# Chapter 2: Analysis

## 2.1 Introduction

The analysis is the process of requirement gathering for the successful completion of projects. All needs, requirement of the project are gathered, studied and analyzed for making the further process more efficient and easy.

As for analysis, there are different techniques for gathering requirement. But for my project, I am choosing Survey for gathering requirement because of its capabilities for representing a huge number of population, low price and its convenient way of data gathering.

The analysis is the most important phases in every project so as for my projects its importance is:

1. It helps to transform or converts the needs and standard level of requirements of the company into complete, measurable and testable requirements.
2. Helps to study the feasibility of the projects by gathering the required information.
3. Helps to understand the problems faced by the company and user then searching alternative solutions to overcomes those problems.
4. It helps in the estimation of cost, time and skilled manpower that required for completing the project successfully.

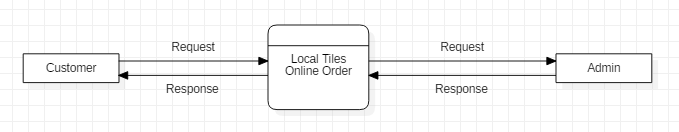
## 2.2 Analysis Methodology

Different types of analysis methodology are available in the market. But I have chosen **Hard Approach to Systems Analysis** and among it, I have chosen **Structured Systems Analysis and Design Methodology(SSADM)**. SSADM divides the developing project into stages, module, steps and provides a framework for describing projects in a way that can be managed.

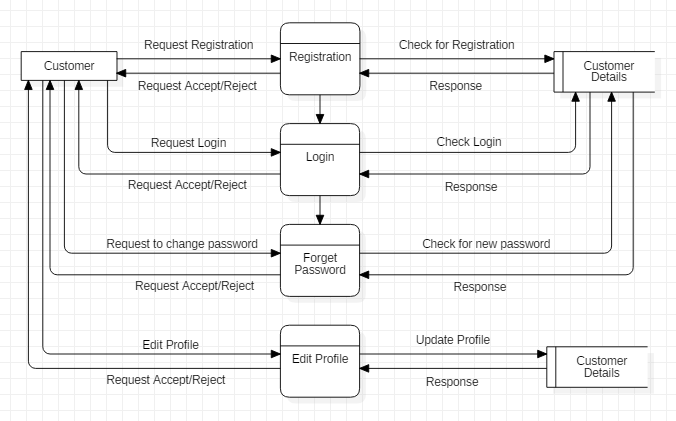
SSADM follow six steps and if all these steps are done thoroughly it can help to produce accurate information and well-documented system. The six steps are:

1. **Feasibility Study:** Feasibility study decided and examines. If a project is socially, technically, and financially feasible or not.
2. **Analysis and Requirement Specification:** Software, hardware and other details requirements of the system are analyzed. Business activity model is developed, define and investigate requirements and the logical views of the current system is obtained
3. **Design:** Design of all the aspect of the system like a behavioral, structural, user interface (UI) and database design are done.
4. **Implementation:** Once the project is completed and the system is ready. Then the system is deployed to the company either directly or running with the existing system in parallel until the new system is working successfully.
5. **Testing:** Testing is done when the development of the system is finished. It ensures that whether or not the functionality integrated to the system is working or not and it also ensure if there are other errors.
6. **Documentation:** After the system is completed then it is documented mentioning all the works and things that were done during the development of the system.

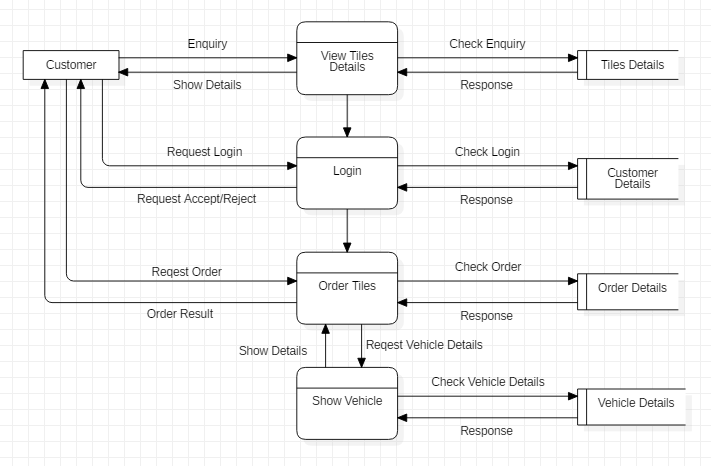
DFD is a step while undertaking this methodology. So, A Data Flow Diagram is a technique of representing a flow of data of a system. DFD describes the processes or steps which are involved to transfer data to or from the system.



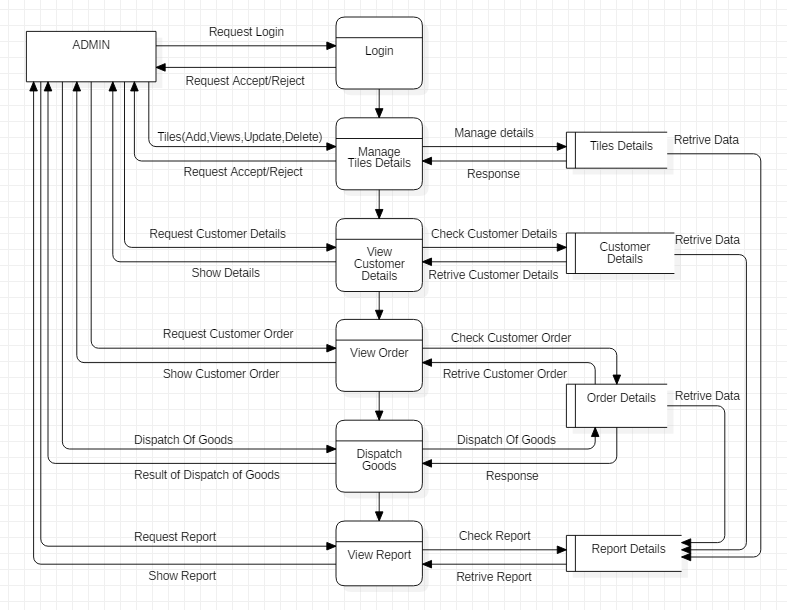
Screenshot: DFD of Local Online Tiles Order



Screenshot: DFD of Customer Profiles



Screenshot: DFD of Customer Order



Screenshot: DFD of Admin

Reason for choosing Hard Approach and SSADM in my project is:

1. As I am familiar with the waterfall model and SSADM is more or less similar to the waterfall model. It will be easier.
2. As it uses the waterfall model each step were completed before starting the next steps. Which ensure that every procedure related to every step are undertaken.
3. Progress of every step can be measured easily by the help of objectives defined for each step

## 2.3 Feasibility Study

A feasibility study is a study that is used to measure or finds out whether the project is feasible in all the relevant factor before it has been developed. Some relevant factor that the feasibility study addressed is technical, cultural, economic, operational and schedule.

Since it is part of the analysis it plays an important role in completing the project successfully which are:

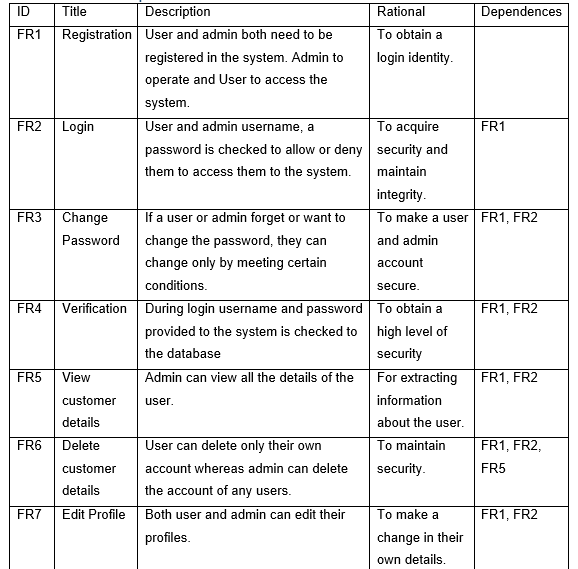
1. By identifying almost all the project related problems and providing alternative solutions.
2. It also helps to know the level of acceptance of the project by the users before the project finished.
3. It also helps to know if it is worth to invest in the projects or not.

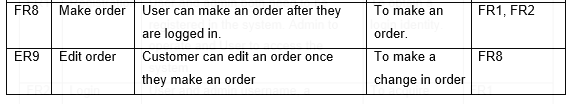
There are different types of feasibility study. Some of them are:

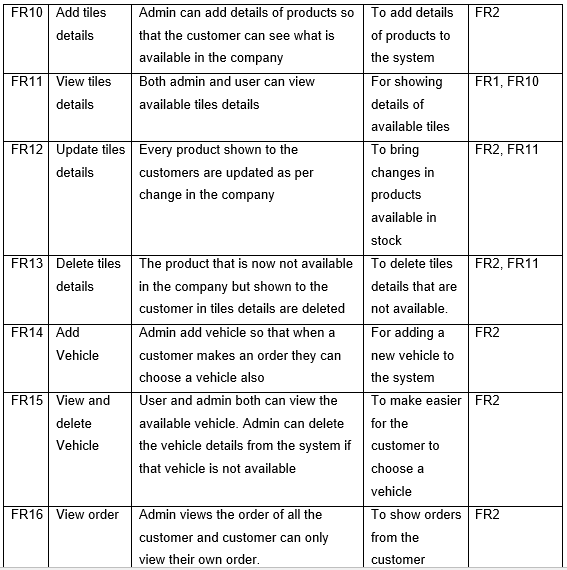
1. **Technical Feasibility:** Technical feasibility is the study about whether the technical resources required to undertake the projects successfully are available or not. It also analyzed if the existing technologies in the company can execute or support the system after it has been developed.
2. **Cultural Feasibility:** This is one of the feasibility studies that help to know whether there will be a positive or negative impact of the project on both local and general culture. Once it deployed in a real environment.
3. **Economic Feasibility:** In this feasibility whether the project is economically feasible or not is measured or identified. That means do the profit gain from the new system exceeds or greater than that of the cost required to develop a new system.
4. **Operational Feasibility:** This feasibility study identifies certain problems, its importance in the project and alternative solution to solve. It also analyzed whether the developing system is easier for users than the existing system.
5. **Schedule Feasibility:** The most important feasibility study for the successful completion of the project. It allocates time for the different module development and measure if there is time available to do the project. It also determines whether or not the project can be completed within the given deadline.

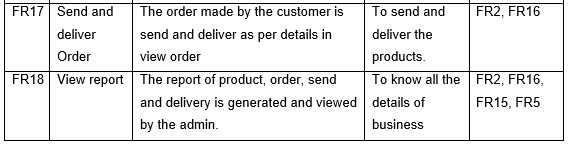
## 2.4 Requirement Analysis

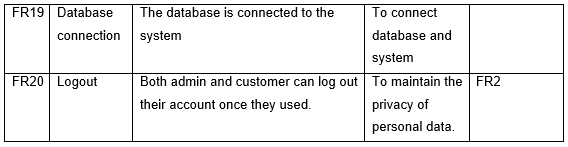
### 2.4.1 Functional Requirement



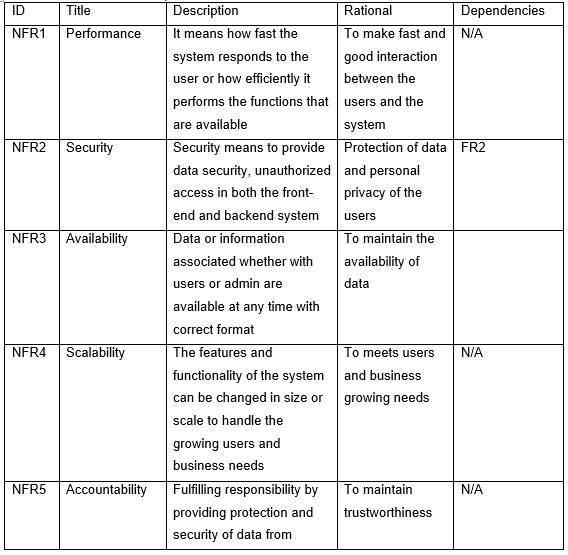


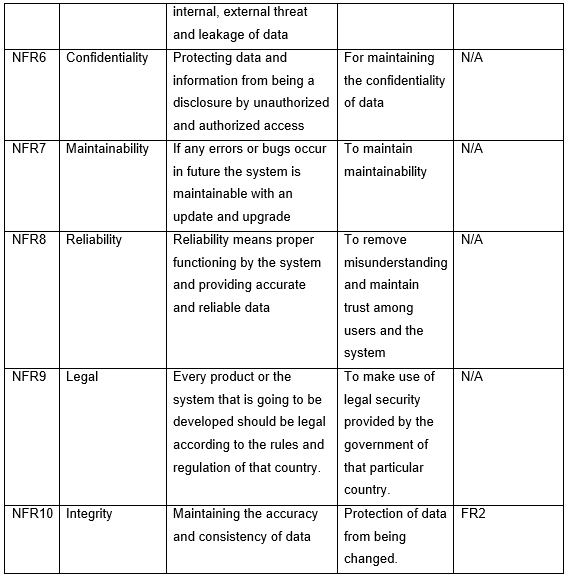






### 2.4.2 Non Functional Requirements





### 2.4.3 MoSCoW prioritization

MoSCoW prioritization is a technique which is used to prioritize the functionality and non-functionality requirement of the system. Which means every functionality and non-functionality of the system are not equally important, some of them can highly use, some can be the least use and some may not be used. So, to prioritize the functionality I used MoSCoW which means Must have, Should have, Could have and Won’t have.

**Must have**

Any functional and non-functional requirement which are guaranteed to implement and must be to make the system perform.

**Should have**

Requirements which are important to the system but they are not vital. So, if included they add significant value and if not included the system still function.

**Could have**

Requirements that are not required to perform the core function of the system. Compared with should have they have less impact on the system if not include.

**Won’t have**

Any functional and non-functional requirement which are not included and prioritized in a specific version but can be included in the upcoming version.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Title | MoSCoW | Description(why) |
| FR1 | Registration | Must have | To ensure and identify the person or company that they are who they claim to be making an order of good. |
| FR2 | Login | Must have | To maintain the security of the system from unauthorized access. |
| FR3 | Change Password | Must have | If by any reason users forget a password or someone knows the password then change password is required to mitigate those risk. |
| FR4 | Verification | Should have | Provides additional security during accessing some key functionality of the system. |
| FR5 | View customer details | Must have | To know information about the user who registered and logged in to the system. |
| FR6 | Delete customer details | Must have | To remove unnecessary registered or logged in user from the system. |
| FR7 | Edit profiles | should have | To provides an additional feature to the users. So that users can edit their information if anything changes. |
| FR8 | Make order | Must have | Since the system is about ordering the local tiles online. Users must be able to order tiles online. |
| FR9 | Edit order | Must have | By any reason, if users order different then they can edit their order. |
| FR10 | Add tiles details | Must have | The system is about selling the tiles online. So the admin adds the details of tiles online that are available. |
| FR11 | View tiles details | Must have | To see all the tiles details online at a time that are available in the company by the users or admin. |
| FR12 | Update tiles details | Must have | Admin must update the details of tiles from the system as per availability, non-availability and changes in the company. |
| FR13 | Delete tiles details | Must have | To remove tiles details that are not currently available in the company. |
| FR14 | Add vehicle | Must have | To provide promote business and provide extra facilities for the customers. So that they can easily dispatch their order as their needs. |
| FR15 | View and delete the vehicle | Must have | To show the available vehicle and delete those which are not available in the company this time. |
| FR16 | View order | Must have | To know details of order made by different customers |
| FR17 | Send and delivery order | Must have | To send and deliver the order of the customers as per their orders in the right place and the right time |
| FR18 | View report | Must have | For the success of the company every report of orders, add tiles, send and deliver tiles needs to be shown in the report. |
| FR19 | Database Connection | Must have | To connect the system to the database. So that to can perform the business smoothly and successfully |
| FR20 | Logout | Must have | Every personal privacy of the users is in the system. So, as to protect them logout is done. |
| NFR21 | Performance | Should have | The performance of the system with the users should be fast. So as to satisfy customers. |
| NFR22 | Security | Must have | To protect the customer and the company personal information. |
| NFR23 | Availability | Must have | Both users and admin need to access their information at any time so availability is needed. |
| NFR24 | Scalability | Should have | The company or users requirement can grow in coming days so the system should be scalable. |
| NFR25 | Accountability | Must have | To secure the data from internal, external threats and leakages of data. |
| NFR26 | Confidentiality | Must have | To maintain the security of data from being disclosure. |
| NFR27 | Maintainability | Should have | If any errors in coming days then the system should be maintainable. |
| NFR28 | Reliability | Must have | To increase trust among users or customers. |
| NFR29 | Legal | Must have | To protect the system legally. |
| NFR30 | Integrity | Must have | To increase and maintain the accuracy of sensitive information. |

### 2.4.4 System Requirement Specification

A System Requirement Specification is a complete described document of the system to be developed. Which include non-functional, functional and features of the system. System Requirement Specification also describes the user's interface, hardware, and software requirements required to develop and run the system after being deployed in the real world.

The software and hardware required for my projects are listed bellows:  
Pre-project requirements: All the software and hardware required for completing the system from requirement gathering to deployment stages.

|  |  |
| --- | --- |
| Hardware | Software |
| * Laptop | * Windows 10 Operating System |
| * Processor i5 intel | * XAMPP (Local Server) |
| * Minimum Hard Disk 300GB | * MySql (Database) |
| * RAM (4GB) | * Star UML |
|  | * JDK |
|  | * JVM |
|  | * Net Beans |
|  | * Project Libre |

Post Project Requirement: Hardware and Software which are required at least a minimum requirement to run the system in the users existing systems.

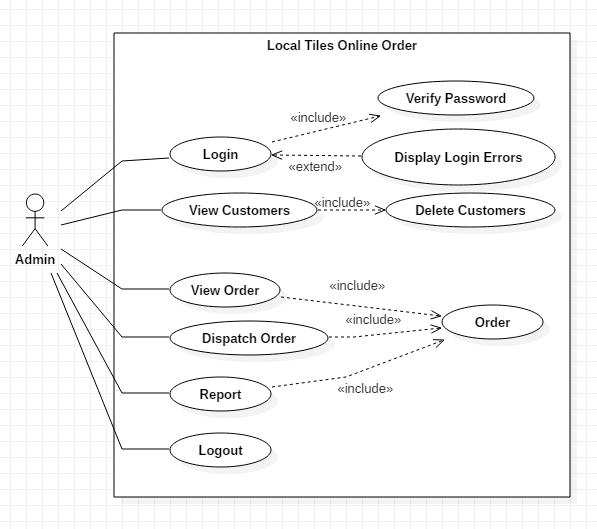
|  |  |
| --- | --- |
| Hardware | Software |
| * Laptop, PCs | * Operating System Windows 10, 8, 7. |
| * Minimum processor i3 intel | * Opera, Google Chrome, Mozilla Firefox |
| * Minimum RAM (2GB) |  |
| * Minimum Hard Disk (300 GB) |  |

## 2.5 Use Case Diagram

Use Case Diagram is a visual and textual representation of a user’s interaction with the different use-cases of a system. It also shows the relationship between the system, actor, and use-cases.

To draw use case diagram different notation are used which are as follows:

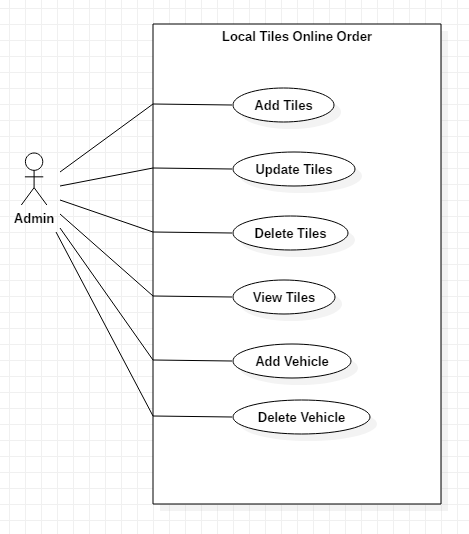
* **Actor:** A role played by the customer in the system with defined roles to interact with the use-cases in the system.
* **Use Case:** Steps or functions of the system which represents a user goal and can be achieved by accessing the system.
* **System:** Represent by a shape in which the system use cases are placed inside it.
* **Relationship:** Notation that shows the relationship of use-cases between and among customers.



Screenshot: Use Case from the admin

Above use-case shows the functionality that actor(Admin) can access:

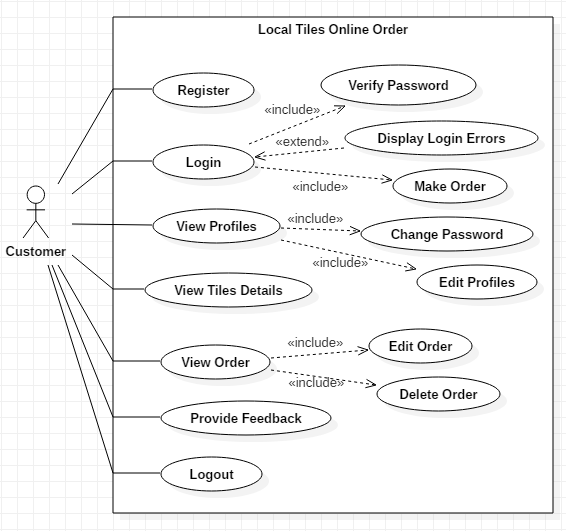
* At first admin login to the system then can access the functionality of the system.
* Admin views customers detailed information and can also delete the customer as per needs.
* Tiles order by the customers is also viewed by admin and dispatch orders as per information provided by customers.
* Admin generates a report of orders and customers at the end of the day and Log out the system to maintain security.



Screenshot: Use Case of Admin

Above use case also show the functionality than actor(Admin) access:

* To access this functionality also the admin first Login to the system.
* Admin adds tiles detail at the starting of the day and deletes, update details at the end of the day.
* Admin too adds new vehicles detail if needed and delete if not available.



Screenshot: Use Case of Customer

The above use case shows the functionality that actor(Customer) can access:

* Customer first needs to register and then log in to access some key functionality.
* Customer can view details of tiles available in the company.
* They can place an order and select a suitable vehicle to transport their order.
* They can edit and delete their orders, profiles as their needs.
* Customer can provide feedback about the systems and products.
* Finally, customer Logout to the system to secure privacy.

## 2.6 Natural Language Analysis(NLA) and Initial class diagram

Shreesh Tiles Company is a small scale industry located in Galkot, Baglung. Till now all the works of the company is done manually and with phone calls. So the owners of the company now want to computerized whole systems.

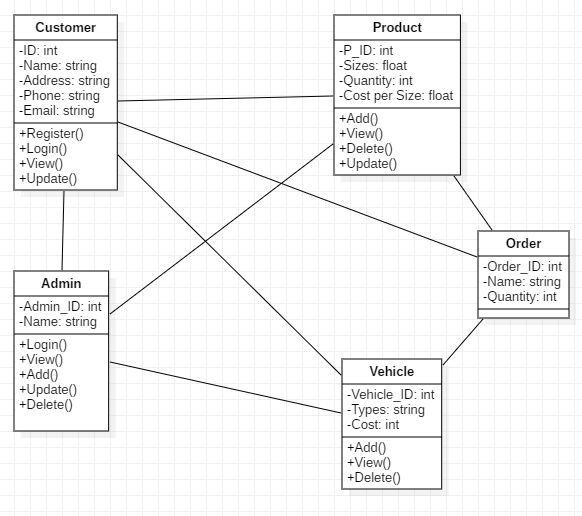
The company wants the system where there will be two users Admin and Customers. Admin or the owner of the company can add, update and delete details of products and vehicles. The owner also can see all the details of the customers and order place by customers then dispatch the order as information provided by the customers. In the other side, customer can see all the details of tiles available in the company but need to register to place an order. Once they register and place an order they can edit or delete their profiles and order as their needs.

From the above scenarios, Natural Language Analysis(NLA) is performed and candidate classes and method are separate from nouns and verbs.

|  |  |
| --- | --- |
| Candidate Classes | Candidate Methods |
| * Admin | * Add |
| * Customer | * Update |
| * Product | * Delete |
| * Order | * Register |
| * Vehicle | * Login |
|  | * Generate Report |
|  | * Change |

**Initial Class Diagram**

Initial Class Diagram is a static diagram that helps to define the structure of the system by showing classes and attributes that fall under it, operation of the system and relation between them. The initial class diagram also shows initial classes of the projects that further help in making a final class diagram.



Screenshot: Initial Class Diagram