# Chapter 2: Analysis

## 

## 2.1 Introduction to Analysis:

Analysis is the first step in the phases of software development life cycle (SDLC) in which requirement of the clients are gathered and also focuses on the solution of the problem of the client. Requirement Specification is the main goal of analysis. Requirements of the project are collected with the help of user interaction.

In spite of the fact that requirement gathering is the main aim of this stage, analysis focuses on:

* Analysis focuses on the scope of the project.
* It focuses on understanding the problem domain and requirement.
* Some requirements need prioritization and alternatives. Analysis should focus into this aspect.
* Analysis should focus on the feasibility of the project.
* System Requirement Specification (SRS) should be prepared as it helps in functional and non-functional requirements of the project.
* Conceptual System Design should be prepared as it helps in Management Information System.

## 2.2 Analysis Methodology:

It requires a series of structure to develop a system. Analysis, Design, Implementation and Maintenance are the key stages of Software Development Life Cycle. Among these phases Analysis have its own methodologies to help gather requirements and functionalities to develop a system.

Here, for this project I have chosen **Soft Approach Methodology** among other various methodologies like **People Oriented, Process Oriented, Combined Approach, Hard Approach**.

**Soft Approach:**

Both Technical and Human aspects are main parts of Soft Approach to develop a system. Key Technique for soft approach is the interaction between users.

The consideration of human, social activity and technical functional requirements belongs into this approach.

Following are the number of steps undertaken while using this methodology:

1. **Human activity analysis of soft approach involves rich picture, root definition and conceptual model**

**Rich picture:**

The collection of pictures, symbols, text to show relationships, connections between requirement of the users are called rich picture. It is basically a rough sketch diagram to shows the relationships.

**Root definition:**

Root Definition clarifies the processes and problem of the system. There are two types of root-definition i.e.:

Primary task root-definition: focuses on system processes

Issued-based root definition: focuses on problem statement

There are three elements of root definition. i.e. **WHAT, HOW AND WHY**

WHAT: It defines the aim of the system.

HOW: It defines how to achieve the aim.

WHY: It defines why to use the system.

.

**CATWOE**

**CATWOE** stands for

**C Customers/Clients**

**A Actors / Agents**

**T Transformations**

**W World View**

**O Owners**

**E Environment**

**CATWOE** analysis helps to define a root definition

**Customers** are the most important factor for any business or project. They are the one who use system. So without considering the customers, the project will have no meaning. **Actors** are the defined employees, suppliers, agencies and officials. Changes are inevitable. The system must change according to time for the easiness of user, so **Transformation** is changes occurred by the system for the easiness of user. **World Wide** covers the issues related with wide reaching and long lasting effects or relatively contained and short life. **Owner** is the person who owns the problem they must be the parts of its solution or both. **Environment** deals with the budget, resources, laws and regulations of the project.

**Conceptual Diagram**

The conceptual diagram of Online Vehicle Tax Pay System represented as follows:

1. **Social-technical aspects of analysis and design**

Feasibility Study covers the Social and technical aspects of analysis and design of Online Vehicle Tax Pay System and is described in another portion.

1. **Human-Computer Interface design**

To remove duplication of data and diagram human-computer interface design of Online Vehicle Tax Pay System is used.

1. **Design of technical aspects**

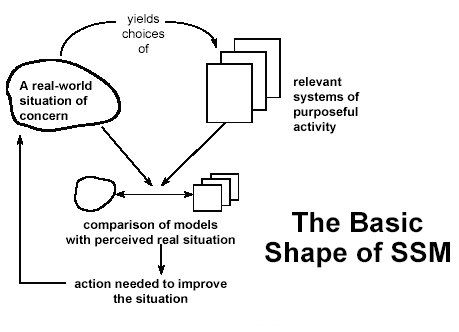
Technical aspects of Online Vehicle Tax Pay System cover System Requirement Specification (SRS). System Requirement Specification (SRS) portion of the project is describe in another part.

1. **Cultural aspects**

Cultural aspects represents the cultural beliefs and practices of the society. Beliefs of the society do not effect with the use of this application.

1. **Political aspects**

The application that we will develop does not effects the politics policies of the country. Before to develop this application we have to know the political policies of the country.

**Overview of the Soft System Methodology:**

Above screenshot represents the overview of the soft system methodology. The problem arises due to cultural and social aspects is manage with the use of this methodology. Human interaction is an important factor to control the problem. In addition, human interaction managed the misunderstanding between developer and users. For sustainable running this application and to build trust between users and developer, human interaction plays important role.

**Advantages of Soft Approach over other Methodology and its areas of focus:**

This methodology considers both human and technical factors of the system. This is the main reason I have chosen this Methodology. Following are the reasons to choose soft approach over other approach are:

* It cover human, technical, cultural and political part of the system analysis.
* It increases the understanding level of analyst of a problem during investigation as well as in later stages.
* To solve the problem successfully it relies on the holistic view.
* It doesn’t need any equipment to rely on. Just pen and paper is enough.
* It involves the participation of user during analysis and design phase.

## 2.3 Feasibility Study

The real world scenario in which a proposed system is financially, technically and operationally practical and workable called feasibility study. Feasibility study comes under social and technical aspects of analysis and design and covers description of product, financial data, legal requirements, tax obligation and many more.

Types of feasibility study and their relation with Online Vehicle Tax Pay System are:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N** | **Types of Feasibility Study** | **Questions (?)** | **Relation with the project** |
| 1 | Operational Feasibility | How this project helps to solve the problems and catch advantages and opportunities? | This project provides oppurtunities and advantages to both the users and developer because in context of our country, people need to stand in queue for hours to pay their vehicle Tax. With this software they don’t need to be in queue and can pay their tax from home or work. |
| 2 | Technical Feasibility | Do you have the technological resources to undertake the project? Are the processes and procedures conducive to project success? | I have sufficient technological knowledge and skills to develop this project. |
| 3 | Economic Feasibility | Given the financial resources of the company, is the project something that can be completed? | This project used the online payment system. I am using 3rd party application to process this. |
| 4 | Schedule Feasibility | Can the project be completed within the time period? | I have already scheduled the project and made gantt chart and work breakdown structure for this project. |
| 5 | Marketing Feasibility | Is the project useful for everyone and of different aged group? | For those who own vehicle, this project is highly feasible. |
| 6 | Cultural Feasibility | Is this project acceptable by all local people? | This is a web based project and doesn’t harm anyone culture so this is feasible. |

# 2.4 Requirement Analysis

Requirement analysis is the process of determining the user expectations for the project. These features must be quantifiable, relevant or detailed. The Requirement Analysis of the project includes functional and non-functional requirement that need to be implemented in the system.

# 2.4.1 Functional Requirements

Functional Requirement are the services or functions of the system that must be implemented into the system. These are the main function of the system that must be included to run the system smoothly. Not having the functional requirement in the system may lead the system to failure.

Functional requirements of Online Vehicle Tax Pay System described as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Rational** | **Dependencies** |
| FR1 | Registration | To login into system new user must register into system. | To create new user. | N/A |
| FR2 | Login | To get access into the system. | Only Registered user can login into the system. | FR1 |
| FR3 | Add Detail | Logged user need to add their personal detail. | To add personal Detail of user. | FR2 |
| FR4 | Email Verification | First Time Logged user verifies the account. | To verify the account. | FR2 |
| FR5 | Edit Profile | Logged user can edit their personal profile. | To edit the profile of the user. | FR2,FR3 |
| FR6 | Add Vehicle Detail | Logged user can add their vehicle information. | To add vehicle detail. | FR2,FR4 |
| FR7 | Edit vehicle Detail | Logged user can edit their vehicle detail. | To edit the detail of the vehicle. | FR2,FR5,FR6 |
| FR8 | Track Tax System | Logged user can track the amount they need to pay. | To know the amount of tax to pay. | FR2,FR5 |
| FR9 | Select vehicle Type | Logged user selects the vehicle type from the category they inserted before. | To select the type of vehicle. | FR2,FR6 |
| FR10 | Select Insurance Type | Logged User can select the type of insurance they want. | To select the insurance. | FR2,FR5 |
| FR11 | Payment Mode Selection | Logged user can select the payment method. | To select the payment method. | FR2,FR6 |
| FR12 | Delivery Date Selection | Logged user can enter their date they want their bill book to deliver. | To select the delivery date. | FR2 |
| FR13 | Delivery Address Selection | Logged user can enter the address they want their billbook to be delivered. | To get the delivery address. | FR2 |
| FR14 | View Reports | Logged user can view the report of their task. | To view record. | FR2 |
| FR15 | Change Password | Logged user can change password. | For security purposes. | FR2 |
| FR16 | Rating | Rating describes the quality, merit, or amount of a system. | To show the feedback of user. | FR2 |
| FR17 | Logout | Logged user can out from system according to their needs. | To get out from the system. | FR2 |

## 2.4.2 Non-functional Requirements

Non-functional requirements define the system properties and constraints. Non-functional requirements are difficult to test in comparison to functional requirements. . It describes “how” the software will do it. It related to system as a whole. Failure to meet non-functional requirements may make the system unusable.

Non-Functional requirements of Online Vehicle Tax Pay System described as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Rational** | **Dependencies** |
| NFR1 | Authentication | Authorized user only access to system and unauthorized user cannot be access to system. | To maintain network secure by permitting authorized user only. | FR1,FR2 |
| NFR2 | Availability | When the clients need to utilize the system, the system should be accessible and perform the task they need. | To maintain client satisfaction and conveying the requirements in their accommodation. | N/A |
| NFR3 | Documentation | Documentation describes the overall concept of the system i.e. concept, guidance, how to use the system. | Provide learning material to user about how to use the system. | N/A |
| NFR4 | Usability | Design of system should be attractive and easy to use. | To maintain user satisfaction by using this system. | N/A |
| NFR5 | Maintainability | Errors and bugs are easily maintain while using the system. | To maintain sustainable running and to improve the application. | N/A |
| NFR6 | Legal | The application that I have made is legal in the context of law. | To prevent from cybercrime and laws. | N/A |
| NFR7 | User Friendly | The features of the system should be easily understand by the user. | Easy to use and understandable. | N/A |
| NFR8 | Scalability | Scalability includes the RAM, ROM, Database design of system. | To meet user requirements according to the user needs. | N/A |
| NFR9 | Performance | The system that I have made has smooth performance. | To maintain smooth user experience and efficient. | N/A |
| NFR10 | Reliability | Reliability of system includes good performance of hardware, software and firmware. | To deal with misunderstandings and problem of user and to maintain trust. | N/A |
| NRF11 | Confidentiality | Confidentiality defines the privacy of the system. To ensuring confidentiality, data encryption is required. | To control unauthorized access to system. | N/A |
| NRF12 | Integrity | Integrity refers the data that we enter into the system is real and accurate. | To maintain consistency, accuracy and trustworthiness of data. | N/A |
| NRF13 | Security | Here security of user data and security of system occurred. | To maintain security of the user data. | N/A |

# 2.4.3 Hardware/Software Specification

The application that I have been developing requires following hardware/software specification:

|  |  |
| --- | --- |
| Hardware | Software |
| * RAM: 4GB * Processor: Core i3 and above * Hard disk space: 100GB * Good internet connection | * Operating System: Windows 7,8 ,10, Linux * Database: SQL Lite |

## 

## 2.4.4 MoSCoW Prioritization

I have chosen MoSCoW requirement prioritization to prioritize the functional and non-functional requirements of Online Vehicle Tax Pay System among different requirement prioritization like grouping assignment, ranking, bubble sort technique, Analytic Hierarchy Process, hundred dollar method.

Due to the time and budget stakeholders does not meet their requirements. Prioritization of requirement occurs to satisfy stakeholders. MoSCoW stands for Most have (M), Should have (S), Could have (C) and Would have (W).

**Must have (M)** - Vital requirements for the project i.e. system cannot perform without those requirements.

**Should have (S)** - Requirements considered as important but not vital. If we have remaining time it can added in future.

**Could have (C)** – Requirements that considered as desirable but not necessary

**Would have (W)** - A requirement that will not be implemented now, but may be considered in future.

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Requirements** | **MoSCoW** |
| FR1 | Registration | Must have |
| FR2 | Login | Must have |
| FR3 | Add Detail | Must have |
| FR4 | Email verification | Must have |
| FR5 | Add Profile | Should have |
| FR6 | Add vehicle Detail | Should have |
| FR7 | Edit vehicle detail | Must have |
| FR8 | Track Tax | Must have |
| FR9 | Select vehicle type | Must have |
| FR10 | Select insurance type | Should have |
| FR11 | Payment Mode Selection | Should have |
| FR12 | Delivery date selection | Must have |
| FR13 | Delivery Address Selection | Should have |
| FR14 | View Report | Should have |
| FR15 | Change Password | Could have |
| FR16 | Rating | Could have |
| FR17 | Logout | Must have |

|  |  |  |
| --- | --- | --- |
| **ID** | **Non-functional Requirements** | **MoSCoW** |
| NFR1 | Authentication | Must have |
| NFR2 | Availability | Must have |
| NFR3 | Documentation | Should have |
| NFR4 | Usability | Must have |
| NFR5 | Maintainability | Should have |
| NFR6 | Legal | Should have |
| NFR7 | User Friendly | Must have |
| NFR8 | Scalability | Should have |
| NFR9 | Performance | Must have |
| NFR10 | Reliability | Must have |
| NRF11 | Confidentiality | Must have |
| NRF12 | Integrity | Must have |
| NRF13 | Security | Must have |

## 2.5 Use-Case Diagram

In UML, a use-case diagram is also known as dynamic or behavioral diagram. A use-case diagram shows the set of use-cases, actors (special kind of class) and their relationship in graphical way. To manage the system requirements i.e. identify, simplify and organize use-case diagram plays important role during analysis.

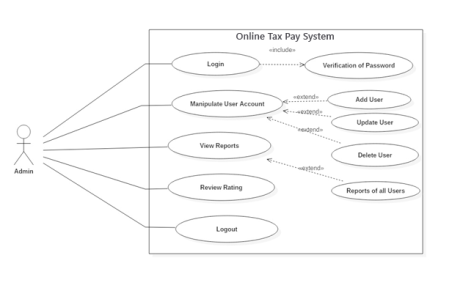
In object modeling and real-world system it has own standard notations. They are boundary, actors, use-cases and relationships.

**Advantages of Use-Case Diagram:**

* Functional requirements of the project gather in use case diagram.
* Use-case diagram gather additional behavior of system and improve in system robustness.
* It describes that how the actor connect with system.
* It can provide the summary of the whole system.
* During requirement gathering this diagram is good choice for system analyst because it is easy to draw and easily understandable.

**Disadvantages of Use-Case Diagram:**

* Use-case diagram cannot gather the non-functional requirements of the product.
* Some products and system have complexity to write and understand the diagram for both developers and users.
* It only summarized the relationship between cases, systems and actors.

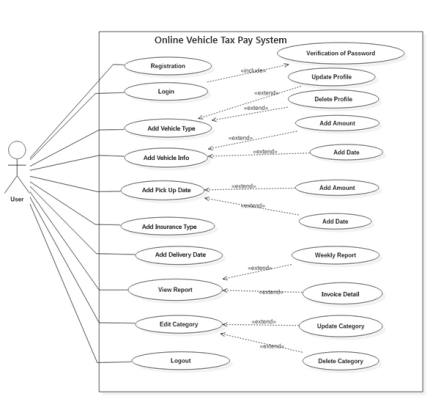


**Scenario Description**

Actor: Actor represents Admin

Admin can access to overall performance of the system. According to the diagram, Admin can perform following things:

* Admin can access to login into system of all user.
* Admin can access to manipulate user account. Manipulation of user account includes Add User, Update User, and Delete User etc.
* Admin can access to view reports, review and rating of all user.
* Admin can logout from system.



**Scenario Description**

Actor: Actor represents User.

User can access to following things of system:

* User can register to login into the system.
* User can login to use system.
* User can manipulate their own profile. Manipulation of the profile include Update Profile and Delete Profile.
* User can add their income and expenses.
* User can view their list of expenses.
* User can calculate their saving with the use of expenses and income.
* User can view their own report.
* User can edit their own category like food, clothes, hospital etc.
* User can logout from system.

# 2.6 Natural Language Analysis

Online Vehicle Tax Pay System is an web based application that helps to manage your tax payment of vehicles and calculate the amount of tax need to pay. Reducing time and money is the main aim of this project. It also helps to reduce human effort and provide satisfaction to the user.

This project consists of two actors i.e. Admin and User. The database of this project stores user’s data. User data includes Id, First name, Last name, Email, Password, vehicle type, insurance type and many more. System also includes review and rating of user towards application. Future plans of application based on review and rating from the users.

The system provide many features to the user with proper access control like Login, Registration, add date, expenses and income. User can able to enter into the system after registration and access to manipulate their own account. Manipulation of account include add, edit, update and delete their own personnel profile and data.

Natural Language Analysis is a process to separate noun, adjective and verb from the scenario.

* Noun represents candidate class
* Adjective represents candidate attributes
* Verb represents the operations or methods

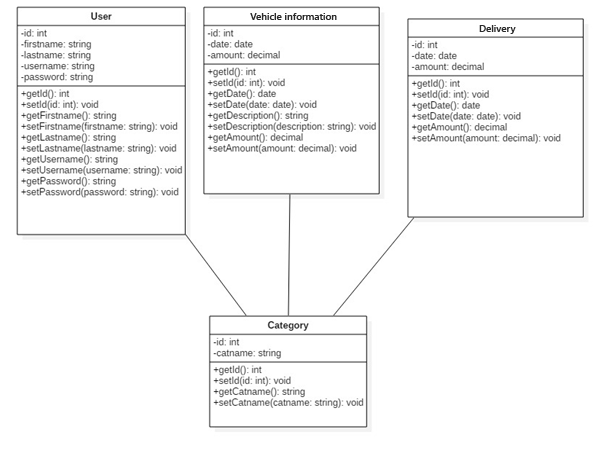
Following are the Candidate class, attributes and methods from the Natural Language Analysis:

|  |  |  |
| --- | --- | --- |
| **Noun (Candidate Class)** | **Adjective (Candidate Attributes)** | **Verbs (Candidate Methods)** |
| User, Category, Expenses , Income | First name, Last name, Email, Password | Add, Edit, Update, Delete, Review, Rate |

# 2.7 Initial Class Diagram

The popular UML diagrams and represented by the blue print of a system called class diagram. It shows the relationship and graphical representation between class, object, attributes and methods of a system. Class diagram is directly mapping with object-oriented languages thus it can used widely due to the phase of construction. It also helps in constructing executable code for application.

To draw class diagrams we have used many rectangular boxes with three sections: 1st section of the rectangular box represents the class name of the system, 2nd section represents the attributes of the class and finally third section represents the methods of the class. Association, Aggression and Composition are the components to show the relationship between different classes.



Screenshot 14: Initial Class Diagram