

Complaint Management System

OOAD tutorial project report submitted in partial fulfilment of the requirements for
the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

VARUN KAPOOR (RA1711003030106)
KHEM PRASAD SHARMA (RA1711003030107)
SHUBHANKAR MEHROTRA (RA1711003030109)

Under the guidance of

Dr. KANIKA GARG
Assistant Professor



at

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Modinagar, Ghaziabad - 201204

(November 2020)

AIM

To identify Use Cases and develop the Use Case model for Complaint Management System.

ABSTRACT

The main idea for developing this project is to replace existing manual complaint system with online service support. In order to provide better complaint response for any kind of corporation problems this system should be effective. This system will increase corporation reputation by providing better services and request feedback from people. As we know people are key for identifying problems.

The main objective of our project is to provide a website for people .They can send their quires to our website through the enquiry form we created any other harmful issues in their local areas.

In present market scope for using this project is high. This Online Complaint Response System for Corporation provides services for peoples through online. This feature can save time. As the information technology is growing widely, developing and maintain this application can be done with a minimum cost. This application can reduce four times the cost compare to the existing manual system.

INTRODUCTION

1.1 OBJECTIVES AND SCOPE OF THE PROJECT

An effective complaints management system is integral to providing quality customer service. It helps to measure customer satisfaction and is a useful source of information and feedback for improving services. Often customers are the first to identify when things are not working properly.

It is web based project this project is advanced and user friendly to the end-user. Any user can able to free member of this website. They can easily download any environment issues and they can directly communicate with welfare officers.

Implementing effective complaints management systems within public sector agencies:

- Improves Companies internal complaints handling
- Reduces recurring complaints
- Raises standards of administrative decision-making
- Improves standards of service to the community

1.2 THEORETICAL BACKGROUND

A good complaints management system is one of the crucial requirements for successful businesses when managing customers' needs and protecting their brand. Through the implementation, assessment, certification and training of a complaints management system we can help you to make big leaps in delivering satisfaction.

A complaint is an expression of dissatisfaction made to an organization, related to

its products, or the complaints handling process itself, where a response or resolution is explicitly implicitly expected." Definition from ISO 10002:2004

It costs an organization at least four times as much to recruit a new customer as to maintain an existing one. Organizations that regularly lose customers struggle to repair their damaged reputations.

In today's competitive environment, product and service innovations are re-defining accepted levels of performance. A good Complaints Management System is one of the crucial requirements for successful businesses when managing customers' needs and protecting their brand.

Effective complaint management is fundamental to the provision of quality service. Complaints systems provide a mechanism for obtaining feedback from customers, resolving disputes and reforming policies and procedures.

Complaints management is a vital component of every decision-making framework and is especially relevant to agencies that have service-oriented roles in the public sector. With increasing expectations from the public, agencies need to respond to complaints in an effective and timely way.

1.3 METHODOLOGY ADOPTED

Prototyping Model has been used for software development according to which a throwaway prototype of the proposed system, based on the currently known requirements, is given to the user so that he has a fair idea about how the proposed system is going to be like. This will help him in deciding the interface, input and output requirements.

It can be easily adjudged that inputs and outputs are big in number, can increase exponentially and may create a big chaos if not restricted properly.

As the user spends some time on the prototype, he will become more precise about his own input output Requirements. This prototype will provide him with an environment analogous to the proposed system's environment.

Due to object oriented support in .NET, various concepts (like reusability, polymorphism, isolation etc.) are already there but for the efficient management of system components, Component based Software Engineering will also be exercised which will help in a resultant library of components, the benefit of which will be reusability and fast development.

Due to lack of hierarchical structure in object oriented approach, there is no meaning of Bottom-up or Top-down testing. Testing will begin from the rudimentary levels of the system and will move towards higher level components, which will be based on design phase rather than coding phase Words.

1.3 LITERATURE REVIEW

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy in company strength. Once these things are satisfied, then next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support.

Data center Security staff utilizing video surveillance, state of the art intrusion detection systems, and other electronic means, when an employee no longer has a business need to access datacenter his privileges to access datacenter should be immediately revoked. All physical and electronic access to data centers by employees should be logged and audited routinely.

Audit tools so that users can easily determine how their data is stored, protected, used, and verify policy enforcement.

Data Provider should also make a contractual commitment to obey local privacy requirements on behalf of their customers, Data-centered policies that are generated when a user provides personal or sensitive information that travels with that information throughout its lifetime to ensure that the information is used only in accordance with the policy.

Data store in database of provider should be redundantly store in multiple physical locations. Data that is generated during running of program on instances is all customer data and therefore provider should not perform backups.

Data Sanitization is the process of removing sensitive information from a storage device. What happens to data stored in a cloud computing environment once it has passed its user's "use by date "What data sanitization practices does the cloud computing service provider propose to implement for redundant and retiring data storage devices as and when these devices are retired or taken out of service.

UML DIAGRAMS

We prepare UML diagrams to understand a system in better and simple way. A single diagram is not enough to cover all aspects of the system. So UML defines various kinds of diagrams to cover most of the aspects of a system.

You can also create your own set of diagrams to meet your requirements. Diagrams are generally made in an incremental and iterative way.

There are two broad categories of diagrams and then are again divided into sub-categories:

- Structural Diagrams
- Behavioural Diagrams

STRUCTURAL DIAGRAMS:

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram which forms the main structure and therefore stable. These static parts are represents by classes, interfaces, objects, components and nodes. The four structural diagrams are:

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

CLASS DIAGRAM:

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations and collaboration. Class diagrams basically represent the object oriented view of a system which is static in nature. Active class is used in a class diagram to represent the concurrency of the system. Class diagram represents the object orientation of a system. So it is generally used for development purpose. This is the most widely used diagram at the time of system construction.

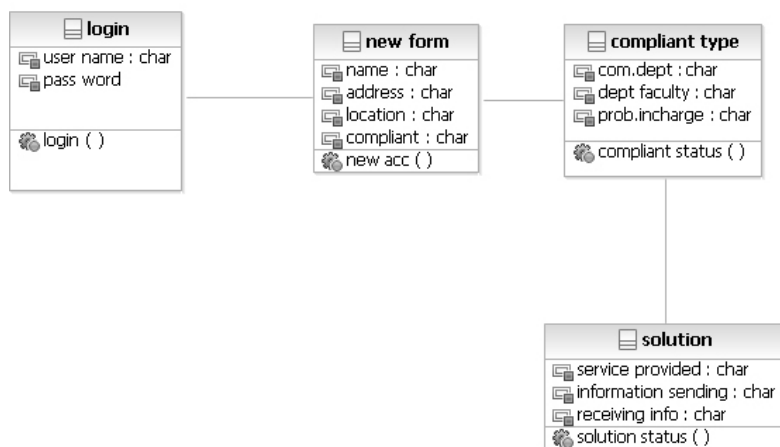


Figure1: Class Diagram

USE CASE DIAGRAM:

Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. So use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

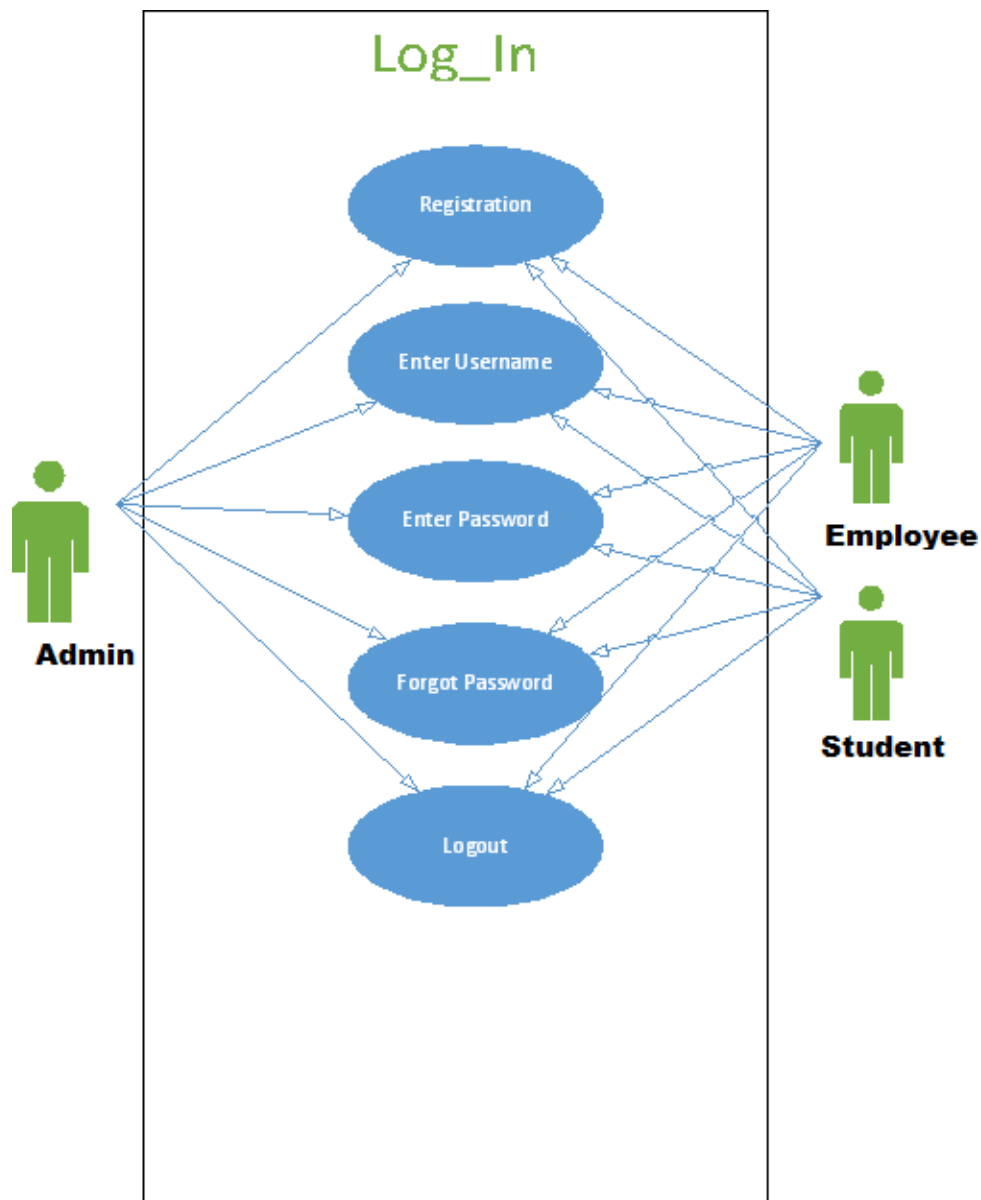
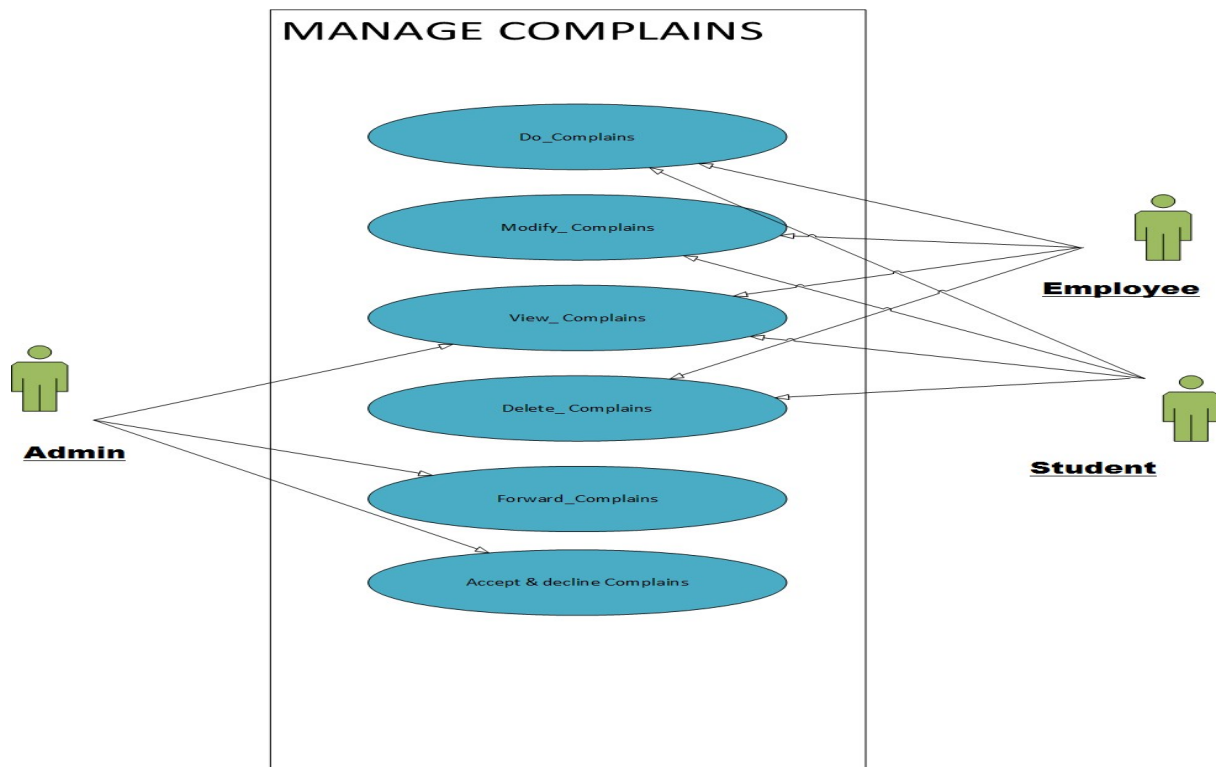
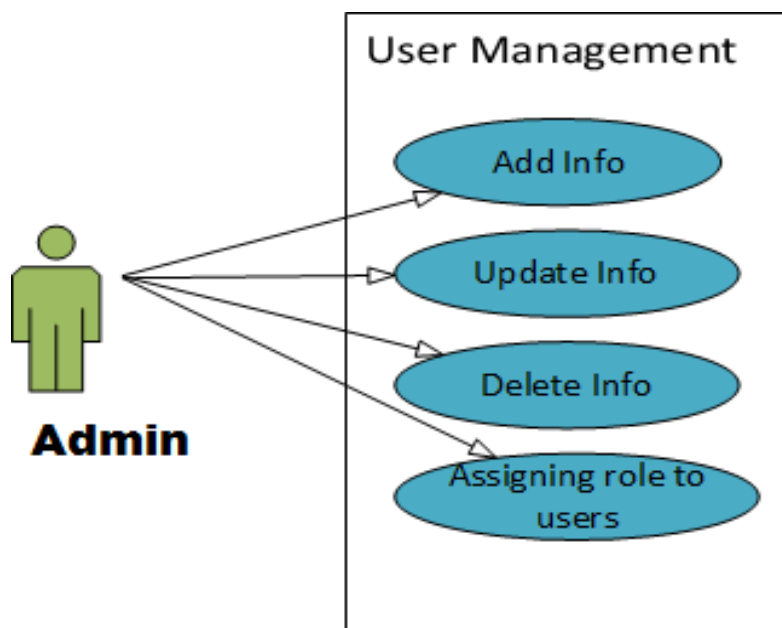


Figure 2: Use Case Diagram

Client Module:



Administrator Module:



SEQUENCE DIAGRAM:

A sequence diagram is an interaction diagram. From the name it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective. So Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

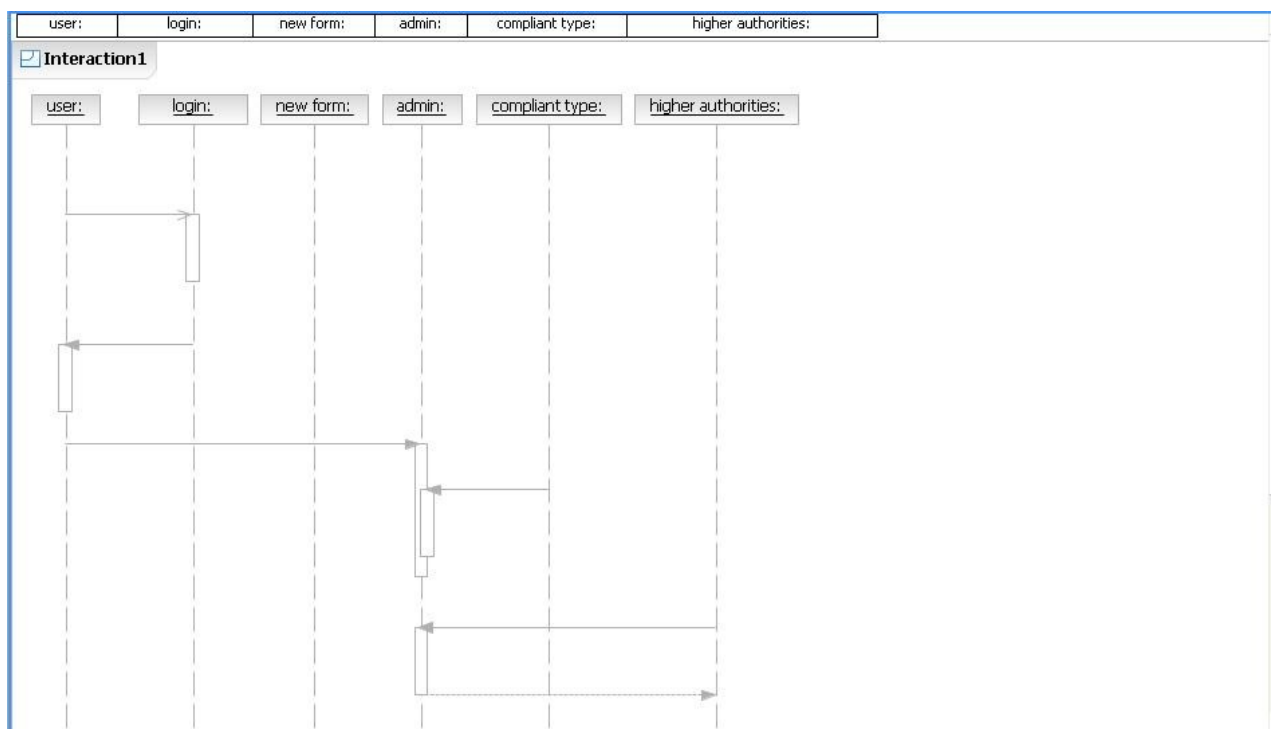


Figure 3: Sequence Diagram

ACTIVITY DIAGRAM:

Activity diagram describes the flow of control in a system. So it consists of activities and links. The flow can be sequential, concurrent or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

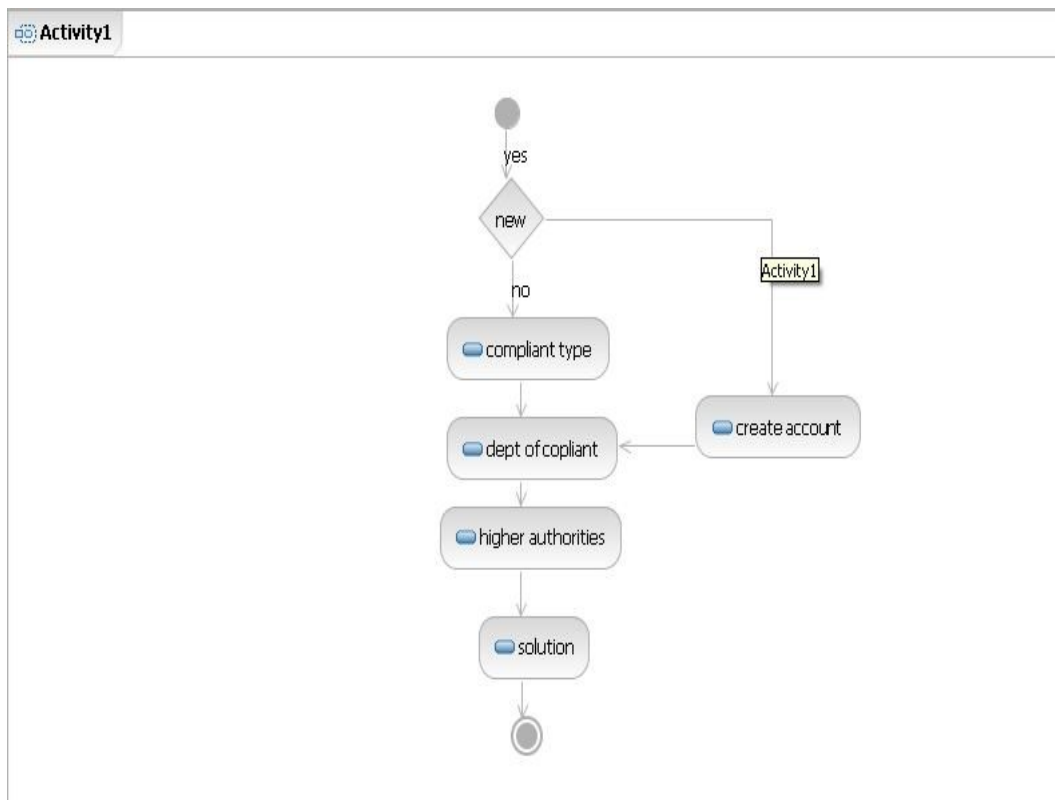
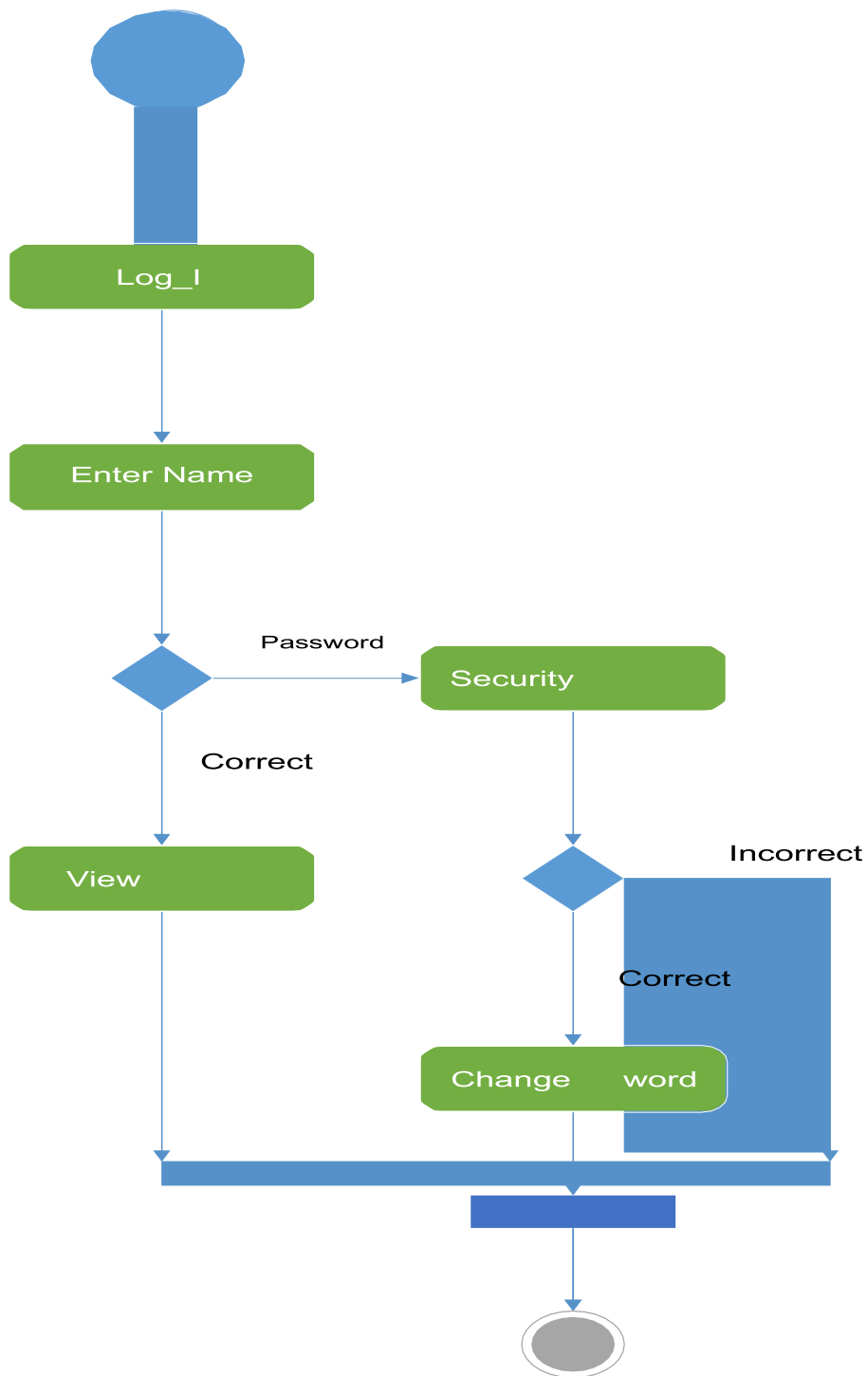
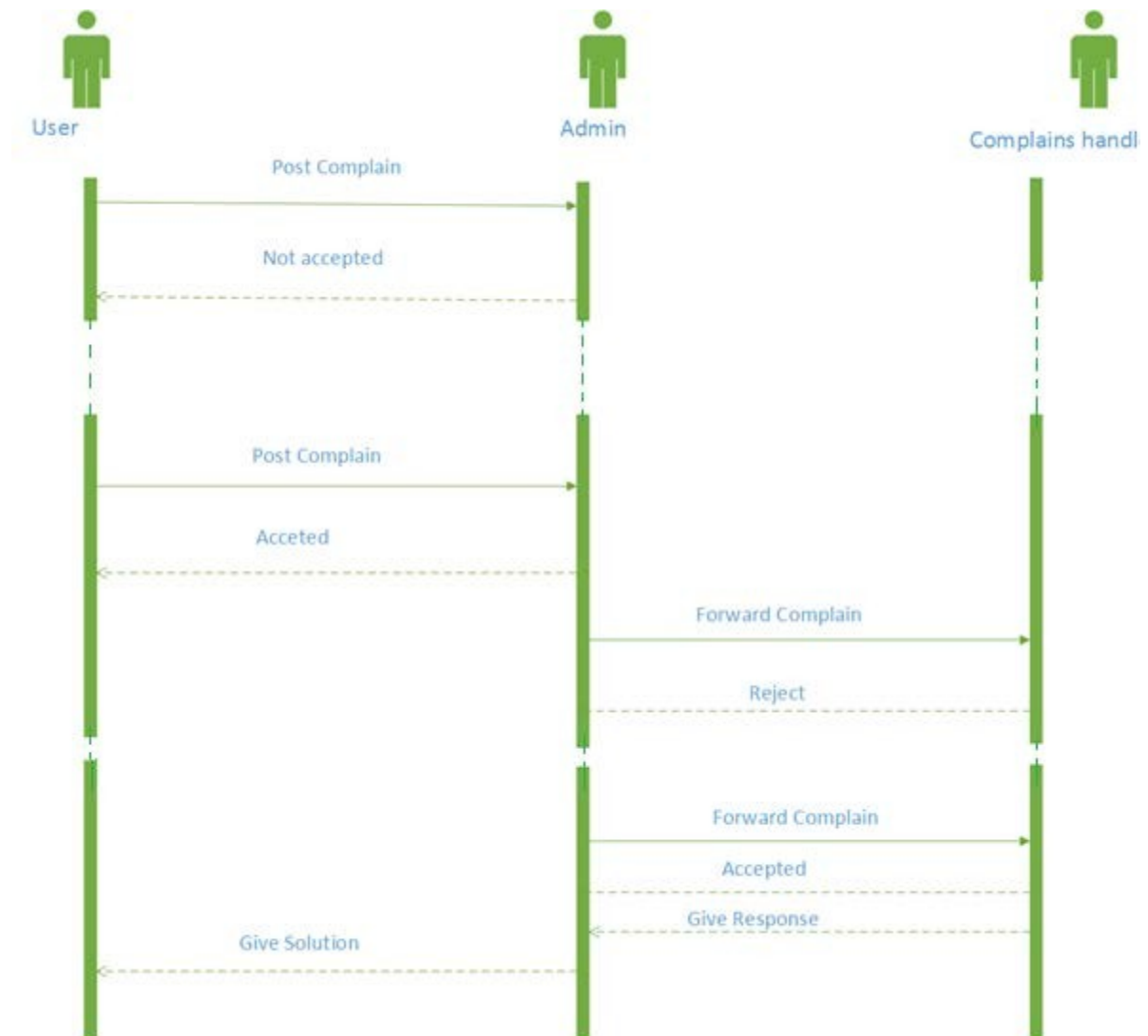


Figure 4: Activity Diagram

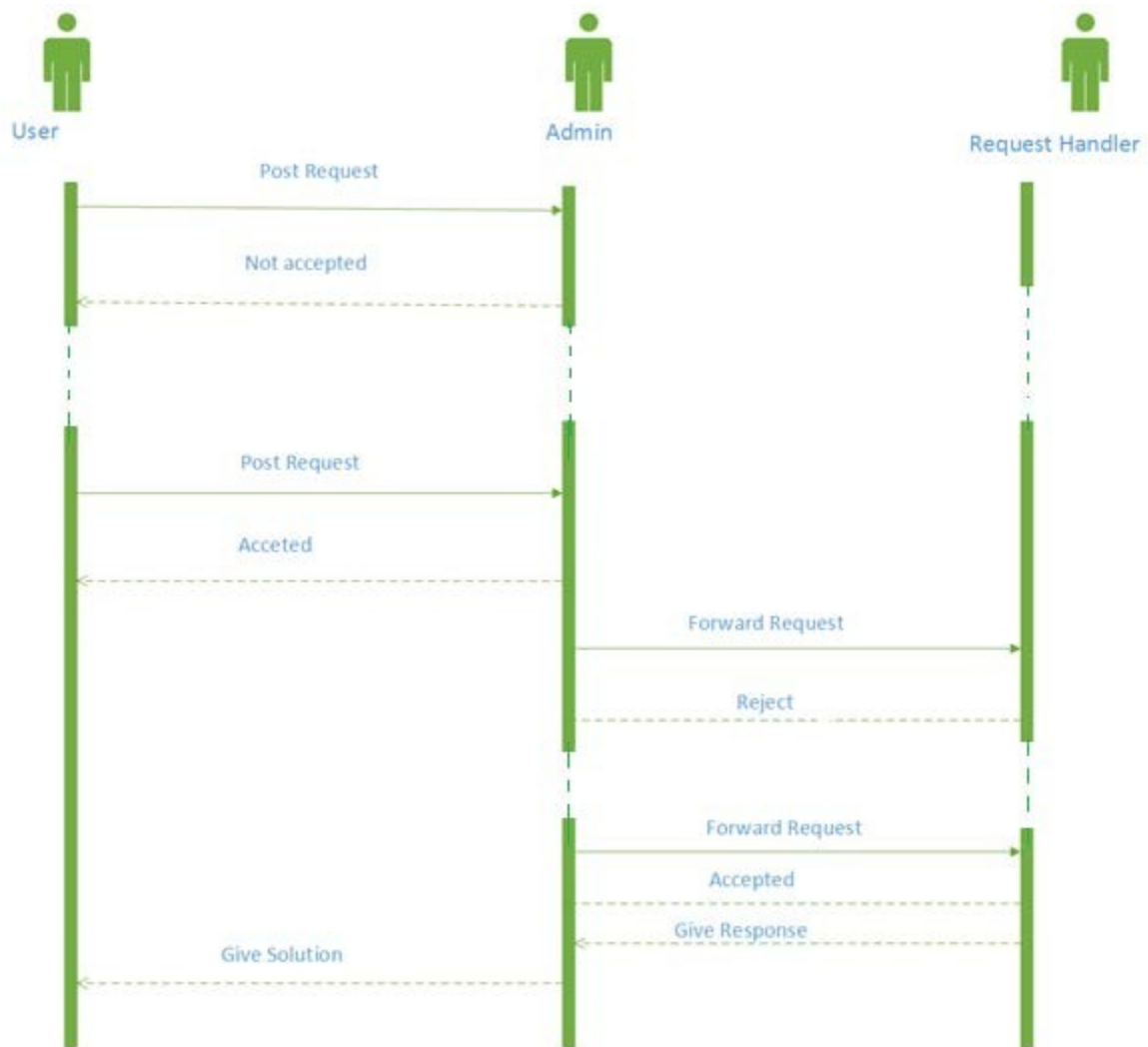
Login :



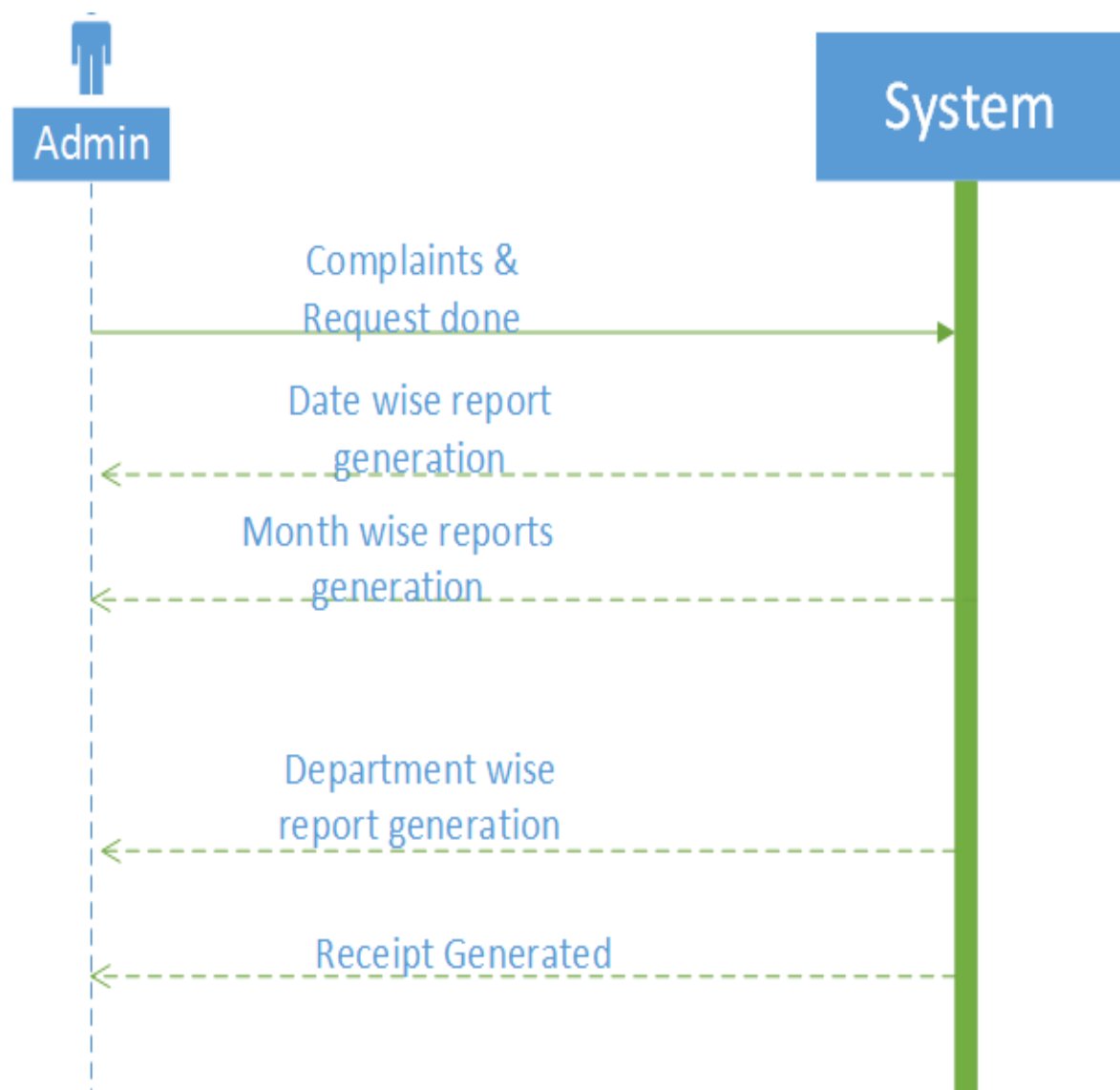
Complain:



Request :

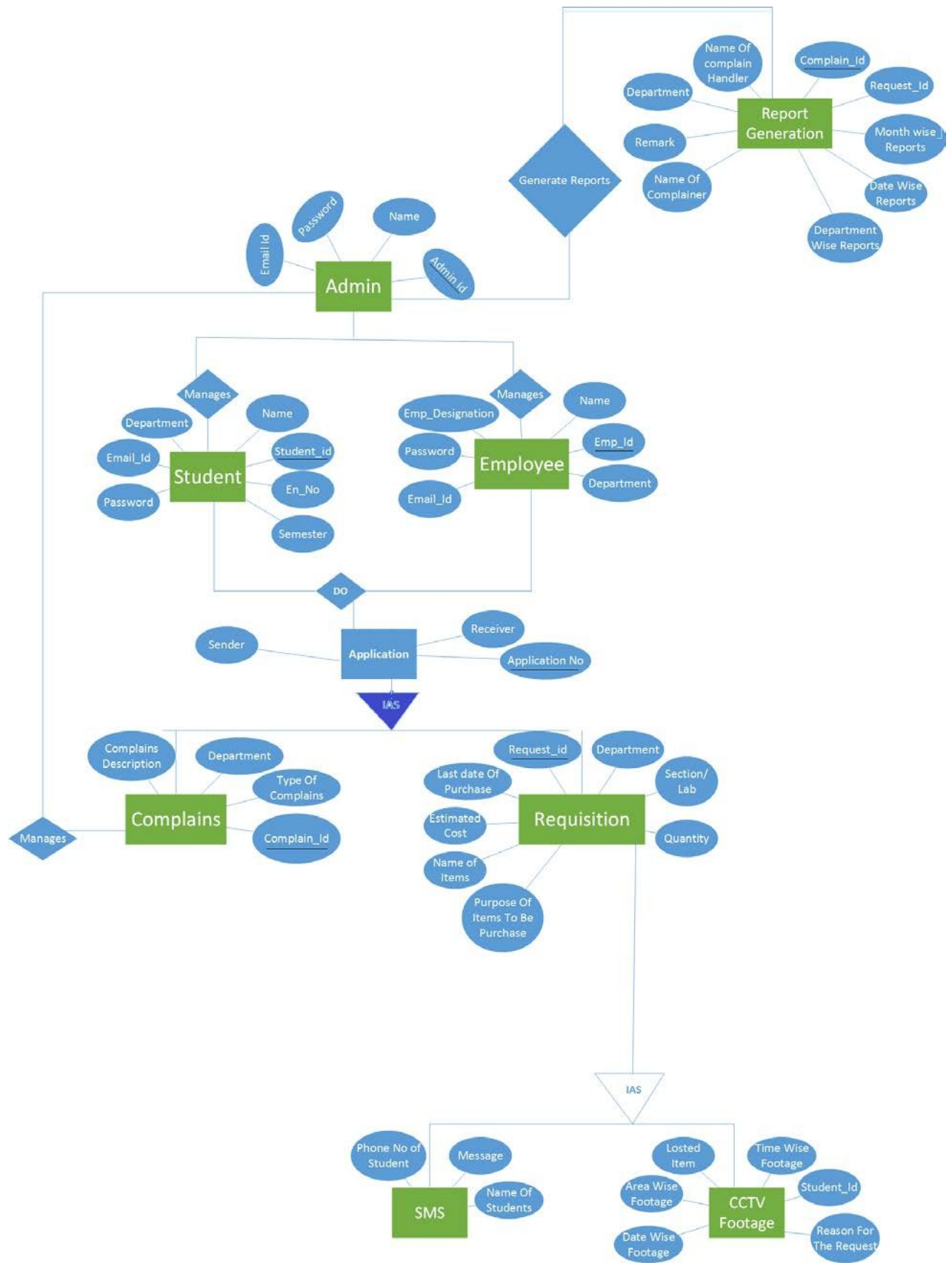


Report Generation:

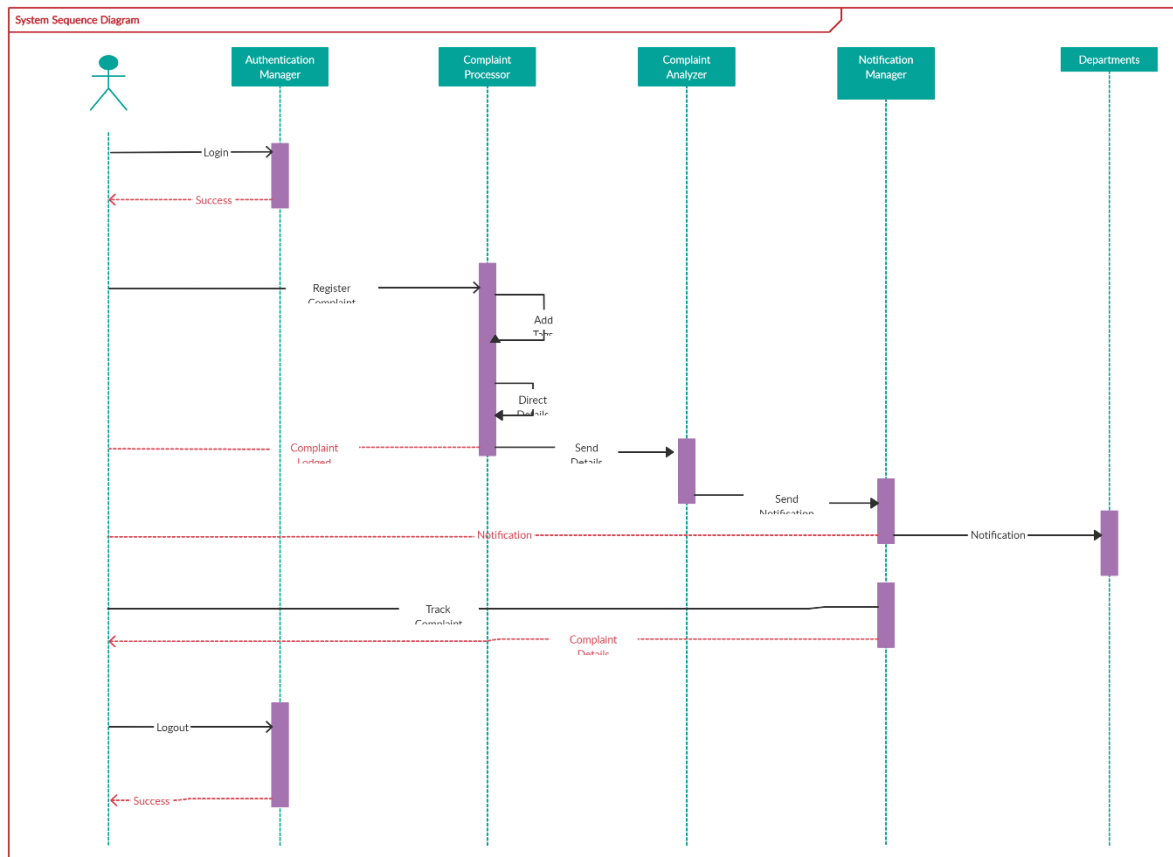


Entity Relationship model:-

The Entity-Relationship data model is based on a perception of a real world, which consists of a set of basic objects called entities and relationships among these objects. An entity is an object that exists and is distinguishable from other objects/entity is an object as a concept meaningful to the organization. An entity set is a set of entities of the same type. A primary key is an attribute which, when taken, allows us to identify uniquely an entity in the entity set.



Communication Diagram:

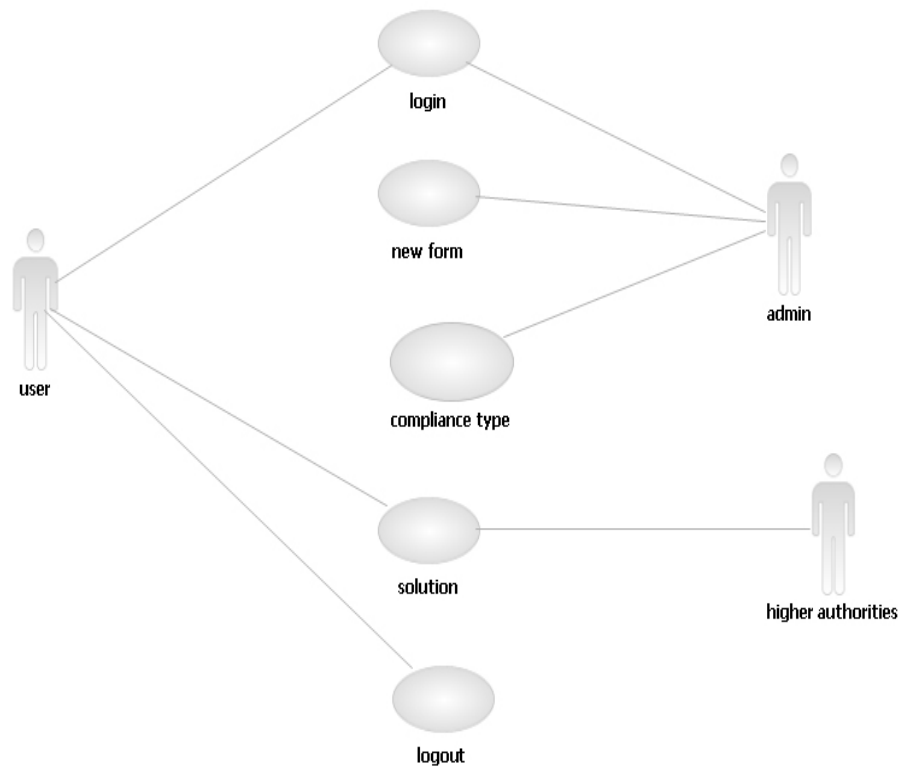


BEHAVIORAL DIAGRAMS:

Any system can have two aspects, static and dynamic. So a model is considered as complete when both the aspects are covered fully. Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams:

- Use case diagram
- Sequence diagram
- Activity diagram



REFERENCES:

- <http://www.projects-forum.com/Thread-student-information-system-Project-report>
- www.developer2blog.com
- <http://aspsnippets.com/Articles/Upload-images-to-folder-and-display-uploaded-images-in-ASPNet-GridView-using-C-and-VBNet.aspx>
- www.stackoverflow.com
- <http://mcaprojects.org/viewproject.php?topic=168>
- www.codeproject.com

- <http://www.mikesdotnetting.com/Article/125/ASP.NET-MVC-Uploading-and-Downloading-Files>
- <http://forums.asp.net/t/1531645.aspx/1>

BIBLIOGRAPHY :

The following books were referred during the analysis and execution phase of the project

- The Complete Reference ASP.NET(3rd Edition) by Matthew MacDonald, Tata McGraw Hill Publications, New Delhi.
- ASP.NET-OREilly.Learning.ASP.NET.3.5.2nd.Edition
- ASP.NET- dot net tutorial for beginners
- Software Engineering (Fifth edition) by Mr. Roger S Pressman.
This book helped us for understanding the concept of software engineering which is needed for making the project's relevant UML Diagrams and other useful notions.