**Chapter 2**

**2.1 Introduction to analysis**

Any detailed examination of anything complex in order to understand nature or to determine its essential features is called analysis **(Merriam-webster.com, 2019)**. In other words, it is the process of breaking down complex thing into small parts that can be understand easily. For any project it is important to understand what is happening and how the things will work. This all is done through a process called analysis. It is an important phase in Software development life cycle. In this phase the requirements are captured accordingly, in this phase a customer/client expresses what he wants the system to do in order to fulfill his requirements. Technical person or the person from developer team analyse each requirement provided by the users/clients and make sure that the requirements can be fulfilled in the software in the system without affecting its system functionality and without causing any problems.

Analysis is done for following reasons:

**Studying the current system**

Any project is initiated because there might be problem with the existing system. In order to make the current system efficient and make it free from errors and bugs new system needs to be designed. Analysis will help in collecting the facts from existing users also it will help to collect information about current system boundary, details of the system and the people affected by the system.

**Defining new system objectives**

Analysis will help to prioritize user requirements. It will give a clear idea of what the system should be and will make the developer team (Technical person) clear about the needs of user/client from the system. Analysis will help in understanding each and every aspects of current system and indicate how the things can be made efficient with the deployment of new system.

**2.2. Analysis Methodology**

The literal meaning of methodology is theoretical, systematic study of the methods and ways applied to any fields of study. Analysis refers to studying and methodology refers to the systematic approach, so combing these, analysis methodology means the methods of studying the system by help of various resources that will help in making the study efficient and less time-consuming. There are mainly 9 analysis methodology that can be used in studying the system. Among them I will use Soft System Methodology. Soft approach involves six main steps.

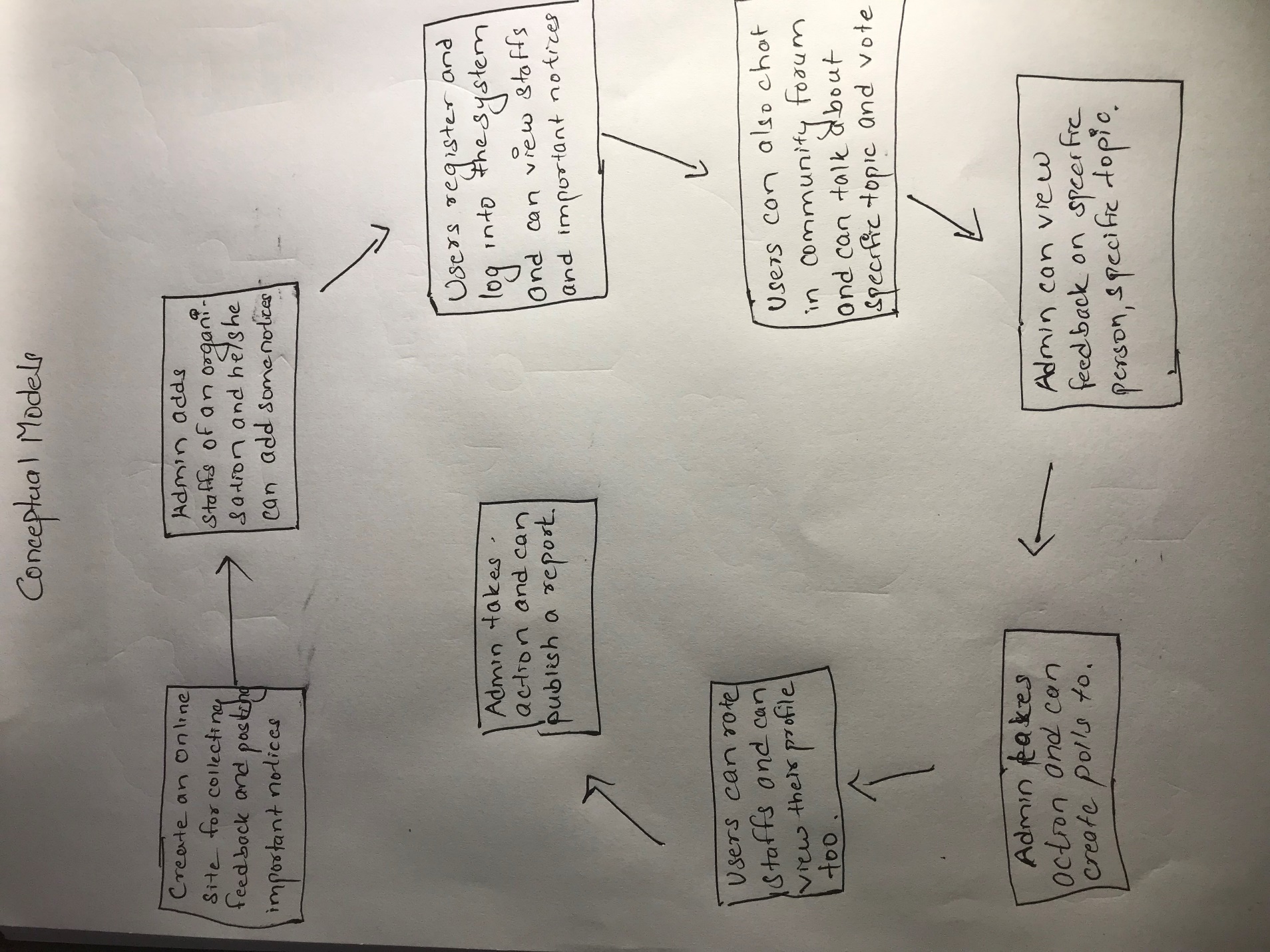
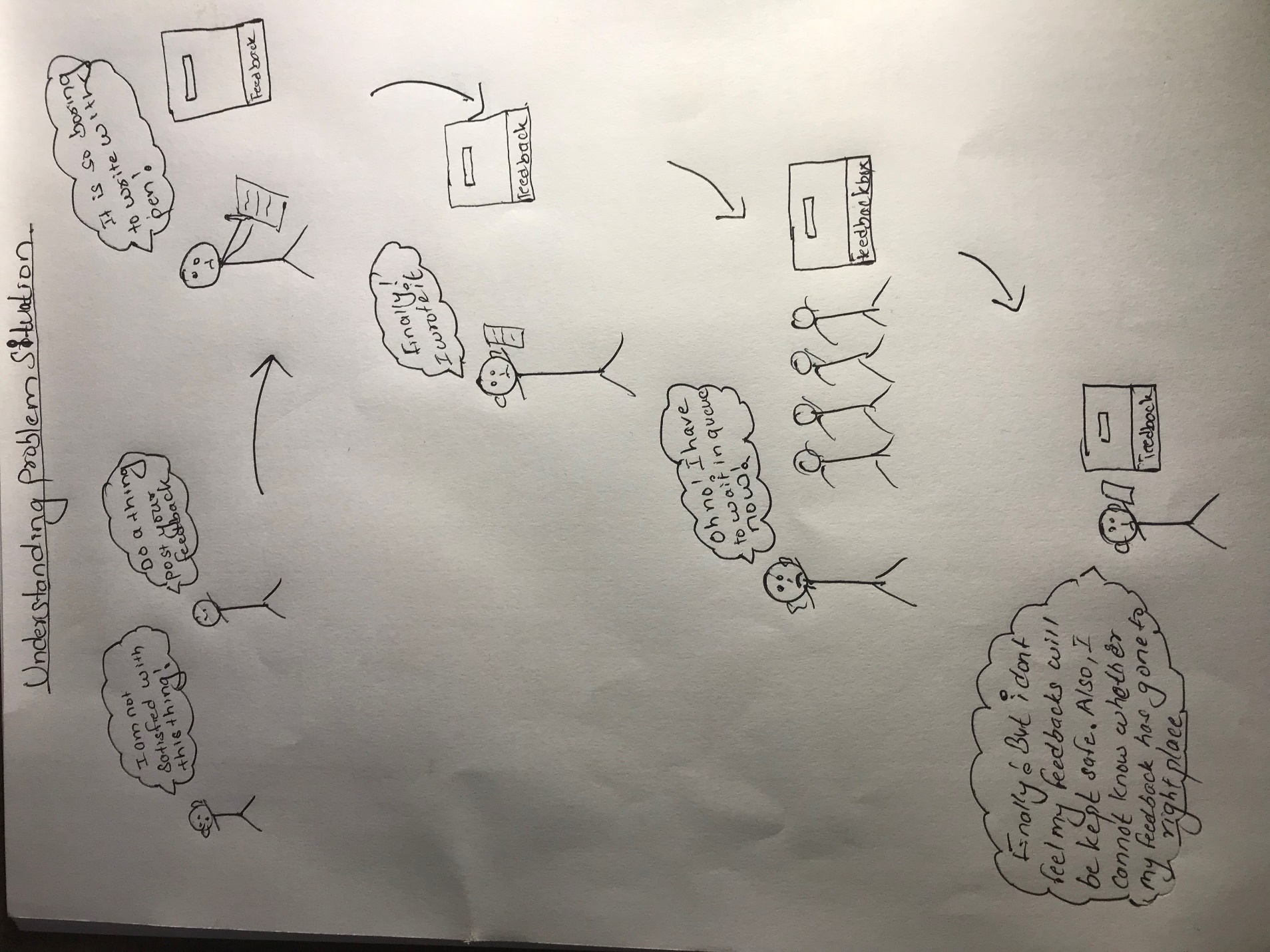
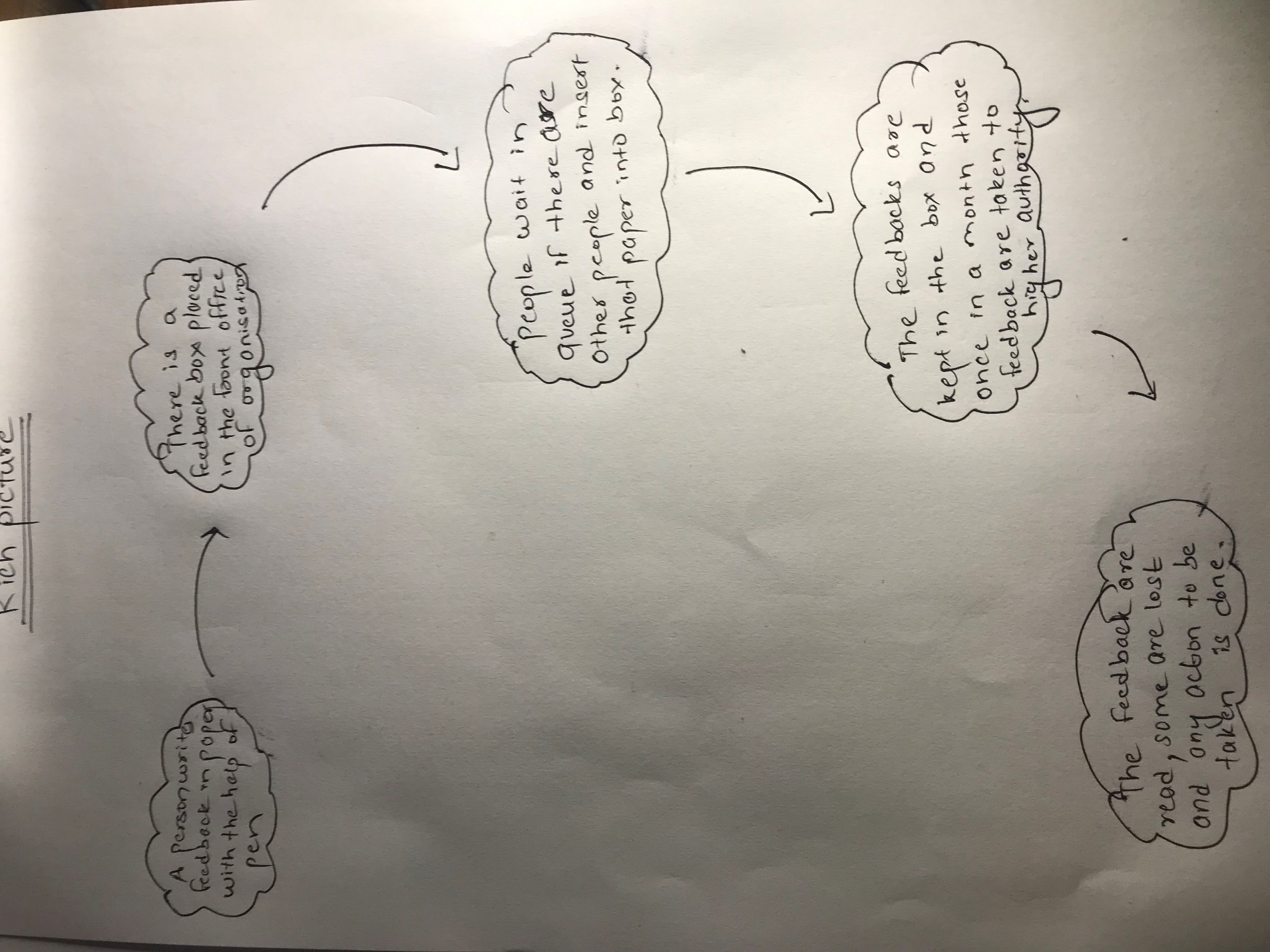
* Analyse current system and produce a rich picture. A rich picture is somehow like a Data flow diagram but the specific process is not followed while drawing a rich picture.
* Second step is to define a root definition of significant parts of the system. Root definition can be issue based and primary task based. Problems faced by the current system is shown in this diagram.
* Third step is to conceptual diagram. It is a diagram in which an ideal system is shown, it models on how the data should flow and it gives an idea on how the system should be and function.
* Fourth step is to compare the conceptual system with the actual problem.
* Fifth step is to select feasible options that assist for easy development of the system, both developer and client team discuss on this topic.
* Sixth and the final step is to implement the new system.

**Advantages**

* It helps to structure a complex organisational and political situations.
* It also focuses on user involvement rather than technical specification.
* It allows developer to roll back and repeat the phages if necessary.

**Disadvantages**

* It has no definite technique modelling tool, so it is hard to know about the correctness of diagrams.
* Since no specific rules and process are followed it is impossible to track the progress.
* This can sometime take too long time to reach an agreement, so there is chance that the software will not get delivered.



**2.3. Feasibility study**

It is a study which involves an estimation of the level of proficiency mandatory for a project that can provide qualitative, qualitive assessments of various other resources, identification of mandatory points, general timetable and general cost estimation. **(My Accounting Course, 2019)**. This type of study helps for determining the possibility of an idea. And also, it helps in confirming that the project is legally, technically, socially, economically feasible. It also gives an idea on whether a project is worth the investment. Mainly there are five types of feasibility that should be done:

1. Technical feasibility: It gives a clear idea of capability of the developer/technical team of any organisation.
2. Economic feasibility: It gives a clear idea on the economic status of the project. It also gives an idea about the economic benefit that the project will give after the project is completed. So, in this phase a decision on whether a project should be initiated or not.
3. Legal feasibility: This type of study gives an information on whether a project is legally valid. This type of study gives an idea on legal boundary of the system.
4. Operational feasibility: It studies on how the project is going to solve the current problem faced by an organisation.
5. Scheduling feasibility. It is the most important study that needs to be done before initiating a project. It will give a clear idea on whether the project will get completed in given time frame and complete the given requirements.

**2.4. Software Requirement Specification**

Software requirement specification is an explanation of system that is to be developed. SRS helps to lay out non-functional and functional requirement and this involves a set of use cases that explains user interaction that the software must have for the user satisfaction. It helps in maintaining an agreement between client and developer team on how the software will be and how a software should function. It helps in providing a realistic estimation on product costs, risks, schedules etc. It can help to prevent software project failure by identifying risk in early phase and the ways to mitigate them. Software Requirement specification helps to minimize the efforts and time that developers require a specific goal and also help in minimizing development cost. Software requirement specification serves other purposes. They are listed below:

* It provides feedback which guarantee to the client that the developer team understand the problems and issues that needs to be solved and the behavior of the system that are necessary to address those problems.
* It helps to make agreement between user and the developer, and also it helps user to determine whether the stated requirements are fulfilled.
* It helps in determining the requirements of system which further helps in rough estimation of time and cost.

**2.4.1 Functional Requirement**

It is a document which provides list of the operations and task or activities that the system must be able to perform. Functional requirements should include description of data to be entered into the system, operations performed by each interface, system reports or other outputs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Title | Description | Rational | Dependencies |
| FR-001 | Login into system | Admin as well as normal users can login with the credentials provided | Authentication of username and password | FR-002 |
| FR-002 | Registration | Users can register themselves in the system by providing necessary credentials | For being registered into the system. |  |
| FR-003 | View Staffs | Users can view staffs according to different category | To view staffs and also to assist in deleting, updating | FR-001 |
| FR-004 | Update Details | Users can update their credentials after being logged in to the system. | To update details if necessary | FR-001 |
| FR-005 | Delete Account | Users can delete their account and take break from the system if they want | To delete account permanently if needed. | FR-001 |
| FR-006 | View Individual Staffs account | User can check individual staffs accounts that are added in the system and get some information | To view account and edit if necessary | FR-001 |
| FR-007 | Rate staffs | User can give stars to individual staffs by going on their profile | To give rating so that other users can view highest rated and least rated staffs | FR-001 |
| FR-008 | Give Feedback | Users can log into the system, select the staff and give feedback | To give feedback which is the main thing to do on this system | FR-001 |
| FR-009 | View other important notices | Users once logged in the system can view notices posted by the admin. | To view notices that are posted by admin | FR-001 |
| FR-010 | Post Query | Users can post query in community forum | To post query if user faces any problems | FR-001 |
| FR-011 | Reply To posted Queries | Users can reply to others query too. | To reply to questions if some users have specific knowledge on that topic. | FR-001 |

**Functional requirements for admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Title | Description | Rational | Dependencies |
| FR-012 | Login into system | Admin can login to the system by entering username and password | Authentication of Username and password | - |
| FR-013 | Update User details | Admin can update user details if required | To update user details if user provide any error details. | FR-012 |
| FR-014 | Delete User | Admin can also remove user from the system if he/she violates the rules. | To permanently delete user account if any user violates the rules. | FR-012 |
| FR-015 | Add Staffs | Admin can add staffs according to various category | To add staffs so that user can view in their UI. | FR-012 |
| FR-016 | Remove Staffs | Admin can remove staffs if necessary | To remove details from system if required. | FR-012 |
| FR-017 | View Feedback | Admin can view feedbacks posted by various users. | To view feedbacks and take actions. | FR-012 |
| FR-018 | Add notices | Admin can add notice so that the user can view them | To post some important notices. | FR-012 |
| FR-019 | Create Poll | Admin can create poll | To get knowledge on user likes and dislikes. | FR-012 |

**Non-functional requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Title | Description | Rational | Dependencies |
| NF-001 | Security | Data of users as well as the system must be kept secure from theft | Conserving security | NF-003 |
| NF-002 | Availability | System should not be down. It should function 24\*7 | Conserving availability | NF-003 |
| NF-003 | Reliability | System must be reliable. User input and output must be handled properly | Data loss and incorrect data must be avoided | NF-001, NF-002, NF-005 |
| NF-004 | Testability | System must be tested thoroughly | Maintain Testability and correctness of software |  |
| NF-005 | Scalability | It should fit in any conditions. It should function even if users of system grow rapidly and in large number. | For expansion of business | NF-003 |
| NF-006 | Maintainability | System components must be maintained easily. | Solving small problems and fixing bugs. | NF-001 |
| NF-007 | Serviceability | Help service should be provided by developer team if necessary. | Helping users to solve the problems that they are facing. |  |
| NF-008 | Performance | Software should be optimized and less resources should be used. | Helps to boost performance and response time. | NF-001 |
| NF-009 | Recoverability | Data should be recovered in case of accidental deletion and damage. | Ensuring availability | NF-003 |
| NF-010 | Interoperability | Data are allowed for unrestricted sharing between different system. | To share information and resources. |  |

**Mosque prioritization**

In a DSDM project time is fixed. It is very necessary to understand the importance of work that needs to be done in specific order to meet deadlines. Moscow is a tool or technique that helps to understand and manage priorities in which following letters stand for:

* **M**ust Have
* **S**hould Have
* **C**ould Have
* **W**on’t Have this time

Prioritization of requirement of Feedback management system is given below:

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Functional Requirements** | **MOSCOW** |
| FR-001 | Login into system | Must Have |
| FR-002 | Registration | Must Have |
| FR-003 | View Staffs | Must Have |
| FR-004 | Update Details | Should Have |
| FR-005 | Delete Account | Should Have |
| FR-006 | View Individual Staffs account | Should have |
| FR-007 | Rate staffs | Must have |
| FR-008 | Give Feedback | Must have |
| FR-009 | View other important notices | Should have |
| FR-010 | Post Query | Should have |
| FR-011 | Reply To posted Queries | Should have |

MOSCOW prioritization for Admin

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Functional Requirements** | **MOSCOW** |
| FR-012 | Login into system | Must have |
| FR-013 | Update User details | Should have |
| FR-014 | Delete User | Must have |
| FR-015 | Add Staffs | Must have |
| FR-016 | Remove Staffs | Should have |
| FR-017 | View Feedback | Must have |
| FR-018 | Add notices | Must have |
| FR-019 | Create Poll | Should have |

**Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Non-functional requirements** | **MOSCOW** |
| NF-001 | Security | Must have |
| NF-002 | Availability | Should have |
| NF-003 | Reliability | Should have |
| NF-004 | Testability | Must have |
| NF-005 | Scalability | Should have |
| NF-006 | Maintainability | Must have |
| NF-007 | Serviceability | Won’t have |
| NF-008 | Performance | Won’t have |
| NF-009 | Recoverability | Won’t have |
| NF-010 | Interoperability | Won’t have |

**2.4.4. Hardware Software Specification**

For development of any application/software there is use of Hardware and Software. Similarly, for this project use of Hardware and software is done. Following software and hardware are used in order to build the system

**Software**

IDE: Php storm

Xampp, Gitbash are used for programming purpose and GitHub is used for uploading the files in online repository so that we can have their backups.

**Hardware**

Ram used: 6 GB

Processor: Intel Core i7-2640M @2.80Ghz

Graphics: NVIDIA GeForce 520M

**2.5. Use case Diagram**

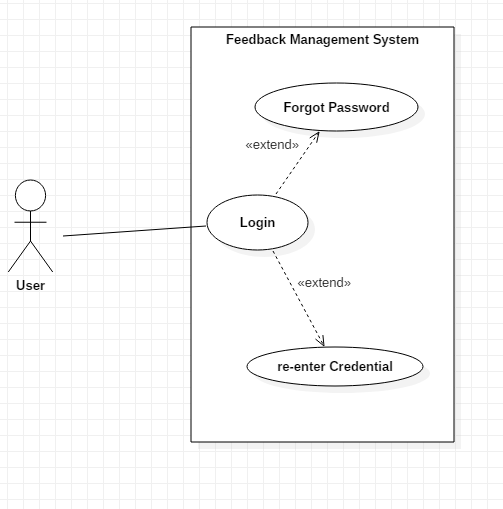
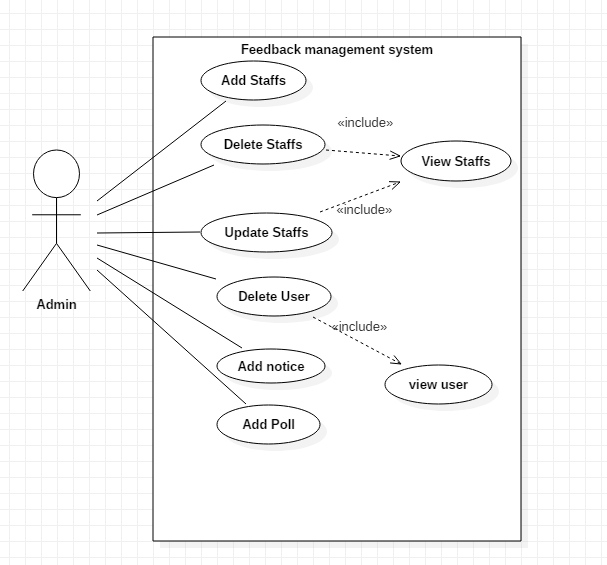
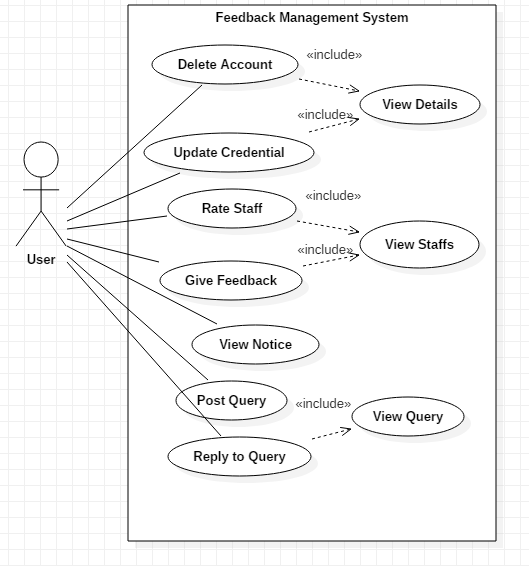


Fig 1: Use case for login

* Users can login to the system.
* They can click on forgot password and change their password.
* They can re enter credentials if the provided details are wrong.



* Admin can add staffs.
* Admin can Delete and update staffs by viewing staffs.
* Admin can view user and delete user if they violate rules.
* Admin can add notice.
* Admin can add poll.



* User can delete and update their credentials and view them.
* User can view, rate and give feedbacks to staffs.
* User can view important notices posted by admin.
* User can post query in community forum.
* User can view Query and reply to query.

**2.6. Initial Class Diagram**

Softwarica college is using paper-based feedback collecting system in order to collect feedbacks. But recently they are facing some problem with the current paper-based system because the existing data aren’t used properly. In order to solve the existing problem an online feedback management system is to be made. The new application should be web based and it should Allow Admin to add Staffs/Users that are in organisation. Users should be able to register on the system, login to the system. Following Functionalities should be provided

* Users should be able to view delete and update their given credentials.
* Admin should be able to add staffs, delete staffs.
* Admin should be able to add notices, delete notices and view notices.
* Users should be able to view staff’s profile and provide feedback.
* Users should be able to cast their votes in poll and reply and post queries.
* User should be able to give reviews and rate staffs.

From above scenario we can classify potential class and methods as

|  |  |
| --- | --- |
| **Potential Classes** | **Potential Methods** |
| Users | View Details |
| Staffs | Update Details |
| Admin | Delete Account |
| Feedback | Add Staffs |
|  | Delete Notices |
|  | View Notices |
|  | View Staffs |
|  | Give Feedback |
|  | Post Query |
|  | Reply Query |
|  | Create Poll |
|  | Vote on Poll |
|  | Give review |
|  | Rate Staffs |

