# Analysis

## Introduction to Analysis

Analysis is a systematic evaluation where the informative subject is break down to a small chunk for a better understanding which aids in decision making and solving problems. Analysis is detailed investigation about certain topic to find out its nature or to find out its necessary features. In this phase meeting with client and project holders and discussing about the project is done. It falls under Second stage of Software Development Lifecycle (SDLC) Gathering requirement for project is the main goal of this stage. If the requirements of project are not studied properly there might be failure of project. There are different types of analysis like as SWOT Analysis, PEST, CATWOE, etc.

## Analysis Methodology

There are various methodology which are used to plan, structure and control the development process of a system. Some of them are Waterfall, DSDM (i.e. Dynamic System Development Model), Agile, object-oriented, etc. Among them, I have chosen to follow the *Waterfall methodology* i.e. ‘***Hard System Methodology’*** which focuses on technical aspectfor this project. Unlike other methodologies, this one is the oldest and is easiest to understand. It is also known as a linear-sequential life cycle model. The concept of this methodology is very simple i.e. each phase must be completed before beginning the next phase. Following this concept, first all of the system requirements analyzed and then after its completion user requirements are analyzed and so on.

The waterfall methodology is the traditional model for software development. It is the most relevant methodology for my project as it is well planned and each phase must be completed before the going to next phase and there is no overlapping in the phases which is the principle of waterfall model.

Some of the reasons of why I choose this methodology are as follows:

* Simple and easy to use and understand even for beginners.
* Thorough planning and scheduling is ensured.
* Simple enough to handle as model is rigid.
* Easy to keep track of the task due to clearly defined phases.
* Provides easy analysis and testing.

Since a hard approach is used for analysis, DFD (Data Flow Diagram) of the proposed system was created and is shown as below:

## Feasibility Study

It is a study that specializes on identifying, describing and evaluating a proposed system and then measuring how beneficial the development of the system would be to an organization. Hence, before starting work on this project, a feasibility study of this project is a must to ensure that this project will be beneficial rather than just a waste of time. The final report of this study displays the nature and scope of the proposed system.

The following feasibility studies were carried out for this project:

1. **Technical feasibility:** Here, availability of the technology and technical skills required for the proposed system are determined. Since, the required technology i.e. web server, server software, database system, domain names, networking, etc. alongside staff with corresponding skills are available, this project can be concluded as technically feasible.
2. **Economic feasibility:** This feasibility study can also be called as cost-benefit analysis. Financial benefits of the system is compared against the cost associated with development of system. The project is economically feasible as the benefit is greater than cost.
3. **Legal feasibility:** For a project to be legally feasible, one must follow all rules specified, should not involve and encourage crime and not have any copyright issues. Likewise, this project meets all the criteria mentioned above so it is legally feasible.
4. **Social feasibility:** In this feasibility study, we determine if our project is accepted by the public. Currently, there is lack of online vehicle rental services in our country so, people will be pleased if we launch one of these services. In a nutshell, the project is socially feasible
5. **Operational feasibility:** This study in particular addresses the performance and reliability of the proposed system. The system must be able to provide right information at right time and right place. This very project promises to deliver these requirements so it can be considered operationally feasible

## System Requirement Specification (SRS):

A System Requirement Specification (SRS) is a report/document that defines all the data, behavioral and functional requirements of a system that is being developed. Not only it focuses on the behavior of the system, but also defines the main business processes that are going to be supported at a high level. Following it, the document also defines the key performance parameters that need to be met by the system. Moreover it allows clients and project contractors to agree on the functions of the system. Likewise, estimation of product cost, risks and scheduling are also aided by this specification.

The main components of this requirement specification are functional requirements, non-functional requirements and hardware and software requirements respectively. Each are discussed in detail below:

## Functional Requirements:

These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. It specifies the application functionality that the developers must build into the product to enable users to accomplish their tasks.

Following are the functional requirements of the proposed system i.e. Motors International:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.N | Req. Name | Data | Description | Rational | Dependencies |
| Fr1 | User Registration | User Details | User should be register in order to buy/sell vehicle | To register new user as buyer or seller | N/A |
| Fr2 | Login | Email, Password | Valid user is required to login into the website | Authentication of valid email and password | Fr1 |
| Fr3 | Add vehicle | Vehicle Details | New vehicles of category car and bike can be added | For the addition of new vehicle | Fr2 |
| Fr4 | Edit Vehicle | Vehicle Details | Registered vehicle details can be edited | To update vehicle’s Information | Fr3 |
| Fr5 | Delete Vehicle | Vehicle Details | Admin can delete a vehicle. | To delete vehicle Information. | Fr3 |
| Fr6 | Display Vehicles | Vehicle Information | Both registered and un-registered customer should see the vehicles and its details. | To display all available vehicles for sell. | Fr3 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| Fr7 | Search Vehicle | Vehicle name | Customer should be able to search vehicles | To search vehicle | Fr6 |
| Fr8 | Buy Vehicle | Vehicle Details | Register customer are allowed to purchase vehicle | To buy vehicle | Fr6 |
| Fr9 | Sell Vehicle | Vehicle Details | User are allowed to sell vehicle | To sell new/old vehicle | Fr3 |
| Fr10 | Registration date for vehicle service | Available date information | Admin should add available date for servicing | To add available date |  |
| Fr11 | Book for vehicle servicing | Service date | User should book date for vehicle service | To service vehicle | Fr10 |
| Fr12 | View booked registration | User and time details | View date & time booked by user | To view registration date | Fr10,Fr11 |
| Fr13 | Cancel servicing registration | Booked details | User are allowed to cancel vehicle service registration | To cancel vehicle service registration | Fr11 |
| Fr14 | Book Vehicle | Vehicle details | User are allowed to book upcoming vehicle before certain time of its release | To book vehicle | Fr6 |
| Fr15 | Update profile | Customer Details | Registered customers can update their profile | To successfully update user profile | Fr1 |
| Fr16 | Review on Vehicle | Vehicle Details | User are allowed to provide review on vehicle |  | F3 |
| Fr17 | Change Password | Email and previous password | Registered customer can change the password | To provide secure services to customer | Fr1 |
| Fr18 | Online Payment | Customer Details and Payment Details | Registered customer can pay to buy vehicle and its accessories | To provide easy payment service | Fr11,Fr8 |
| Fr19 | Rent Vehicle | Vehicle Details | Bike and Car can be rented by booking. | To rent vehicles |  |

## Non-functional Requirements

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may define constraints on the system implementation such as the capabilities of I/O devices or the data representations used in interfaces with other systems. Non-functional requirements, such as performance, security, or availability, usually specify or constrain characteristics of the system as a whole.

Following are the non-functional requirements of the proposed system i.e. Motors International.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement. ID | R. Name | Rational | Description | Dependency |
| NFR1 | Usability | To gain more users | The system should satisfy the end-users and must be accepted by them | NRF2 |
| NRF2 | Security | To control unauthorized access and maintain confidentiality of data and integrity. | The system provides username and password to prevent the system from unauthorized access. |  |
| NRF3 | Performance | To run the system effectively and efficiently | The system should meet the defined quantitative metric | NRF7 |
| NRF4 | Reliability | Eliminate the problems | System should be reliable that every part works smoothly | NRF2 |
| NRF5 | Availability | To maintain availability | The system should always be available for access 24/7 | NRF4,NRF8 |
| NRF6 | Maintainability | To keep the system running by accomplishing new requirements | scheduled patches and updates to improve system’s features and functionalities | NRF8 |
| NRF7 | Scalability | To make system flexible | The system should be able to withstand the increasing number of data and functionalities over time. | NRF4 |
| NRF8 | Recoverability | To ensure all time availability of data. | The system should must have a backup service in case of any system failures. | N/A |
| NRF9 | Accuracy | To maintain accuracy of system | The system should be able to provide accurate result. | NRF2,NRF6,NRF7 |
| NRF10 | Error handling | To recover from an error | The system error must be considerably minimized and recover through appropriate error messages. | NRF1 |

## Moscow Prioritization

Typically, it is a process of arranging items or activities in an order of importance relative to each other. For this project, we will be using MoSCoW technique. It is one of the easiest method to understand for the stakeholders as it rates the requirements as Must, Should, Could or Won’t. It is also better than the numeric rating system. Specifically, the use of MoSCoW provides a clear indication of that requirement and the expectations for its completion.

The four categories of ***MoSCoW*** are discussed below:

* **Must have:** This category contains Minimum Usable SubseT (MUST) of requirements which the project guarantees to deliver.
* **Should have:** The requirements that are important but not that vital are placed in this category. Even if these are left out, there is still a solution available.
* **Could have:** Over here are the desirable or wanted requirements but are not so much important. The impact of leaving them is usually low.
* **Won’t have:** Likewise, this category contain the requirements which will be not delivered this time but can be added later.

The following table represents a MoSCoW prioritization of functional and non-functional requirements of this project:

1. ***Prioritization of Functional requirements:***

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional requirements** | **MoSCoW** |
| FR1 | User registration | Must have |
| FR2 | Login | Must have |
| FR3 | Add vehicles | Must have |
| FR4 | Edit vehicles | Should have |
| FR5 | Delete vehicles | Should have |
| FR6 | Display vehicles | Must have |
| FR7 | Search vehicles | Should have |
| FR8 | Buy vehicles | Must have |
| FR9 | Sell Vehicle | Must have |
| FR10 | Registration date for vehicle service | Must have |
| FR11 | Book for vehicle servicing | Should have |
| FR12 | View booked registration | Must Have |
| FR13 | Cancel servicing registration | Should have |
| FR14 | Book Vehicle | Should have |
| FR15 | Update profile | Should have |
| FR16 | Review on vehicles | Should have |
| FR17 | Change password | Must have |
| FR18 | Online payment | Could have |
| FR19 | Rent Vehicle | Could have |

1. ***Prioritization of Non-functional requirements:***

|  |  |  |
| --- | --- | --- |
| **ID** | **Non-functional requirements** | **MoSCoW** |
| NFR1 | Usability | Must have |
| NFR2 | Security | Must have |
| NFR3 | Performance | Must have |
| NFR4 | Reliability | Should have |
| NFR5 | Availability | Must have |
| NFR6 | Maintainability | Won’t have |
| NFR7 | Scalability | Could have |
| NFR8 | Recoverability | Should have |
| NFR9 | Accuracy | Must have |
| NFR10 | Error handling | Should have |

## Hardware Software Specification

The effective operation of system completely relies in its hardware and software components. Hardware requirements defines physical computer resources and software requirements defines the software resource requirements and prerequisites that are essential for optimal functioning of the system/application.

The following are the respective hardware and software requirements of the proposed system:

* **Hardware requirements:**

Processor: Celeron 500MHZ or any PENTIUM processor.

RAM: Minimum 1GB but recommended 2GB for smooth experience.

Hard disk capacity: Minimum 80GB**.**

* **Software requirements:**

Operating system: Windows XP, 7, 8, 10, macOS, Android

Frontend: Any web browser.

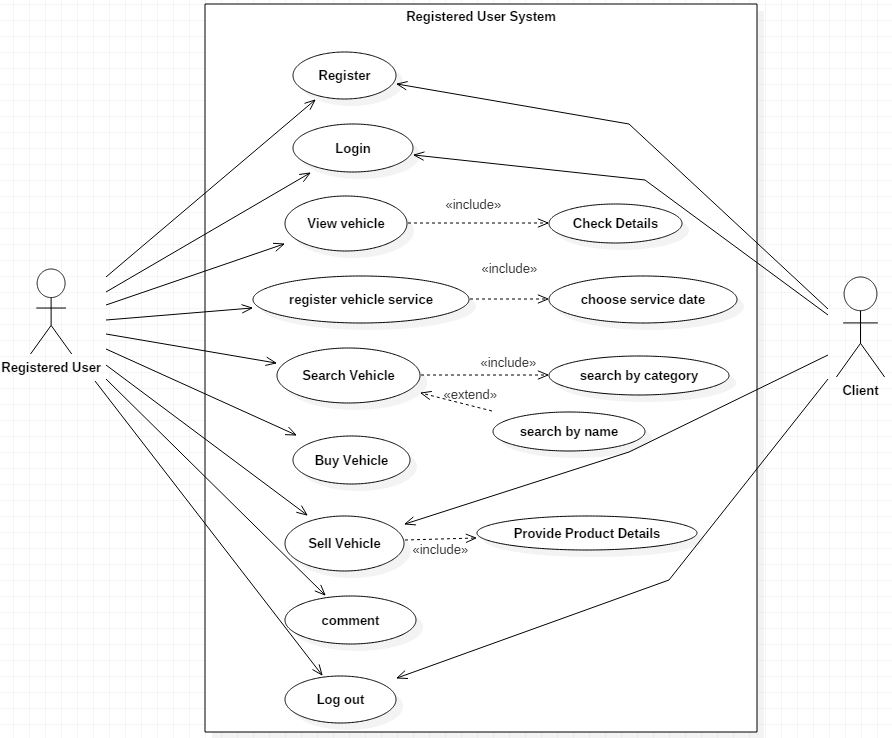
Database: MySQL

## Use Case Diagram

Use-case diagrams fall under the UML (Unified Modelling Language) diagrams. A use case diagram in general is a graphical representation of the interactions among the elements of a system. Its use in analysis is to identify, clarify and organize system requirements. There is a total of four main components in a use-case diagram and they are as follows:

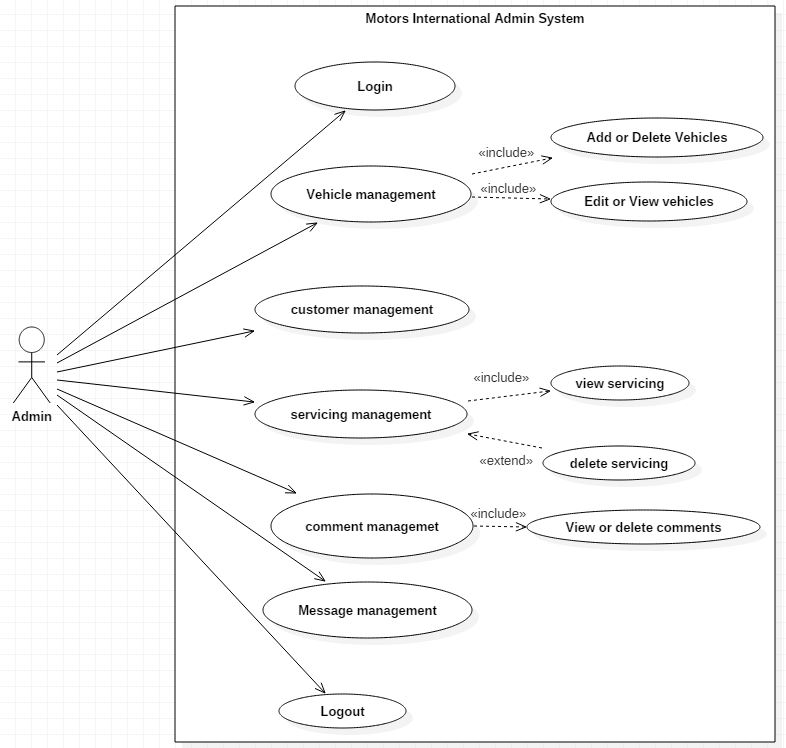
* **Boundary:** This component defines the system of interest in relation to the world around it.
* **Actors:** As the name suggests, it refers to the individuals having a specific role in the system which means they are involved with the system.
* **Use-cases:** These are defined as the specific roles played by the actors within and around the system.
* **Relationships:** Here, the relationships between and among the use-cases and actors are discussed.

Use case diagram for the proposed system is as follows:



*Fig: User/Customer use case diagram*

|  |  |
| --- | --- |
| **Name of Use case** | **User System** |
| Actor | Users |
| Alternate Actor | Clients |
| Primary Flow | 1. User enter current password. 2. Checks the current password is valid or not. 3. If correct then enter new password. 4. Encryption of new password is done. |
| Alternative Flow | N/A |
| Justification | This use case shows the overall features of customer and client on buying/selling, search, book product and register for vehicle service, etc. |



*Fig: Admin use case diagram*

|  |  |
| --- | --- |
| **Name of Use case** | **Admin System** |
| Actor | Admin |
| Alternate Actor | N/A |
| Primary Flow | 1. User enter current password. 2. Checks the current password is valid or not. 3. If correct then enter new password. 4. Encryption of new password is done. |
| Alternative Flow | N/A |
| Justification | This use cases shows the ability of admin to manage the system. |

## NLA (Natural Language Analysis)

Let’s start with the introduction of **NLA (i.e. Natural Language Analysis).** In straightforward language, it is a procedure/technique that guides while recognizing classes which further need to be included into class diagram. By using it, we can conclude candidate class alongside their relationships and attributes. Following this method, we search and point out nouns, adjectives and verbs from the given scenario. Here, ***nouns*** identifies with potential ***classes***, ***adjectives*** as potential ***attributes*** and **verbs** identify with potential ***functionality***.

Scenario

The main purpose of Motors International is that it provides provision to customers to buy or book vehicles and its accessories through online. The current system is offline system, to purchase vehicle the customer should visit to showroom. So this current system is very difficult because its time consuming. So our project aims at creating an web application which tracks Customer records, Online booking, Online vehicle records, vehicle servicing, etc. and it provides easy to use web based interface for customers where customers can search for vehicles, view complete details, models, features, pricing of the vehicles and book the vehicles.

Now breaking down our scenario, the nouns, adjective and verbs from our situation are listed below:

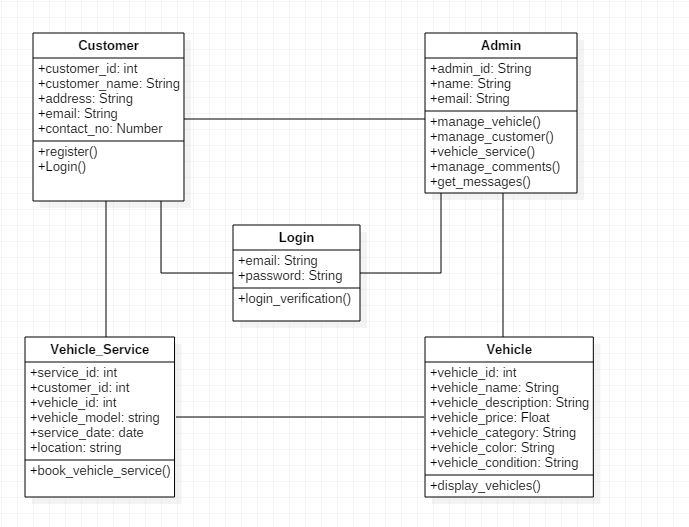
|  |  |  |
| --- | --- | --- |
| Nouns: | | |
| * Customer * Vehicle * Car * Bike * Forms * Vehicle Parts | * Users * Price * Admin * Application * Inventory * Record |  |

|  |
| --- |
| Verbs: |
| * Add * Show * Organize * Service * Register * Book * Buy * Remove * Sell * Track * Edit |

|  |  |
| --- | --- |
| Adjectives: |  |
| * Type * Size * Category * Permit * Interactive * Detail | * Model * Company * Color * Accessories |

Initial Class Diagram

The following diagram represents the initial class diagram of our system based on the NLA process mentioned earlier.

****

*Fig: Initial class diagram*