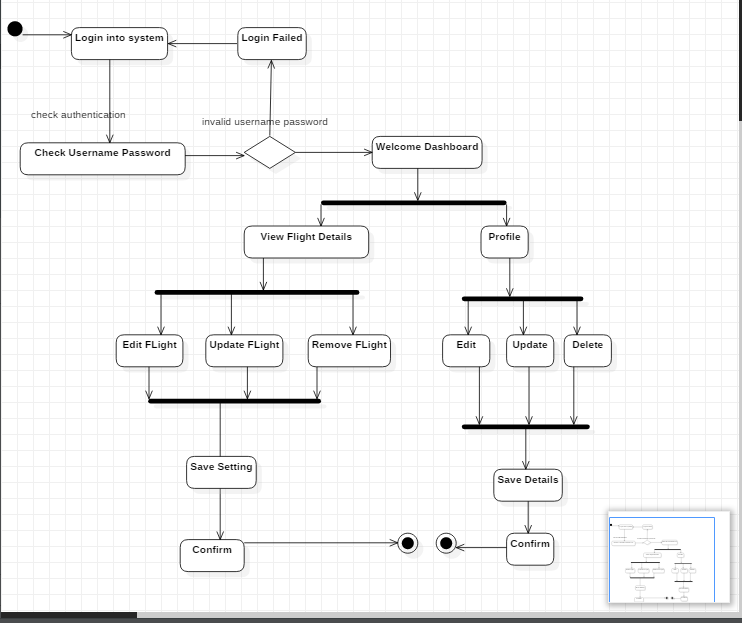


Figure: Activity Diagram for Admin

# Justification

The above diagram is an activity diagram for admin which includes all the functions that are performed by admin inside the system. It gives clear look of what are the task that are allocated for admin and his/her criteria.

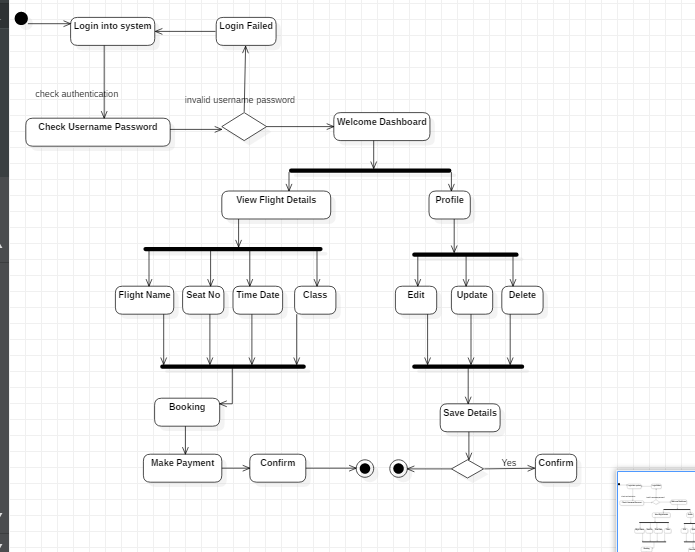


Figure: Activity Diagram for Customer

# Justification

The above diagram is an activity diagram for customer which is showing all the activities related to customer inside the system. What things are allowed to user and what things are not allowed to them. It also shows the flow all functions.

# 3.5.2 Sequence Diagram

Sequence Diagram can be defined as the diagram that shows interaction in a sequence. It shows the classes and object involved in the system. It is mainly related to the use case of the system.

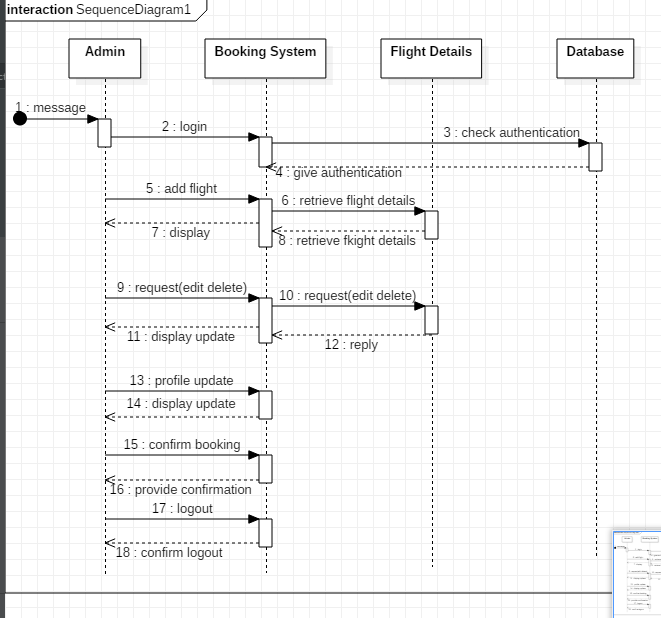


Figure: Sequence diagram for Admin

# Justification

The above sequence diagram is for admin it shows the task of the admin in sequence. It includes all the tasks performed by the admin in order. Here four entities are used and functions are performed inside them in sequence.

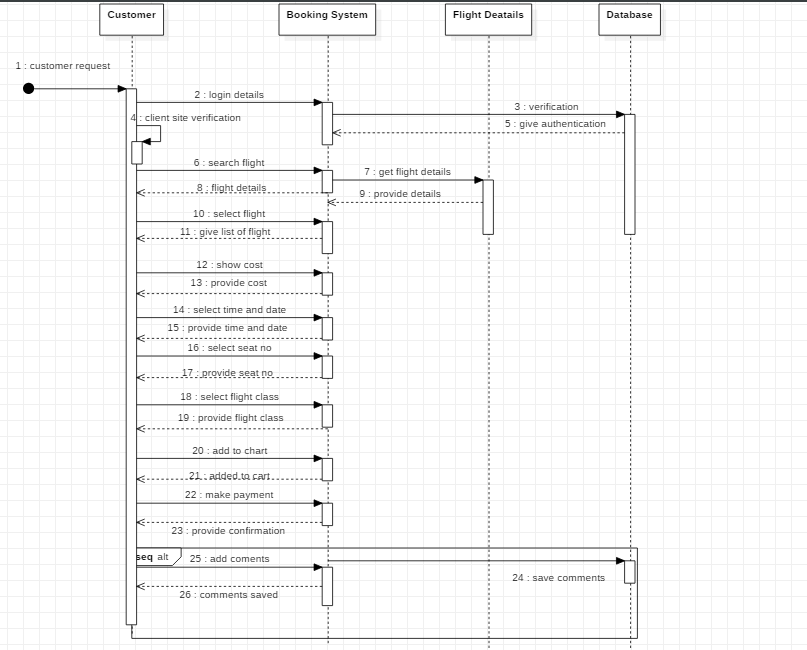


Figure: Sequence diagram for Customer

# Justification

The above is sequence diagram for customer which includes four entities and functions area performed between them. It consists of lifeline and message arrows which is performing tasks inside entities.

# 3.5 Prototype (UI Design)

A prototype is an early example, model, or arrival of an item operated to test an idea or procedure or to go about as a thing to be copied or increased from. It is a term applied in a collection of surroundings, counting semantics, plan, gadgets, and programming. A model is normally used to evaluate additional plan to advance exactness by framework specialists and customers.

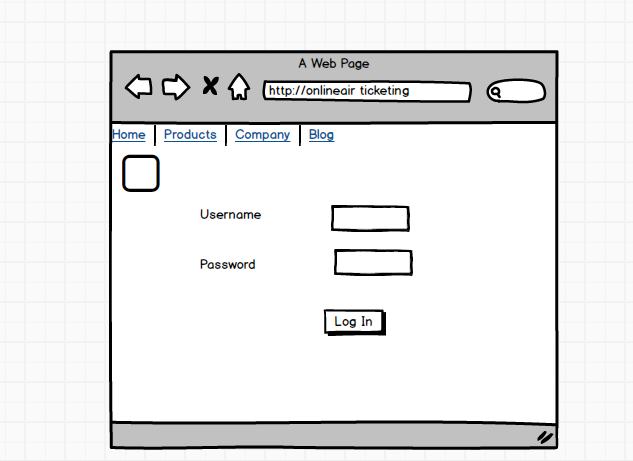


Figure: Prototype for Login

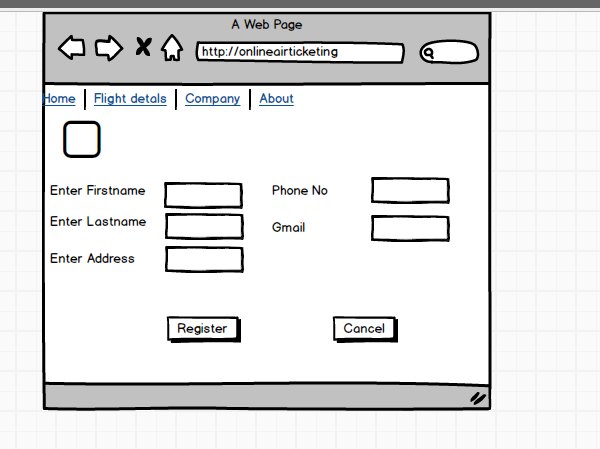


Figure: Prototype for Registration

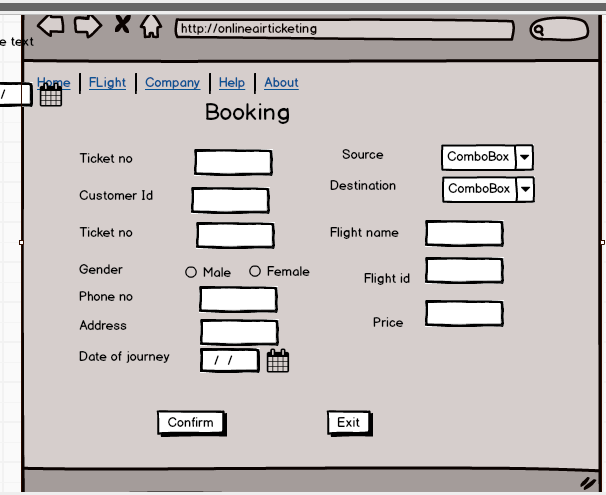


Figure: Prototype for Booking

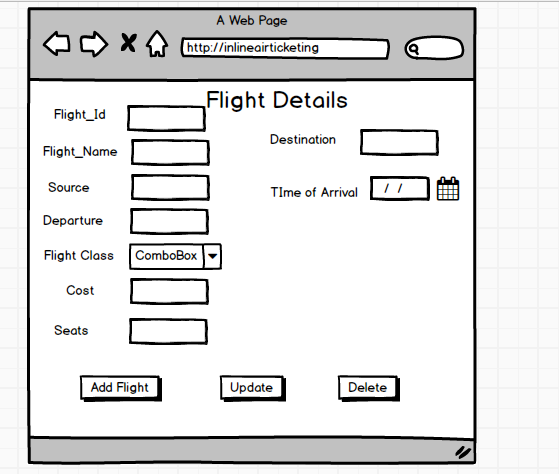


Figure: Prototype for Flight Details

# 3.6 Conclusion

Finally, the designing part of online air ticketing came to an end. It includes structural model which consists of class diagram and data flow diagram. It also consists of behavioral model which includes entity relationship diagram, activity diagram and sequence diagram. Protype is also designed which is showing outlay of interface inside the system. After designing implementation process will be carried out.