# Chapter – 2 Analysis

## Introduction to analysis

Analysis is way of dividing a whole topic into suitable parts for understanding. Analysis provides support to task. Analysis covers all the areas that is need to be search deep down and provide better understanding of task.

### **Why to perform analysis?**

This is an important part of task, which helps to make better decision to perform the task. Here I am preparing a web application for which analysis helps to complete this task by providing necessary information. This information will help this project to give better performance and better solution of project.

## Analysis methodology

This project is small-scale project. For this project, I have used Waterfall technique for developing and completing this project. In this project, I have determined to use ‘Hard approach’ methodology.

Hard approach methodology rigid techniques and procedures to provide unambiguous solutions to well-deﬁned data and processing problems, focused on computer implementations. This methodology contains many processes, which are explained briefly.

**Problem definition**

Problem definition gives what are problems or opportunity. This shows the problem that we are facing and opportunity to solve (improve) the current existing system. The initial step is to identify and describe problem or opportunity.

**Analysis of situation**

Analysis of situation defines the current state and performance level of the system. Here system boundary is defined.

**Identification of objectives and constraints**

Identification of objectives and constraints is about defining where we would like to be and the constraints that make affect our ability to achieve the new state. It is an important step because it forces stakeholders to clarify what they hope to achieve, but also to understand the external factors and constraints that will restrict our change choices and therefore the level of change.

**Measures of performance**

Measures of performance is about defining measurable means of assessing the efficacy of any definite possibility. It’s really asking and answering the question “how will we know if the change has occurred?

**Implementation**

Implementation is about design, development, deployment and installation tasks required to get the agreed proposal. This part also for collecting data on the measures of performance to show that the required change as occur. Here all planned phases will be carried out to reach the project aim.

**Data flow diagram (DFD)**

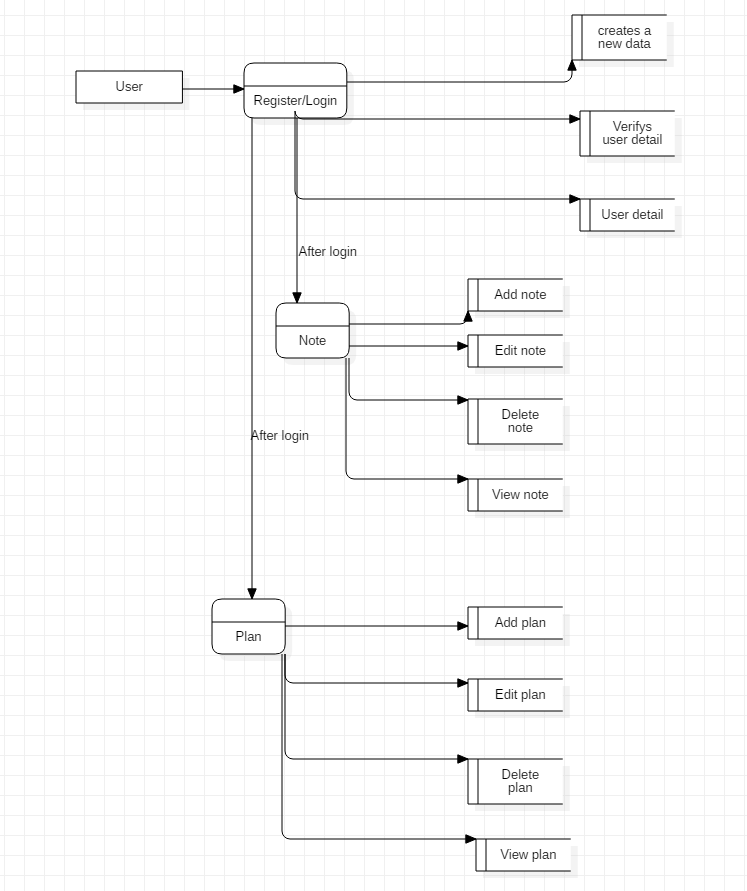
****

Figure 1 Data flow diagram

## Feasibility study

Feasibility study is topic that comes under analysis part. Feasibility study uncovers weakness and strength of an existing business or purposed one. This shows opportunities and threats which exists in the surrounding. A proper feasibility study evaluates the project potential success rate and helps to gain best outcome.

**MARKETING ASPECT**

This project is designed to replace all the physical (hard copy) note entry into digital form. The demand of this kind of idea is very huge for people who are into converting from physical to digital form of data. To attract all the prospective users, this system will be shared into social media and other various media. This process will make this project feasible to produce.

**TECHNICAL ASPECT**

The proposed project is suitable for the entire web platform. This platform is suitable for producing the system. All the required hardware resources are available so make feasible to deploy this project.

**FINANCIAL ASPECT**

This project is form academic purpose, which does not need any base cost for production. This make this project feasible to create.

**SOCIO-ECONIMC ASPECT**

This projects aims to contribute benefits to the people of any group age because people who are interested into shifting to digital form will get many benefits. One of the main benefits is that they will save cost to buy physical copy, that saved cost can be beneficial into another form.

## Software Requirements Specification

## Functional requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependencies** |
| FR-01 | User creation | Users should able to register account. | To recognize users to perform system function. | **-** |
| FR-02 | User authentication | Users | To maintain authenticity, confidentially and integrity to access the system. | FR-01 |
| FR-03 | Add notes | Users should be able to add notes. | To add notes details for manipulating in future. | FR-02 |
| FR-04 | Update notes | Users should be able to edit all existing notes. | To edit existing notes. | FR-03 |
| FR-05 | Delete notes | Users should be able to delete any notes created by that specific user. Confirmation dialog message box must be shown before deleting. | To delete notes which are no longer in need. | FR-03 |
| FR-06 | Display notes | Users should be able to display all their existing notes. | To display all note and for other functionality. | FR-03 |
| FR-07 | Search notes | Users should be able to search their notes as from their note titles. | To search notes which have been created. | FR-03 |
| FR-08 | Sort notes | Users should be able to order the view of all notes based on all attributes. | To sort all searched notes base on the attribute. | FR-03, FR-06 |
| FR-09 | Create notebook | Users should be able to create notebooks in which they can add new or existing notes. | To create notebooks which contains many notes. | FR-02 |
| FR-10 | End user session | Users should be able to log-out form system. |  | FR-02 |
| FR-11 | Add to do list | User should be able to add new planning. | To add planning details. | FR-02 |
| FR-12 | Edit to do list | User should be able to edit existing planning. | To edit existing planning for any changes. | FR-12 |
| FR-13 | Delete to do list | User should be able to delete existing planning. | To delete planning if not necessary. | FR-12 |
| FR-14 | View to do list | User should be able to view existing planning. | To display all planning which are complete or not complete. | FR-12 |
| FR-15 | View progress | Users should be able to view progress of planning. | To display current planning progress by showing complete and uncompleted plans. | FR-14 |
| FR-16 | User manual | Users must be able to view manual for any kind of help or to use system. | To provide guidelines to users. To show how to use system. | FR-01 to FR-19 |
| FR-17 | Change theme | Users should able to change theme. | To switch between light and dark theme for eye-comfort. | FR-01 |
| FR-18 | Text layout | User should be able to change text layout inside notes. | To change text layout according to user’s choice. | FR-03,FR-11 |
| FR-19 | Change font | User should be able to change fonts. | To change fonts according to users choice. | FR-03,FR-11 |

## Non-functional requirements

|  |  |  |
| --- | --- | --- |
| **ID** | **Category** | **Description** |
| NFR-01 | Interface | Interface of application must be easy and proper amount of guide should be provided. |
| NFR-02 | Performance | The system performance should be smooth and without any high latency. In normal condition, system should not lag more than 3 seconds. |
| NFR-03 | Security | Only valid users should be given access to system. All users must be authenticated before entering into system. |
| NFR-04 | Integrity | All the data of users must be kept safely and data must be as user saved. |
| NFR-05 | Usability | Application must be easy to use. Proper guidelines and instruction should be provided. |
| NFR-06 | Accessibility | The system must be accessible all the time when user’s needs. |
| NFR-07 | Pre-installed templets | Provide pre-installed templets to help write down note easy. |
| NFR-08 | Share-note | Share note between users. |

## Requirement Prioritization

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **MoSCoW** | **Rational** |
| FR-01 | User creation | Must | For recognizing users |
| FR-02 | User authentication | Must | For security purpose and for valid users to enter into system |
| FR-03 | Add notes | Must | Important function of the system |
| FR-04 | Update notes | Must | Important function of the system |
| FR-05 | Delete notes | Must | Important function of the system |
| FR-06 | Display notes | Must | Important function of the system |
| FR-07 | Search notes | Must | Important function of the system |
| FR-08 | Sort notes | Would | For sorting displayed notes |
| FR-09 | Create notebook | Must | For organizing the notes |
| FR-10 | End user session | Must | For logging out from system of current logged in users |
| FR-11 | Add to do list | Must | Important function of the system |
| FR-12 | Edit to do list | Must | Important function of the system |
| FR-13 | Delete to do list | Must | Important function of the system |
| FR-14 | View to do list | Must | Important function of the system |
| FR-15 | View progress | Must | Important function of the system |
| FR-16 | User manual | Must | Important function of the system |
| FR-17 | Change theme | Would | For changing themes |
| FR-18 | Text layout | Would | Change layout of text |
| FR-19 | Change font | Would | Change fonts of text |
| NFR-01 | Interface | Must | Important function of the system |
| NFR-02 | Performance | Must | Important function of the system |
| NFR-03 | Security | Must | Important function of the system |
| NFR-04 | Integrity | Must | Important function of the system |
| NFR-05 | Usability | Must | Important function of the system |
| NFR-06 | Accessibility | Would | Important function of the system |
| NFR-07 | Pre-installed templets | Would | Functionality of system to make note easy to add. |
| NFR-08 | Share-note | Would | Functionality of system to share note between users. |

## Hardware software specification

**Minimum hardware requirements**

* Processor (CPU) with 2 gigahertz (GHz) frequency or above
* A minimum of 2 GB of RAM
* Monitor Resolution 1024 X 768 or higher
* A minimum of 10 GB of available space on the hard disk
* Internet Connection Broadband (high-speed) Internet connection with a speed of 2 Mbps or higher
* Keyboard and a Mouse or some other compatible pointing device
* Sound card

**Minimum software requirements**

**Operating system:**

* Windows 7/8/10/Vista
* Mac OS X or Linux

**Browsers**

* Chrome\* 36+
* Edge\* 20+
* Mozilla Firefox 31+
* Internet Explorer 11+ (Windows only)
* Safari 6+ (MacOS only)

## Use case diagram

Use case diagram is diagram representing dynamic diagram in UML. Use case diagram shows functionality of a system using use cases and actors. Here actors refer to people or sub-system operating the particular system. System refers to something that is developed.

**Why make Use Case diagrams?**

Use case diagrams are very important for visualizing the functional requirements of a particular system. They also may provide aid for external or internal factors that may influence the system. They also provide high-level analysis from outside the system. They show how the system will interact with actors.

**Use of Use Case Diagram for my project**

All the necessary Use case diagram needed for my project are shown below. The description of each use case diagram scenarios are explained into table showing the necessary explanation.

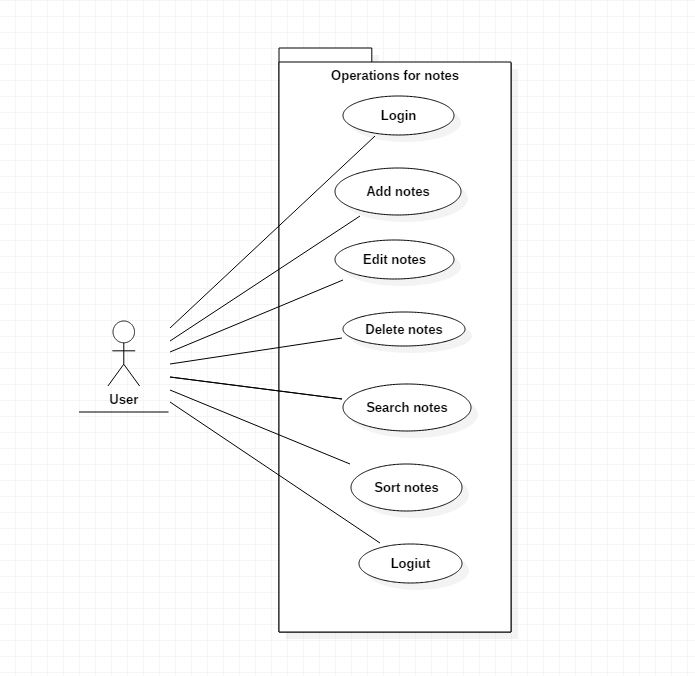
**Title: Add note**

Figure 2Use case diagram for notes

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | An important fundamental use case where data can be stored in form of note. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User goes into ‘add new note section’. 3. User inputs all the data. 4. System stores the data into database. |
| **Alternative flow** |  |

**Title: Edit note**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which edit the data from existing notes. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User goes for note to edit. 3. Users edits note. 4. User click on save button. 5. System process the data and saves into database. 6. System redirect outside the edited note. |
| **Alternative flow** |  |

**Title: Delete note**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which delete the data from existing notes. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User searches the particular note to delete. 3. System displays all the possible note. 4. User selects the note and click on delete button. 5. System displays a dialog box for confirmation. 6. User confirms and system deletes note. 7. System redirect to dashboard. |
| **Alternative flow** |  |

**Title: Update note**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which update the data from existing notes. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User search note that need to be updated. 3. System displays all the possible notes. 4. User select the note that needed to be updated. 5. User updates the notes. 6. System updates the note and saves into database. |
| **Alternative flow** |  |

**Title: Search note**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which search the data from existing notes. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User goes into search bar where user types the note title. 3. System search the title given by user into database. 4. System displays all the related data if found. If not found message is displayed to user that notes is not found. |
| **Alternative flow** |  |

**Title: Sort note**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which sort the data from existing notes. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User click on button to sort data accordingly. 3. System takes data from user input and sorts all the notes as accordingly. |
| **Alternative flow** | 1. User enters into the system from login. 2. User search the note title. 3. System displays the notes if found. 4. System sorts the note accordingly. |

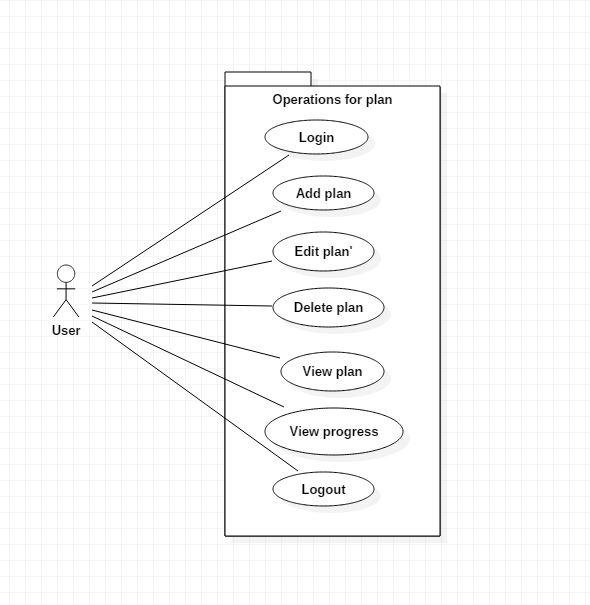
****

Figure 3Use case diagram for plan

**Title: Add plan**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which adds plan. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User goes into plan content. 3. User click on add plan. 4. System redirect into a new window. 5. Users inputs plan data and click on save button. 6. System saves into database. |
| **Alternative flow** |  |

**Title: Edit plan**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which edit the data from existing plan. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User selects the plan. 3. Users edit the data from that selected plan. 4. Uses click on save button. 5. System saves the edited plan into database. |
| **Alternative flow** |  |

**Title: Delete plan**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which delete the data from existing plan. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User selects the plan. 3. User click on delete button. 4. System shows a dialog box for confirmation. 5. User confirms. 6. System delete data from database. |
| **Alternative flow** |  |

**Title: View plan**

|  |  |
| --- | --- |
| **ID** | FR- |
| **Justification** | One of an important function which view the data from existing plans. |
| **Primary actor(s)** | User |
| **Supporting actors(s)** | NA |
| **Primary flow** | 1. User enters into the system from login. 2. User search the plan from title. 3. System takes user input and displays plan. 4. Users opens plan for viewing in detail. |
| **Alternative flow** |  |

## Initial class diagram

### **Brief description about project**

Todolist & Note management is a platform into which users can take notes and plan. This system allows users to create a new user. Users can also edit their profile. They can add, edit, delete and view their notes. They can view the existing notes by searching and sorting them. Users can also create a notebook where they can add one or more notes. In this notebook, users can add notes from existing notes or they can add new fresh notes. They can edit and delete notebook as per user’s view.

Users can make plan where they can add plan. They also view, edit and delete plans. Users can also view plan progress by showing how much have is completed and how much is left.

### **Natural language analysis (NLA)**

Natural language analysis is a method of finding out classes and functions that a program could have. This is made by reviewing the documents. In this case, description of project is a document.

**Nouns from document**

All the nouns are noted and listed in a single place for identifying candidate class.

* Platform
* System
* Note
* Plan
* User
* Profile
* Notebook
* Progress

**Selection of candidate classes**

* System
* Note
* Plan
* User
* Profile
* Notebook
* Progress

**Verbs from documents**

All verbs are identified and listed

* Take
* Allows
* Create
* Edit
* Add
* Delete
* View
* Search
* Sort
* Show
* Completed
* Left

**Selection of methods**

Since all the verbs are useful in this software so all the listed verbs are selected for methods used in this software.

**Class diagram (initial)**

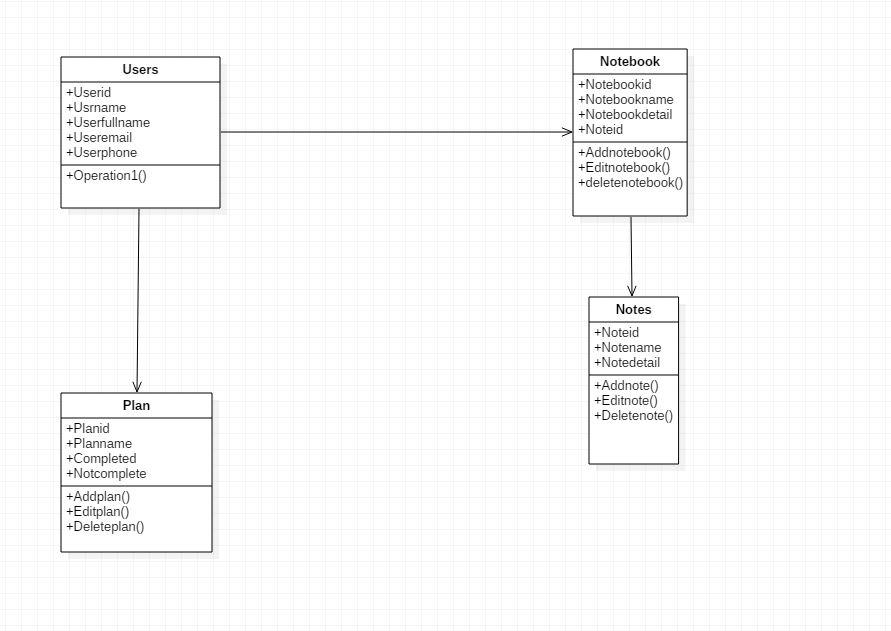
****

Figure 4 Initial class diagram