# 2. Analysis:

Analysis is the detail study of requirement for development purposes. It is the process of studying or examining something in an organized way to learn more about it.

Analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a technique for solving problems that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

# 2.1 Analysis Methodology:

It is the method of analyzing the performance of the system. There are numbers of methodologies which can be chosen according to the requirements and specifications. Some of them are:

1. Hard approach: - focus on standard structure
2. Soft approach: - focus on people views and opinions
3. Combined approach: - focus on both human and technical aspects

Hard approach is appropriate for this project. In this project, we’ve to focus more on technical aspects rather human views and opinion. We will be following standard structure, rules in development process. Also, it maintains the quality of the system. Appropriate for our kind pf organization is another reason of using it. Hard approach must be applicable for this project because of following reason:

* Focus on system need i.e. technical aspects
* Follow standard rules and regulation
* Easy to track down progress
* Maintain standard and quality

# 2.2 Feasibility Study:

Feasibility study is an evaluation of the practicality of a proposed plan. It analyzes whether the project is economically, technically, legally feasible or not. Its main aim is to detect all the possible issue, problems that could arise throughout development process. Feasibility study also allow business to address where and how it will deal with the problems and obstacles. It will also help to make strategy.

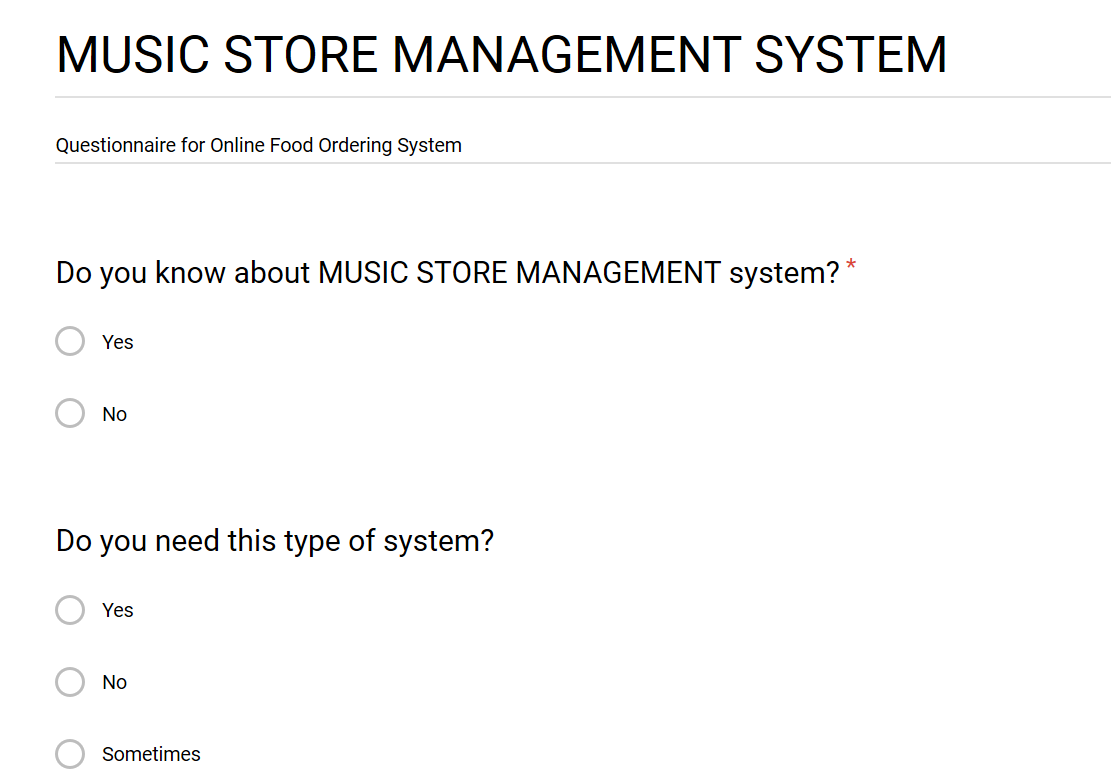
Different components of feasibility study that helps in project are:

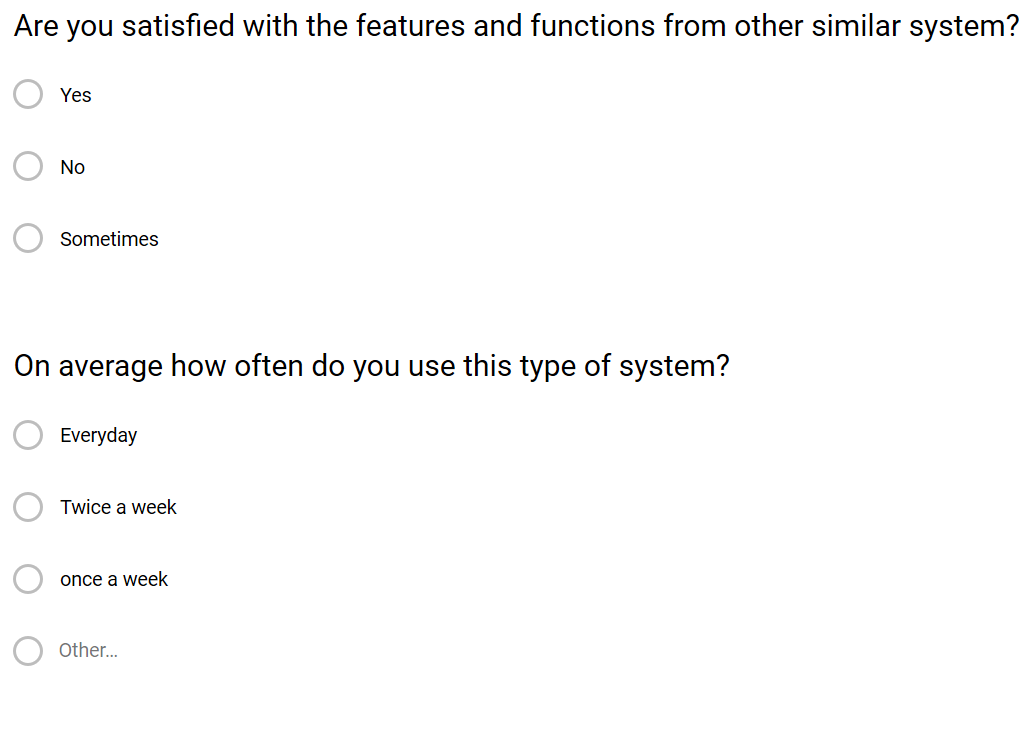
1. Financial feasibility: - It is the prediction of money needed at first, how it will be using and the return from the investment. Software that we will use for the development process are open source such as SQL Server. These softwares are reliable, trustworthy, widely used and are developed by renowned company. Developing this system is financially possible according to the budget.
2. Technical feasibility: - This is the study of the currently available technical environment, gadgets, softwares. Since we will be using open source softwares, good hardware and the system will be developed by well trained, qualified developer, there will not be any issue regarding to the technical aspects. Thus, the system is technically feasible.
3. Market feasibility: - To make the software/system successful we have to study current market scenario carefully. It studies the current market needs and ensure us that the system will fulfill the demands. This also helps us to analyze the similar product available in the market and makes sure that the system will able to compete with them.
4. Legal feasibility: - Our system must have to fulfill the legal criteria defined by the government. Also, it doesn’t contain any things that may affect other. This study finds out that our system will not break any rules and regulation. Our system will be fully This help to study whether software will be legal or not. This system will contain only ethical things. Our study finds out that it doesn’t have any issues with the law.
5. Social feasibility: - System need to be accept easily by the users. It provides us the knowledge of what users want and like from the new software. It will help us to develop the software which society accept easily. Basically, this deals with the ethics. And in our case, we will do things which are ethical. This makes possible to develop system.

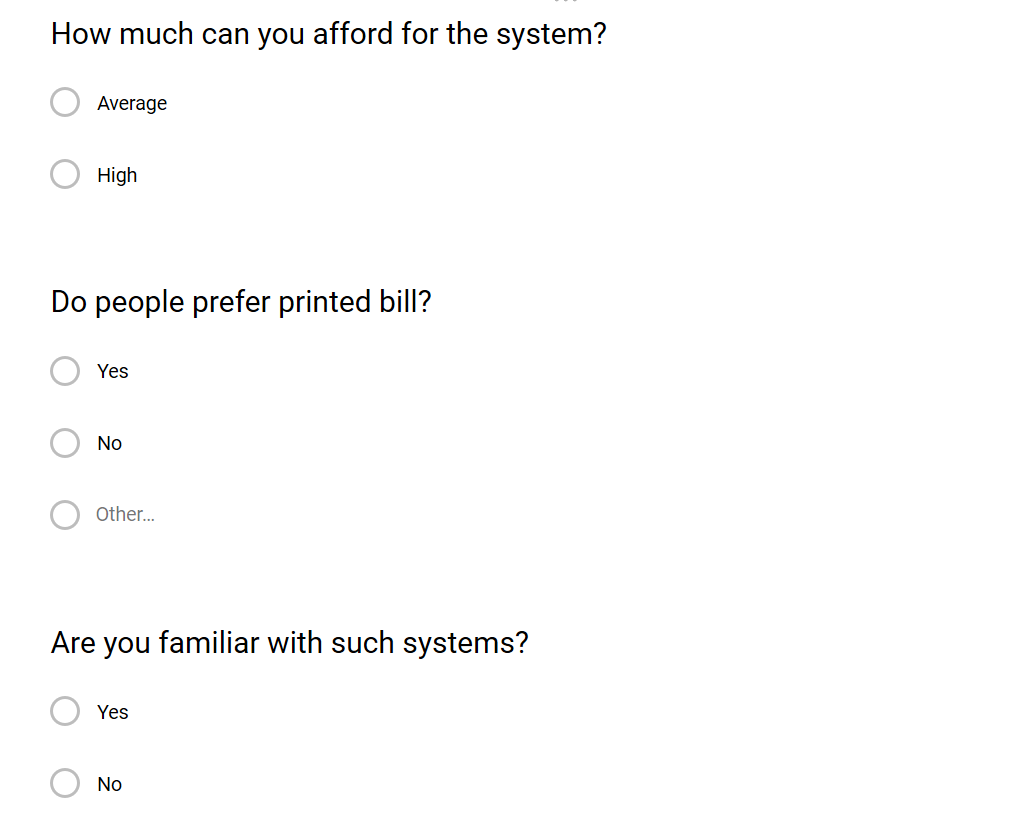
# 2.3 Requirement analysis:

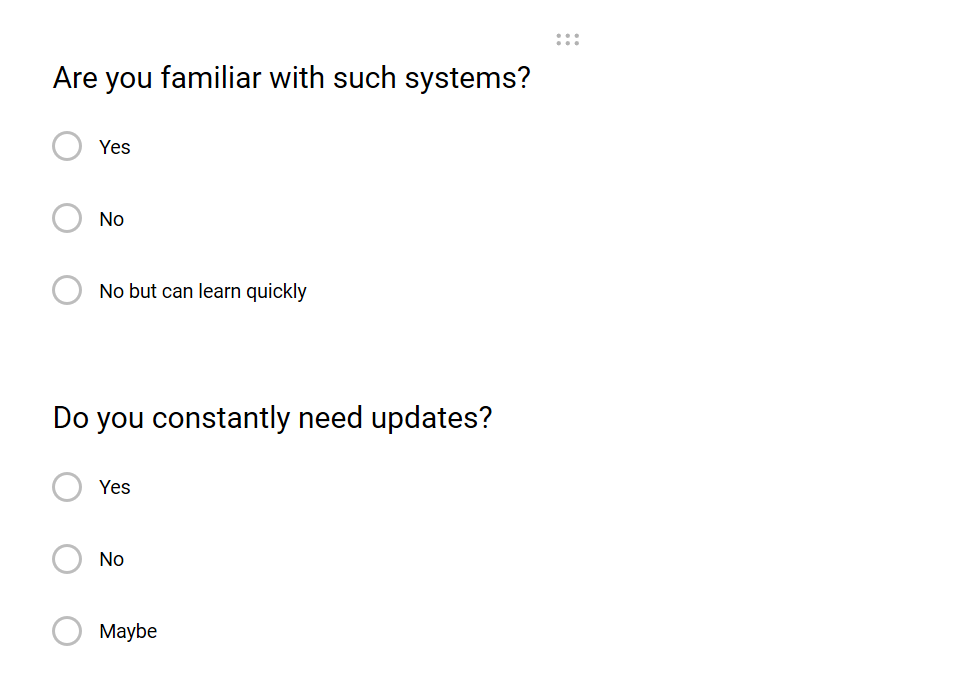
Analyzing the user prospective for new or modified product is requirement analysis. Correct requirements should be identified to make the software successful. To find these, we need to focus on users’ demand, their problems with similar kinds of software. There is multiple way of requirement analysis. Some of them are:

1. Interview: - Face to face conversation with people is known as interview. This is more accurate and reliable technique of gathering requirement and information. This help to know the different views and opinions. This technique is more suitable if we have to consider people more than technical aspects. Accurate and reliable requirement gathering technique comes with more consume of time and money.
2. Questionnaires: - Questionnaires is the technique of gathering information from large number of people without talking face to face. It contains a set of questions and their possible answers and user just have to choose from them. This process is very fast, less money consuming but not much reliable and accurate. We can perform this technique if technical aspects are more important.









1. Observation: - Analyzing the user natural behavior of solving problems is observation. It helps us to determine the system features. It understands how user perform any task without using any automated system.

# 2.4 Software Requirement Specification (SRS):

**Functional specification:**

These are the functions or features that must or should be included in the software. This defines what the system is supposed to do. These are the functions which users use. These will be the most using features or functions of the software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **DESCRIPTION** | **RATIONAL** | **DEPENDENCY** |
| FR1 | Registration | Admin, employee, users should able to register in the system | To complete login | FR5 |
| FR2 | Login | Admin, employee, users should able to login in the system | Access to features and functions of the system, to maintain security | FR1, FR5 |
| FR3 | Report generation | System should allow all user to generate report such as bill, information, etc. | For personal and other paper record and use | FR3, FR5 |
| FR4 | Manage details | Update and delete option should be include in the system so that users can update and delete information such as profile, equipment, etc. | For system functionality and usability | FR4, FR5 |
| FR5 | Database | All the data should be store and retrieve when needed | Maintain efficiency, accuracy and well managed | F1, FR4 |

**Non-functional specification:**

These are the functions or features that are not essential to user but must be included in the system. Bad behavior of non- functional specification may affect the functional specification. Imagine one have access to many accounts with same username and password. Some of them are:

1. Performance: Every task should be completed as fast as possible. Also, it should be able to generate accurate and reliable results.

Imagine what will happen if one user has to wait 10 minutes just to login. It will create a bad impression. Worst he will stop to use our software.

1. Security: One of the most important non-functional requirements is security. It should be included to maintain privacy, availability and integrity. Policies such as strong password should be included.

Think one can access to the system with any username and passwords. What will be its result?

1. Scalability: Business may increase in the future. So, we have to develop a software that have capacity to change in size according business growth. It should able to handle the increasing users and also doesn’t affect the performance.
2. Modifiability: This is the process of changing the software partially according to the requirements. We make changes the existing system rather than making new one. It makes the code re-usability which save the both time and cost without compromising performance.
3. Reliability: Maintenance should be done after certain period of time regularly. This helps us to gain the consistent performance i.e. speed and accuracy. This will help to gain trust and believe of the clients.
4. Availability: User should able to access to their data without any problems. These data should be accurate, useful and right.

**MoSCoW prioritization:**

It is the prioritization technique for managing requirements. It is the common agreement between stakeholder and developing parties.

MoSCoW stands for:

M: Must have => must include in the software; insecure, incomplete without it

S: Should have => essential but not necessary in that time

C: Could have => desirable but less required

W: Won’t have => not required

It states that we have to work on must have, should have and could have prioritization respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **REQUIREMENT** | **MoSCoW** | **RATIONAL** | **REMARKS** |
| 1 | Create account | Must have | Register user to use the system |  |
| 2 | Login system | Must have | Give access to the system |  |
| 3 | Manage details | Must have | Provide power to changes the details like username |  |
| 4 | Database | Must have | Store and retrieve data |  |
| 5 | Performance | Should have | Get accurate, reliable data in less time |  |
| 6 | Security | Must have | Maintain credentiality, integrity and privacy |  |
| 7 | Scalability | Should have | Help to change in size according to business |  |
| 8 | Modifiability | Could have | Save money, time by changing existing software partially |  |
| 9 | Reliability | Must have | Access to corresponding data easily |  |
| 10 | Availability | Must have | Data access |  |

**Hardware/software specification:**

Hardware and software play a vital role to run a software smoothly. Our proposed software needs suitable hardware to perform smoothly. They are essential to eachother to run smoothly. I’ve figured out some of them:

**Hardware specification:**

* Processor: Celeron 500MHZ or any PENTIUM Processor
* RAM: 2 GB minimum
* Hard disk: 50 GB minimum (SSD best)
* Display type: Standard VGA or SVGA card

**Software specification:**

* Operating system: windows
* Database: SQL Server
* Platform: .NET/ C# language
* Visual studio

# 2.5 Use Case diagram:

This diagram is used to show how actor perform any function a goal. Actor are those who use the system simply human being. It consists the group of classes. It shows how the system will respond to the request from users’ point of view.

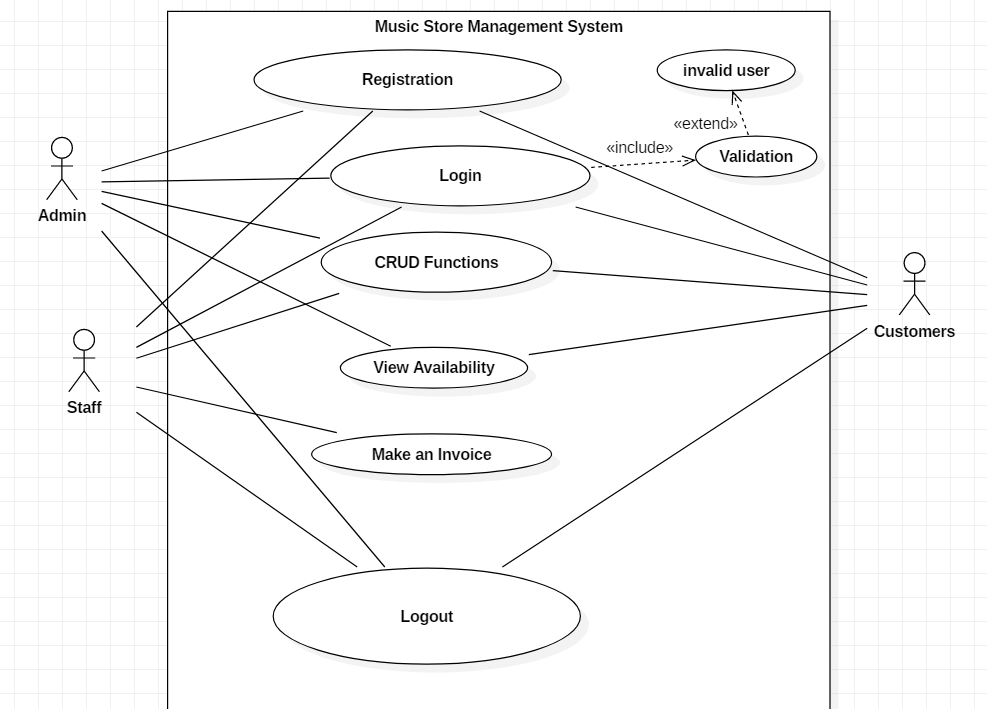


Figure :use case diagram

**NLA:**

Natural language analysis is the process of identifying nouns, verbs and adjectives. Nouns relate to potential classes, verbs relate to potential object (function), adjectives relate potential attributes.

|  |  |  |
| --- | --- | --- |
| **Nouns (Classes)** | **Verbs (Function)** | **Adjectives (Attributes)** |
| Admin  User  Staff  Product | Login  Add  Remove  Delete  Update  Logout  Verify  View | Admin id  Admin Name  User\_id  User\_name  Staff\_id  Staff\_name  Password  Staff\_contactno |

# 2.6 Class Diagram:

We perform this technique to find out class, attribute and operation. This is the diagram that shows the class, their operation and their attributes and relationship between objects. In the table top compartment represent class, middle compartment represent attributes and lower compartment represents operation.

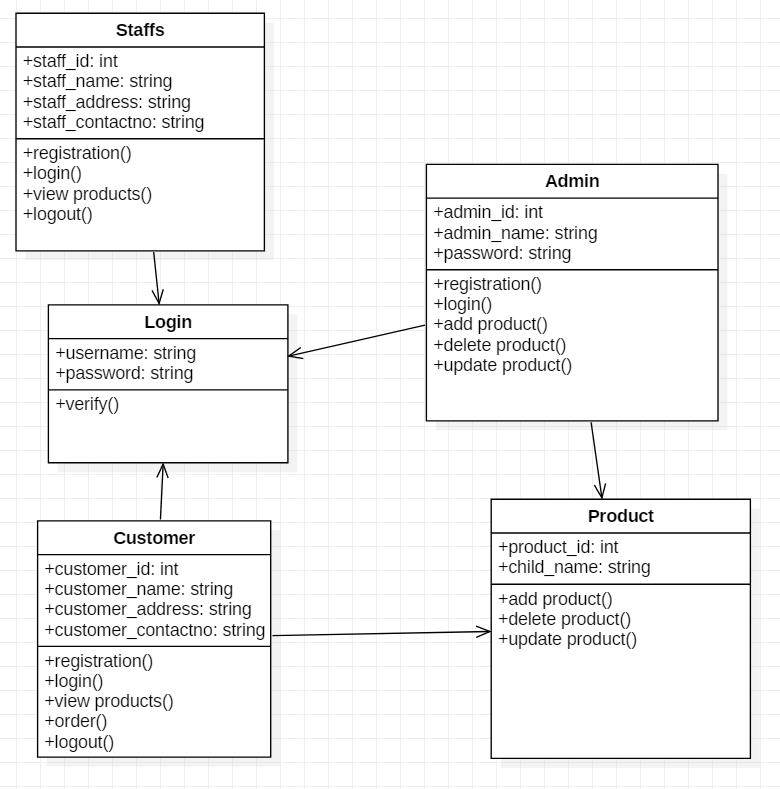


Figure 2:class diagram

# 2.7 System Architecture:

System architecture is conceptual design of a system that defines the structure, behavior and more views of the system. We will use three tier system architecture in this project. It is composed of three layers or tier of logical computing.



Figure : Three-Tier architecture

The tree tiers in three-tier architecture are:

1. **Presentation tier:** It is also known as first tier. This is the front-end layer and consist of the user interface. It translates the tasks and results to something the user can understand.
2. **Application tier:** It is also known as second tier. This layer coordinates the application, process commands, makes logical decisions and evaluation and perform calculations. It moves and process data between other two tier.
3. **Data tier:** It is also known as third tier. Here information is stored and retrieved from the database. the information is then passed back to the logic tier for processing and then eventually back to the user.

The reason behind using three-tier architecture are:

* Maintainability: - because each tier is independent of the other, updates or changes can be carried out without affecting the affecting the application as a whole.
* Scalability: - because ties are based on the deployment of layers, scaling out an application is reasonably straight forward.
* Flexibility: - because each tier can be managed or scaled independently.
* Availability: - application can exploit the modular architecture of enabling systems.
* Reusability: - components are reusable.
* Faster development: - because of division of work.

# **Conclusion:**

We perform an analysis for this project. We use hard approach as analysis methodology. We perform feasibility study and find out the factors that may arise during development and how we will overcome them. MoSCoW prioritization is carried out to prioritized and manage the requirement. Use case and class diagram are created to show the relationship between user and system and between objectives respectively. NLA makes our job to find out the class, function and attributes. Also, we described hardware/software requirements to run the system. We follow the three-tier architecture.