# Analysis

Analysis is the process of collecting the facts, identifying the problems, threats and many more. I t is basically done to study the system and its elements to identify objects of the system. It is also known as problem solving technique which helps in ensuring threat the system is working properly as according to the requirements.  
  
While starting this project I have to do different types of researches analysis also helps to know the problems which can be high or low and also analysis helps to maintain the usability of the system.

## Analysis methodology

For this project I have used hard methodology because it is a problem solving approach. Also, as we know hard approach can be used to develop small as well as large software so as our software is small for now and can be large in future so we have used this methodology as our analysis methodology.

USER DFD

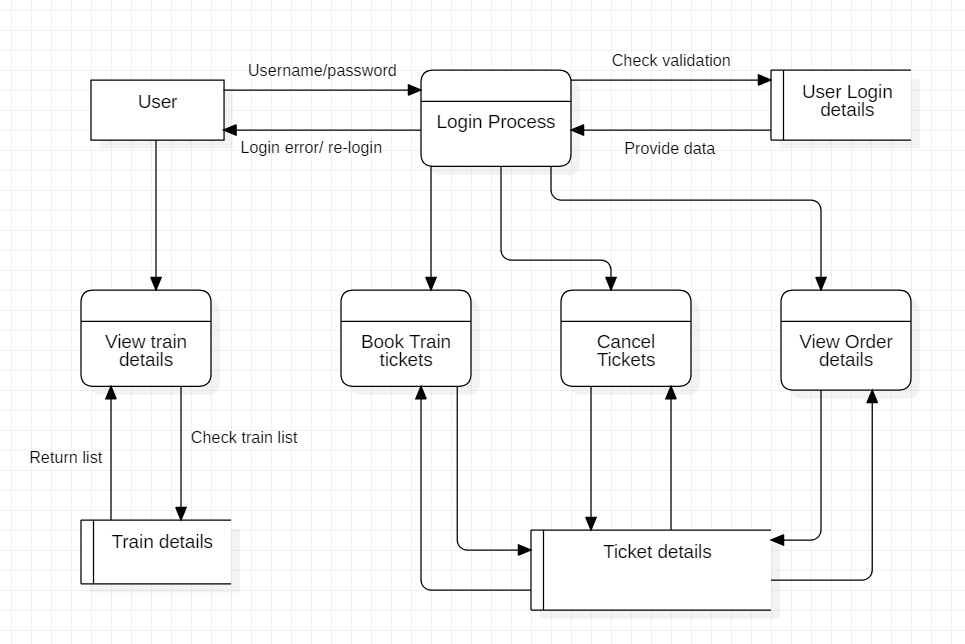


Figure 1: User DFD

Admin DFD

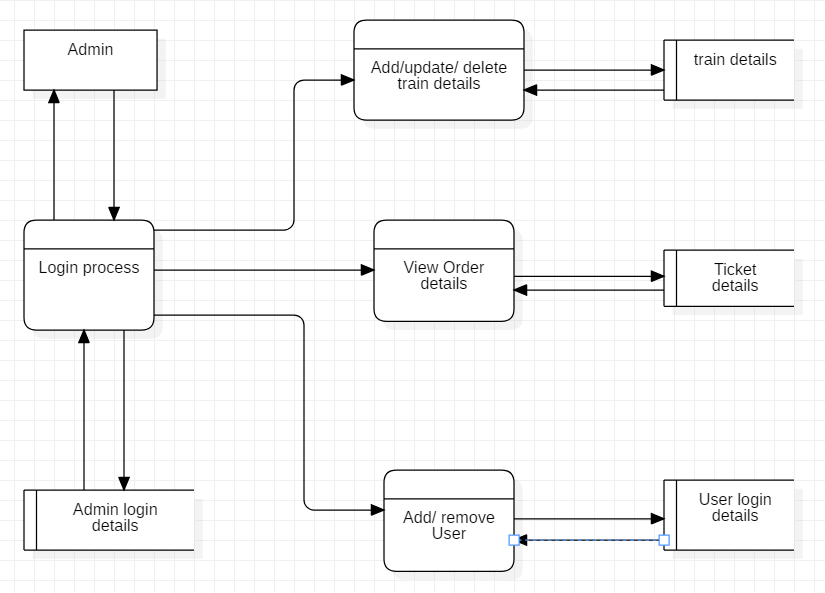


Figure 2: Admin DFD

# Information gathering methodology

Before the developing phrase of this project I have conducted many information gathering methodology in which I have conducted questionnaire and interview with various groups of peoples.

## Questionnaire

It is a method of research in which it consist of series of questions which is used for gathering information from the personals. For this methodology we can ask the random people about their point of views of the peoples of our circle. Below down are some answers to the questions by the peoples,

* Please rate your level of satisfaction if our software would have these functions:

|  | Very Satisfaction | Somewhat Satisfied | Neutral | Somewhat Unsatisfied | Very Unsatisfied |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ease of Installation |  |  |  |  |  |  |
|  | | | | | |  |
| Required Hardware |  |  |  |  |  |  |
|  | | | | | |  |
| Smoothness in OS |  |  |  |  |  |  |
|  | | | | | |  |
| Consistency with Interface |  |  |  |  |  |  |
|  | | | | | |  |
| Easiness of using the software |  |  |  |  |  |  |
|  | | | | | |  |
| Quality |  |  |  |  |  |  |
|  | | | | | |  |
| Instructions |  |  |  |  |  |  |
|  | | | | | |  |
| It can be used from anywhere |  |  |  |  |  |  |
|  | | | | | |  |
| Customer care |  |  |  |  |  |  |
|  | | | | | |  |
| Reasonable price |  |  |  |  |  |  |
|  | | | | | |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | | | | | |  |
|  |  |  |  |  |  |  |
|  | | | | | |  |

# Feasibility study

Feasibility study is said to be one of the important factor of development process. It shows us whether our system is cost feasible, time feasible or not. Feasibility study really plays important role on running the development process smoothly. Feasibility study also gives out that weather our project is legally, technically, economically, socially and politically fit or not. A good feasibility study gives out the strength and weaknesses of the project before it is budgeted. By doing the feasibility study companies can save money due to low defects on the system. Below are some feasibility studies,

1. Economic Feasibility: Economic feasibility test is a method of getting the financial data or it is also called cost/benefit analysis methodology. It also results whether the software is in the budget or not.

Likewise, this system would be economically feasible because we do have budget which was provided by client and we would be finishing the project at its deadline so we won’t get problem economically.

1. Legal Feasibility: This is the method which watches out for the legal issues for the software. It keep tracks of the rules and regulations which the government has provided for the development and also sees weather the system has maintained its legal boundaries or not.

For, Legal feasible we have already hired a lawyer who would be looking after the rules and legal things which would help out management system to be legally feasible.

1. Technical Feasibility: This processes sees that if the company has the technical resources like hardware of the developers or not. This process also evaluate those technical resources.

To make this system we already have gadgets which are required and we don’t require to buy any other extra gadgets so technically we won’t be getting and problems.

1. Social Feasibility: This study shows the social impact towards the society by the software. It helps to analysis whether the system is feasible for the society or not. It also make sure that none of the people of the society are harmed by the system.

As, we are making a system for the society we have taken all the people’s opinions and reviews regarding the system so it would be social feasible.

1. Political Feasibility: In this process we can analysis the negative impact which can be occur by different political parties in the development phrase or afterwards.

Similarly, we would be preventing any promotions of parties or demotion of parties so that we won’t get any problems from that. So, this system is politically feasible.

# Software Requirement Specification (SRS)

SRS also known as Software requirement specification which describes the behaviour of software. SRS fully describes what the system does and how are the expected functionalities. SRS minimize the time and effort taken by the developers for the particular project.

Below down are the types of requirements required for development process,

1. Functional Requirement

Functional requirement shows the documents and activities which a system must do. All the functionalities must be cleared before the software deployment process. So, I have used different information gathering methodologies to collect the requirement accordingly the point of view of individuals. Below down are the functional requirement:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Title** | **Description** | **Rational** | **Dependency** |
| F1 | Sign Up | User can create their account | For the propose of creating a new account for further use of system | N/A |
| F2 | Login | User can be able to login. | For the propose of using the system overall. | F1 |
| F3 | View Train details | Admin and users can view the details of the train. | To view the all the details like cost, arrival, departure etc. | F1, F2 |
| F4 | Add personal record | User should be Users can add their records (details). | To add personal details. | F2 |
| F5 | Update personal record | User can be able to update their records. | To edit or change the personal existing record. | F4 |
| F6 | Delete personal record | User can delete their records from the system. | To delete the records which are not needed. | F4 |
| F7 | Add train details | Admin should be able to add the train details. | To add the schedules of the train and the availability of tickets. | F1, F2 |
| F8 | Delete train details | Admin can delete the train schedules. | To delete the train schedules. | F7 |
| F9 | Update train details | Admin can update the details of the train. | To update the train details like arrival time, departure time, bookings etc. | F7 |
| F10 | Ticket Booking | User can book tickets. | To book train ticket. | F3 |
| F11 | View ticket order | Admin must be able to view the order details. | To view the ticket order details. | F10 |
| F12 | Cancel ticket | After booking the ticket user should also be able to cancel the ticket. | To cancel ticket order. | F10 |
| F13 | Send and receive feedback | User should be able to send their feedback about the system and both admin and user can view the message. | To send and receive feedback. | F1, F2 |
| F14 | Sending notification. | Notification is a means of communication. Admin should be able to send any notice or news to the user about the system. | To send notification to the user. | F2 |
| F15 | Check order history | User should be able to view their train order history. | To check their train reservation history. | F2, F10 |

1. Non-functional requirements

Non-functional requirements show how a system should behave and keep its limitations of its functionality. Non- functional requirements is also known as system quality attribute’s. Below down are some non-functional requirements,

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Description** |
| NF1 | Security | The system should be secure only the authorized users should be able to use the system. |
| NF2 | User Friendly | System should be user friendly so the user could get great experience using the system and this may result in satisfaction of user. |
| NF3 | Performance | Good performance is necessary in order to get the output in real-time speed. There should not be any kind of delay while running the system. |
| NF4 | Reliability | It should be trust worthy so that each and every users may feel secure while using it. |
| NF5 | Maintainability | To maintain the system is the most important factor because errors may occur at any time so the system must be maintainable. |
| NF6 | Scalability | The data store in the system will increase in course of time so, the system must be always ready to change or add new features to manage the data. So the system must have scalability as it handles the amount of growing data in the system. |
| NF7 | Usability | Everyone from each and every background should be able to use this system. |
| NF8 | Serviceability(Supportability) | The system should provide support to handles the errors and the work of the user. It must give good services to the user to improve the business and user performance. |
| NF9 | Availability | The system should be available from anywhere at any time so the users may get full service of the system. |

1. MoSCoW Prioritization

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Requirement** | **MoSCoW** | **Rational** |
| F1 | Sign Up | Must have | For the propose of creating a new account for further use of system |
| F2 | Login | Must have | For the propose of using the system overally. |
| F3 | View Train details | Must have | To view the all the details like cost, arrival, departure etc. |
| F4 | Add personal record | Must have | To add personal details. |
| F5 | Update personal record | Should have | To edit or change the personal existing record. |
| F6 | Delete personal record | Should have | To delete the records which are not needed. |
| F7 | Add train details | Must have | To add the schedules of the train and the availability of tickets. |
| F8 | Delete train details | Should have | To delete the train schedules. |
| F9 | Update train details | Should have | To update the train details like arrival time, departure time, bookings etc. |
| F10 | Ticket Booking | Must have | To book train ticket. |
| F11 | View ticket order | Must have | To view the ticket order details. |
| F12 | Cancel ticket | Must have | To cancel ticket order. |
| F13 | Send and receive feedback | Could have | To send and receive feedback. |
| F14 | Sending notification | Could have | To send notification to the user. |
| F15 | Check order history | Could have | To check their train reservation history. |
| NF1 | Security | Must have | The system should be secure only the authorized users should be able to use the system. |
| NF2 | User Friendly | Should have | System should be user friendly so the user could get great experience using the system and this may result in satisfaction of user. |
| NF3 | Performance | Must have | Good performance is necessary in order to get the output in real-time speed. There should not be any kind of delay while running the system. |
| NF4 | Reliability | Must have | It should be trust worthy so that each and every users may feel secure while using it. |
| NF5 | Maintainability | Must have | To maintain the system is the most important factor because errors may occur at any time so the system must be maintainable. |
| NF6 | Scalability | Should have | The data store in the system will increase in course of time so, the system must be always ready to change or add new features to manage the data. So the system must have scalability as it handles the amount of growing data in the system. |
| NF7 | Usability | Must have | Everyone from each and every background should be able to use this system. |
| NF7 | Serviceability(Supportability) | Must have | The system should provide support to handles the errors and the work of the user. It must give good services to the user to improve the business and user performance. |
| NF8 | Availability | Must have | The system should be available from anywhere at any time so the users may get full service of the system. |

1. Hardware requirements

Processor: Intel® Core™ i5-5200 CPU @2.20 GHZ

RAM: 4GB above

Monitor: VGA colour monitor

1. Software requirements

Operating System: Windows 7 onwards

Developing tool: PHP my admin, notepad++

Database: PHP MY ADMIN

# Use case Diagram

Use case diagram is figure which shows the different way of using the system by the users. To build use case we would be need some special symbols like connectors, symbols etc. It also helps to collect the functionalities requirements before making the system. Below down is the use case diagram of the system,

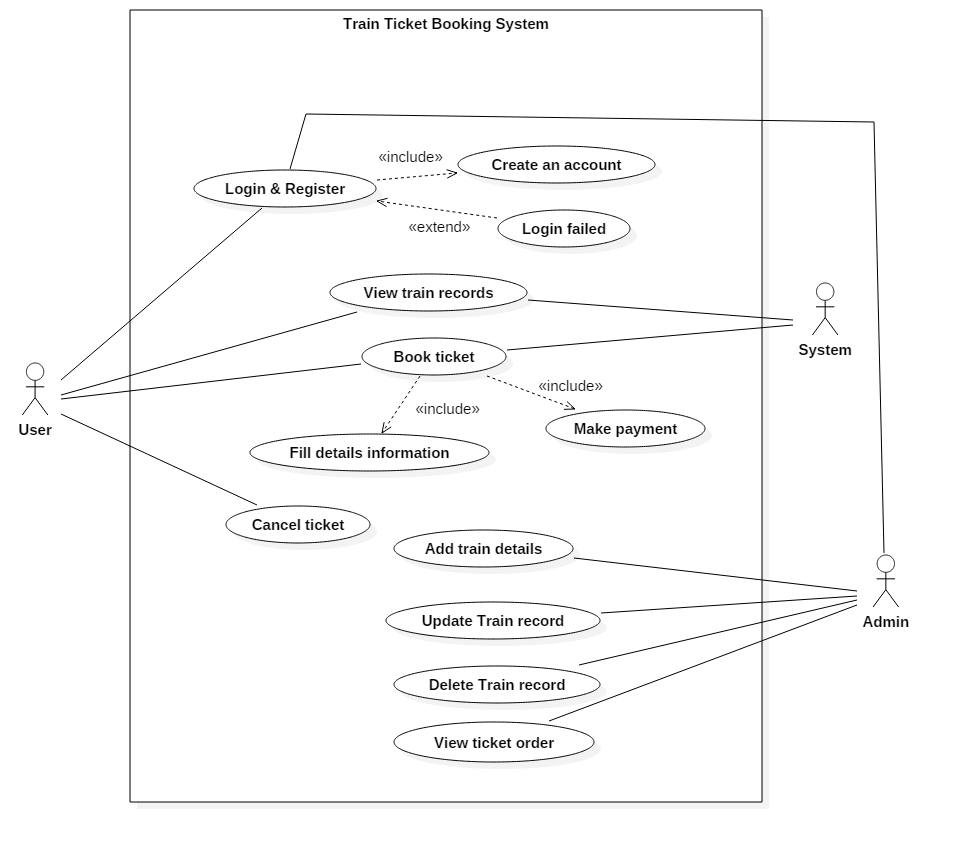


Figure 3: Use case diagram

As in the above diagram we can see there are some primary actors which are user, system and admin their work is to interact with the system. In the given figure admin are those who have the full control of the system they have authority to edit the information. Similarly, users are the one who are also known as customer or clients they have to register themselves in order to login inside the system and once they are logged in they can book tickets or make payments.

Login and register: These features are used by admin as well as user admin uses it to make any changes in the system and the user uses it to make booking and payment secure.

View train records: This functionality is used by the users to see the train’s status its price so that they can book a ticket on their feasible time.

Book ticket: The users can book the tickets after looking the feasible time period of booking and while booking they can see the train time period and they can pay.

Cancel Ticket: This feature is for the users to cancel their bookings if it’s not required any more.

Add Train details: This is the feature which the admin uses to upload the timings of arrival, ticket price and seat details of the train for the customers.

Update train records: This feature is used by the admin to change any details regarding the train.

Delete train record: This feature is used by admin to delete any train status if something went wrong in the train or in the station.

# System architecture

System architecture that I will use in this system will be 3-tier because this is the project of client to server system so the user don’t have to use the database due to this reason I am using this method. 3-tier architecture also improves the horizontal stability, performance and availability. Below down are the three tiers.

1) Presentation tier: It is a tier which is built with HTML, CSS and JavaScript and is deployed to the computer by web browsers.

2) Application tier: Application tier also known as logic tier is written in java which contains the business logic which supports the application core functionalities.

3) Data tier: It consist of database and programs for managing read and write access to a database. The database system for managing read/write access are MYSQL, Microsoft SQL Server and MangoDB.

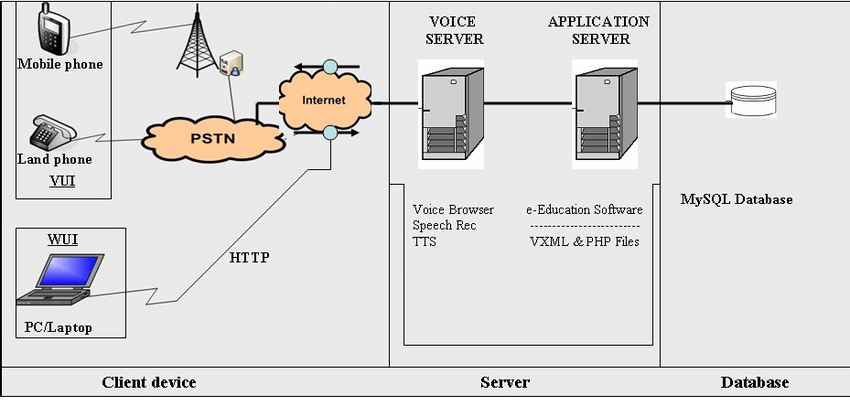


Figure 4: System Architecture

## Scenario

Train ticket booking system is a system making the reservation process much easier for the customers. The system must include the registration, login from which passengers can make an account and book their ticket, edit or delete there reservation through online. They can also view train record by logging in to the system. This system also include the online payment methodology which makes the payment easier. This system should contain the admin panel which will be consisting the features like managing bookings, add, update and delete the status of the tickets, changing the time of arrival of train.

## NLA

NLA also known as natural language analysis is a process of generating verb, noun and adjectives from the given scenario and afterwards changed it to classes, attributes and methods. For this system we have used to remove the duplications, irrelevancy.

* Nouns

|  |  |  |  |
| --- | --- | --- | --- |
| Nouns | | | |
| Train | Ticket | System | System |
| Reservation | Process | Costumer | Passenger |
| Registration | Login | Customer | Account |
| Book | Ticket | Reservation | Online |
| System | Payment | System | Admin |
| Panel | Features | Bookings | Status |
| Tickets | Time | Arrival | Train record |

|  |  |  |  |
| --- | --- | --- | --- |
| Unique nouns | | | |
| Train | Ticket | System | Reservation |
| Process | Customer | Registration | Login |
| Account | Booking | Online | Admin |
| Payment | Panel | Features | Train record |
| Time | Arrival | Passenger |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Irrelevant classes removed | | | |
| Customer | Ticket | System | Reservation |
| Passenger | Customer | Train record |  |
| Payment | Booking | Admin |  |

|  |  |
| --- | --- |
| Synonyms | |
| * Passenger * Customers | * Booking * Reservation |

|  |  |  |
| --- | --- | --- |
| **Noun identification for candidate class** | **Selected as candidate class** | **Justification for selection or rejection as candidate class** |
| Admin | Yes | It play significant role in the system. |
| Passenger | Yes | It play significant role in the system. |
| Payment | Yes | It specifies requirements. |
| Train records | Yes | It provides all the details information regarding train. |
| Ticket | Yes | It play significant role in the system. |
| System | Yes | Requirements specifies and act as main class in the system. |
| Booking | No | It is the method. |
| Customer | No | Alternative to passengers. |
| Train | No | Irrelevant to the requirements and it is also the property of flight list. |
| Time | No | Irrelevant to the requirements. |

|  |
| --- |
| Selected Candidate class |
| * Passengers * Ticket * Admin * System * Train records * Payment |

* Verbs

|  |  |  |  |
| --- | --- | --- | --- |
| Verbs | | | |
| Booking | Update | Include | Logging |
| Can | Make | Update | Delete |
| Include | Make | Should | Contain |
| Will | View | Consisting | Managing |
| Add | Edit | Delete | Changing |

|  |  |  |  |
| --- | --- | --- | --- |
| Unique Verbs | | | |
| Booking | Update | Include | Logging |
| Can | Make | Edit | Delete |
| Should | Contain | Will | View |
| Consisting | Managing | Add | Edit |
| Delete | Changing |  |  |

|  |  |  |
| --- | --- | --- |
| Irrelevant classes removed | | |
| Booking | Changing | Managing |
| Delete | Edit | View |
| Update | Add | Logging |

|  |
| --- |
| Synonyms |
| * Update * Edit * Changing * Managing |

|  |
| --- |
| Selected Method |
| * Add * Update * Delete * Logging * Booking * View |

## Class Diagram

Class diagram is static structure design which shows the structure of the classes by giving out the system’s classes, their attributes, methods and relation among objects. It is helpful for business analyst to model the system for business purposes. Class diagram is also said to be the most important diagrams in UML which shows the whole system structures and components. While, this project is also been developed by being reliable on our class diagram.

Accordingly the functions of each classes I have made abstract classes as well as interface.

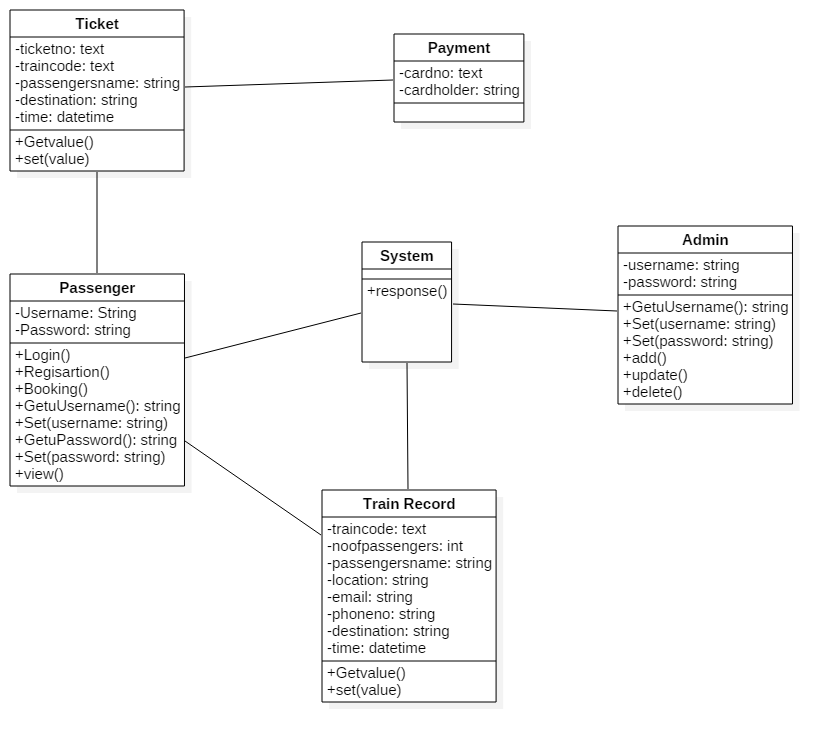


Figure 5: Class Diagram