

UNITY OF COMPLAINT ACCEPTING OF MUNICIPAL CORPORATION

A Project Report

Submitted in partial fulfilment of the Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

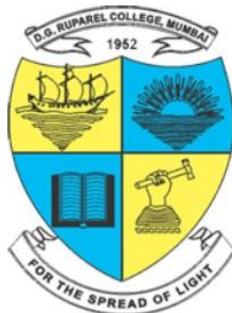
Mr. VIRENDRA PRAKASH MATAL

IT-1124

Under the esteemed guidance of

Mr. MANDAR BHAVE

Course coordinator



DEPARTMENT OF INFORMATION TECHNOLOGY

D. G. RUPAREL COLLEGE OF ARTS, SCIENCE & COMMERCE

(Affiliated to University of Mumbai)

SENAPATI BAPAT MARG, MAHIM, MUMBAI, 400 016

MAHARASHTRA

2020-21

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PNR No.: ...2019016401224104.....

Roll no: IT- 1124

1. Name of the Student

MR. VIRENDRA PRAKASH MATAL

2. Title of the Project

UNITY OF COMPLAINT ACCEPTING OF MUNICIPAL CORPORATION

3. Name of the Guide

MR. MANDAR BHAVE

4. Teaching experience of the Guide: Mr. Mandar Bhave.

5. Is this your first submission?

Yes



No



Signature of the Student

A handwritten signature in blue ink, appearing to read "Virendra Prakash Matal".

Date:

D. G. RUPAREL COLLEGE OF ARTS, SCIENCE & COMMERCE

(Affiliated to University of Mumbai)

SENAPATI BAPAT MARG, MAHIM, MUMBAI, Maharashtra 400 016.

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled, "**UNITY OF COMPLAINT ACCEPTING OF MUNICIPAL CORPORATION**", is bonafied work of **Mr. VIRENDRA MATAL** bearing Seat.No: **IT- 1124** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Place: Mumbai

Date:

Mr. Mandar Bhave.

(Project Guide)

Mr. Mandar
Bhave.

(Course Coordinator)

ABSTRACT

A municipal corporation is a local government in India that administer the urban areas with a population of more than one million. So, providing services like health care, educational institute, transport etc. are easy we have the website of that municipal corporation.

If the user wants to register his complaints, he or she go on particular website of municipal corporation. So, if user want to survey for municipal corporation, he or she wants to go on multiple website of municipal corporation which come under the India. As well as if user wants to add complaints for municipal corporation, he or she want to go on that particular website. There are 27 municipal corporation in Maharashtra, 8 in Gujarat, 13 in Punjab, 10 in Rajasthan, and so on. So, there are around total 241 municipal corporation in India. Hence, they have around 24 websites of municipal corporation.

The project is done to gather all that websites in only one website. So, that user or complainant need not to go on different websites to registering their complaints. User have to just select the area from which he or she is belonging. Once he or she selected the, they will directly redirect to the webpage of that municipal corporation website.

ACKNOWLEDGEMENT

The successful and final outcome of projects require many guidance from many people and I am extremely privileged to have this all along the completion of my project. I am thanks to my project guide Mr. Mandar Bhave Who keep interest in my project work and guided me all along till the completion of my project work. he has always been the source of inspiration to me. I would also like to thank my companions who have helped and encouraged me for the successful completion of my project.

DECLARATION

I hereby declare that the project entitled, "**Unity of complaint accepting of municipal corporation**" done at **D.G. Ruparel College of Arts, Science & Commerce**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university. The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Virendra Matal

Signature: 

TABLE OF CONTENTS

Chapter 1	1
Introduction	1
Background	1
1.2 Objectives.....	3
1.3 Purpose, Scope, & Applicability:.....	5
1.3.1 Purpose:.....	5
1.3.2 Scope:.....	5
1.3.3 Applicability:	6
1.4 Achievements:.....	7
Chapter 2	8
Survey of Technologies	8
2.1 Existing System.....	8
2.2 Proposed System	8
2.3 Requirement Analysis	9
2.3.1 Functional requirement	9
2.3.2 non-Functional requirement	11
2.4 Hardware Requirements.....	12
2.5 Software Requirements	13
2.6 Justification of Platform.....	13
Chapter 3	20
System design	20
3.1 problem definition.....	20
3.2 planning and scheduling.....	20

3.2.1 Gantt chart.....	20
3.3 Data Dictionary	24
3.3.1 Schema Design.....	26
3.3.2 Data integrity and constrain	28
3.4 procedural design	31
3.4.1 logic diagram.....	31
3.5 E-R Diagrams.....	34
3.6 Conceptual model	36
3.6.1 SDLC: Waterfall model	36
3.7 UML diagrams	39
3.7.1 use-case diagram	39
3.7.2 Activity diagram	41
3.7.3 Sequence diagram	42
3.7.4 class diagram.....	47
3.7.5 Object diagram	50
3.7.6 Component diagram.....	51
3.7.7 Deployment diagram.....	53
Chapter 4	56
Implementation and Testing.....	56
4.1 Basic modules	56
4.2 Event table.....	58
4.3 User interface design.....	62
4.4 security issues	73
4.5 Testing Approach	74
4.5.1 Unit Testing.....	75

4.5.2 Integration Testing	78
4.5.3 Test cases design.....	82

LIST OF TABLES

Table 1: Gantt chart.....	23
Table 2: Event table.....	58
Table 3: Security table.....	73
Table 4: Test cases.....	82

LIST OF FIGURES

Figure 1:logic diagram	33
Figure 2:ER diagram	36
Figure 3:waterfall model	39
Figure 4:use case diagram.....	40
Figure 5:activity diagram	42
Figure 6:sequence diagram for user	46
Figure 7:sequence diagram for admin.....	47
Figure 8:class diagram	49
Figure 9:object diagram	51
Figure 10:component diagram	53
Figure 11:deployment diagram	55
Figure 12:start page.....	62
Figure 13:home page.....	63
Figure 14:registration page	64
Figure 15:login page for user	65
Figure 16:admin login	66
Figure 17:commissioner profile	67
Figure 18:department	68
Figure 19:admin dashboard.....	69
Figure 20:complaint registration	70
Figure 21:status interface	71
Figure 22:feedback.....	72
Figure 23: about corporation	73

Chapter 1

Introduction

1.1 Background

A Municipal corporation is a Local government in India that administers urban areas with a population of more than one million. The growing population and urbanization in various cities of India were in needed of a local governing body that can work for providing necessary community services like health care, educational institution, housing, transport etc. By collecting property tax and fixed grant from the State Government.

Unity of complaint accepting of municipal corporation is the project which accept the complaint from complainant or user and pass that complaints to admin. The admin will see that complaint carefully and pass the response to that particular complaint. Also, the admin can take some extra information from the complainant through the communication interface which is also available in this website.

The Unity of complaint accepting of municipal corporation project include selecting area from which complainant or user belongs, by selecting the area he or she will be directly redirect to the home page of Municipal corporation that he or she is belongs. In nowadays, each Municipal corporation have their separate websites but this is all in one Municipal corporation website where all website of Municipal corporation is available with some advance feature. So, there is no need to go on separate Municipal corporation website for registering their complaints. On this website complainant have just select area from which he or she belongs. E.g. If complainant select the Vasai-Virar as the area, he or she will directly redirect to home page of Vasai – Virar Municipal corporation. It also includes the registration page if the complainant is visiting first time to this website, he or she will be first register on the website. In registration details include Name,

Username, password, mobile no., city, state, area and then he or she will login for registering their complaints. Login page include username, password, confirm password. Admin can also login from this website by just selecting the admin login. It includes details like username, password, select Municipal corporation. For registering the complaints, complainant should select ‘register your complaint’ from main menu. Once complainant select this option, he or she will redirect to the registering complaints form. On this page complainant have fill some information and the field with ‘*’ are compulsory to fill. If complainant is not fill this information then he or she will not able to submit their complaints. Complaint registration details include complaints type, ward, name of complainant, date of the complaint, complainant mobile number, upload some photos (if any/optional). Once he or she submit their complaints, they can see the status of the complaint whether it is approved or still pending. If complainant have any doubt related to process or any other fact, he or she can ask to admin through the communication interface. Which enable us to establish the communication between complainant and admin. This will help to clarify the all doubts between complainant and admin.

Once complainant registered their complaints, all that complaints passes to the admin. If admin want more information about to the complaints then he or she will establish the communication with the complainant. When admin receive that complaints, he or she should pass the response to that particular complaint. For that he just has to click on response button. When he or she clicked on response button, he or she can enable to pass the response to the complainant. If there are more complaints received from many users and he or she wants to search for particular complaint, he or she do this by just clicking on search box.

Complainant can also see the other information like information about the municipal commissioner, department, and about the Municipal corporation. If complainant want to give some suggestion to the Municipal corporation, he or she can pass their suggestion through feedback.

This software or website uses JAVA and HTML, CSS, JAVA SCRIPTS as a front end to design an interactive graphical user interface (GUI) and MYSQL server to store its back-end data. This backend data includes accommodation details related complainants or users and admin for effectively managing the software or website. This project is designed to manage all Municipal corporation which comes under the India

1.2 Objectives

1. Making all- in- one Municipal corporation website.
2. Recording information about the complainant or user and admin.
3. Providing easy, smooth interface of Municipal corporation.
4. Establish the communication between complainant or user and admin.
5. Reducing the complexity.
6. Taking complainant or user feedback.
7. Generating alert if admin not give response for any complaint.

This are the various jobs that need to be done in project / website of a Municipal corporation. The project "Unity of complaint accepting of municipal corporation" is aimed to develop to maintain the day to day state of accepting complaints and passing responses.

- To have planned approach towards working:

The working in organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

- To increase reliability:

The reliability of proposed system should be high. The reason for increased reliability of system is that now there would be proper storage of information.

- To reduce redundancy:

No information repeated anywhere in storage. This would assure economic use of storage and consistency in the data stored.

- To easy operation:

The system should be easy to operate and should be such that it can be developed within short period of time and fit in the limited budget of the user.

- To have high accuracy:

The level of accuracy in proposed system will be higher. All operation would be done correctly and ensure that whether the information coming from user is accurate.

- To have immediate retrieval of information:

Any type of information would be available whenever user require

- To have immediate storage of information:

In the manual system there are many problems to store largest amount of information.

1.3 Purpose, Scope, & Applicability:

1.3.1 Purpose:

The main purpose of Software Requirement Specification Document is to describe in the precise manner all the capabilities that will provided by the software application “Unity of complaint accepting of municipal corporation”. It also states various constrains that the system will abide to. This document will lead to clear the vision of software requirement, specifications and capabilities. These are exposed to the development, testing team and end user of the software.

Nowadays, each Municipal corporation have their own websites separately. So if the complainant wants to complaint, he or she need go on that particular Municipal corporation websites which he or she belongs. That's why this software is going to design for making all website in one the website. So, if user want to take survey on many Municipal corporation, he or she don't want to go on different website. They can do their survey through this website and it is easy as well as reduce the effort of user. To do this user needs to select the area that he or she belongs. By the selecting the area, he or she will directly redirect to that Municipal corporation website.

1.3.2 Scope:

1. Information about the complainant is done by just submitting name, username, password, mobile number, state and the area. Whenever the complainant comes up his information is stored freshly.

2. Once complainant registered their complaint, he or she can also add many complaints. Once submitting one complaint, he or she will redirect to ‘get status of complaint’ interface. User can add their complaints from this interface. So, no need to go back at home and click on ‘register your complaint’.
3. As like other website, they are just give’s the contact details for making communication but on this website, communication can be also done using the chat box.
4. No need to go on different website to registering the complaints. In this application user need to just select the area from which user is belonging. Once user selected the area, he will directly redirect to the that municipal corporation website.
5. If complainant submit their complaints but admin does not give the response then one automatic alert will be generating on admin side.
6. Once submitting the complaints, user can check the status of that complaint as well as he can ask some question or doubts to the admin if user have any query.
7. complainant wants to see more about to municipal corporation, he can see detail information in about section as well as he can see departments, profile of municipal commissioner etc.
8. complainant can send their suggestion to the municipal corporation through the feedback section. Which can help municipal commissioner to improve their work.

1.3.3 Applicability:

1. Organizational feasibility: this project will provide considerable assistance and convenience to all the users and excellent information about Municipal corporation.

2. Technical feasibility: the technology that will be used is JAVA with MYSQL as back end. The assessment is based on the outline design of system requirements in terms of input, process, output and procedure. This concerned with specifying equipment and software and hardware that will successfully satisfy the user requirement. The technical need of the system may vary considerably, but might include:

- A) the facility of produce output in a given time
- B) response time under certain conditions.
- C) facility to communication data to distant locations.

3. Schedule feasibility: time evolution is the most important consideration that will be considered in the development process.

1.4 Achievements:

- Managing the time with work.
- We learned how actually design the UML diagram for project.
- Understand that how manage the website.
- Learned how to implement different modules in the website.
- Understood that how to design a good interface for website.

Chapter 2

Survey of Technologies

2.1 Existing System

The existing system of a Municipal corporations is use different website instead of making all Municipal corporation in a one website. This website of a Municipal corporations are so complicated as well as they are incomplete. In some of the Municipal corporations, the information is not written and they leave as the incomplete Municipal corporation website. E.g. Some department information are kept empty, contact details is not available on the website etc. because of no contact details available on website, it is the impossible to establish the communication or interaction with the Municipal corporation. There are total 241 Municipal corporation in India. Hence, they have 241 different websites of Municipal corporations. Some of they have incomplete website, some of they do not have any website of Municipal corporations. In some website, if we try to complaint about some topic which is more important, we could not expect the reply because in some website, there is no option of getting any reply or in some website, a Municipal corporation is replying to that complaint. So, these is become irresponsible example. Complainant didn't got response that means he isn't aware about whether his or her complaints are seen by the Municipal corporation or not, as well as he or she is not aware about whether the complaints are executed or not. So, because of these reasons, user is not able to pass their complaints to the particular Municipal corporation. It will lead in the bad effect on the relationship of user and a Municipal corporation.

2.2 Proposed System

The proposed system is also web-based but it is the more reliable as compare to the existing system. This is because of all website of Municipal corporations are available in only one website. As well as it is not that much complicated as the existing system. Because main goal of this website

is accepting the complaint from complainant and give some response back to the user or complainants. So, the complexity of existing system will be reduced. As we saw in previous section, there is no any contact. But in this project, we are more focused on establish the communication. So, the complainant can pass the complains and once he or she done with the complaints and have some doubt then he can ask some question or query to admin or the Municipal commissioner. Here admin also can pass the response as well as he can ask some questions or the query if he wants some extra information from complainant.

If the response is not pass by the admin then one reminder or alert will generate on admin side. So, the both way communication can be easily established and it is in well managed on this Municipal corporation website. If user or the complainant wanted to complaint about to the particular issue, he or she just need to select the area from which he or she is belongs. Once he selected the area, he will directly redirect to the webpage of Municipal corporation of that area. So, no need go on different website to pass the complaint. All the user or the complainants can use the only one website which is the present system.

2.3 Requirement Analysis

2.3.1 Functional requirement

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

1) Admin

- a) Check complaints: check the complaints which is pass by the complainant
- b) send response: admin can send response to the complainants who pass the complaints

- c) Establish communication: he can ask for some extra information if any information is required. For that he can establish communication with the User.
 - d) Manage user account: admin can manage all user account, because he has access of all user accounts. He can retrieve user details if any requirement.
 - e) Manage department: admin can add or update the department details.
 - f) Login and logout: admin have its own id and password for logging in and logout after performing their task.
- 2) User
- g) Select municipal corporation: user can select municipal corporation by just selecting the area from which he or she belongs.
 - h) Register: to register the complaints, user have to first register on this website. Once he or she register on website, he will easily pass the complaints.
 - i) Login: once user is registered on the website, he or she should to login with the help of username and password.
 - j) Register complaints: user can register their multiple complaints once he or she successfully login.
 - k) Check status of complaints: once user submitted the complaints, he can check the status of the complaints by clicking on get status.
 - l) Ask queries: if user has any doubt or query, he can type it into the chat box and send to the admin.
 - m) Send feedback: user can send feedback once he done with his work.
 - n) Logout: once all the tasks of the user is completed, he can logout from the website.

2.3.2 non-Functional requirement

A non-functional requirement defines the quality attribute of the system. They represent a set of standards used to judge a specific operation of a system. The non-Functional requirement can be as follows:

- planned approach towards working:

The working in organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

- reliability:

The reliability of proposed system should be high. The reason for increased reliability of system is that now there would be proper storage of information.

- reduce redundancy:

No information repeated anywhere in storage. This would assure economic use of storage and consistency in the data stored.

- easy operation:

The system should be easy to operate and should be such that it can be developed within short period of time and fit in the limited budget of the user.

- accuracy:

The level of accuracy in proposed system will be higher. All operation would be done correctly and ensure that whether the information coming from user is accurate.

- retrieval of information:

Any type of information would be available whenever user require

- storage of information:

In the manual system there are many problems to store largest amount of information.

- Maintainability: It is maintained by the system developers for corrective and other heavy problem. The system sill avoids the errors and if any error occurs then the system will provide safe backup.
- Supportability: this system operates in any version of windows operating system. Hence is has more supportability.

2.4 Hardware Requirements

For running any software there is some specification and only if it specifies the specified requirement it will get installed and work as it is designed for.

The most common set of requirements defined by specified operating system or software application is the physical computer resources, also known as hardware. A hardware requirement is often accompanied by a hardware compatibility list, especially in case operating system. An HCL, list tested, compatible and sometimes incompatible hardware device for a particular operating system or application.

This software also has some specification as follows:

Processor: Intel i3 processor or above.

Processor speed: 1.2 GHz or above.

RAM: minimum 4 GB RAM.

Hard disk: 20GB or more.

2.5 Software Requirements

Front end: HTML, CSS, JAVA SCRIPT.

Back end: MYSQL.

Languages: java.

Software/ development environment/ tools: NetBeans IDE 8.2.

Operating system: windows 7 or above.

2.6 Justification of Platform

A programming language is our way of communicating with software. The people who use programming languages are often called programmers or developers. The things we tell software using a programming language could be to make a webpage look a certain way, or to make an object on the page move if the human user takes a certain action.

FRONT END:

So, when a web designer is given an end goal like "create a webpage that has this header, this font, these colours, these pictures, and an animated unicorn walking across the screen when users click on this button," the web designer's job is to take that big idea and break it apart into tiny pieces, and then translate these pieces into instructions that the computer can understand -- including putting all these instructions in the correct order or syntax.

Every page on the web that you visit is built using a sequence of separate instructions, one after another. Our browser (Chrome, Firefox, Safari, and so on) is a big actor in translating code into something we can see on our screens and even interact with. It can be easy to forget that code without a browser is just a text file it's when you put that text file into a *browser* that the magic happens. When we open a web page, our browser fetches the HTML and other programming languages involved and interprets it.

HTML and CSS are actually not technically programming languages; they're just page structure and style information.

- HTML provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and java script.
- CSS is used to control presentation, formatting, and layout.
- Java script is used to control the behavior of different elements.

❖ HTML

HTML is at the core of every web page, regardless the complexity of a site or number of technologies involved. It's an essential skill for any web professional. It's the starting point for anyone learning how to create content for the web.

Markup languages are like *you* just did when you labeled those content types, except they use code to do it specifically, they use HTML tags, also known as "elements." These tags have pretty intuitive names: Header tags, paragraph tags, image tags, and so on. Every web page is made up of a bunch of these HTML tags denoting each type of content on the page. Each type of content on the page is "wrapped" in, i.e. Surrounded by, HTML tags.

Once a tag has been opened, all of the content that follows is assumed to be part of that tag until you "close" the tag. When the paragraph ends, I'd put a closing paragraph tag: </p>. Notice

that closing tags look exactly the same as opening tags, except there is a forward slash after the left angle bracket. Here's an example:

```
<p>This is a paragraph. </p>
```

Using HTML, you can add headings, format paragraphs, control line breaks, make lists, emphasize text, create special characters, insert images, create links, build tables, control some styling, and much more.

ADVANTAGES:

- HTML is Easy to Learn and Use.
- HTML is Free.
- HTML is supported by all Browsers.
- HTML is the friendliest Search Engine.
- HTML is Simple to Edit.
- HTML can Integrate Easily with Other Languages.
- HTML is Lightweight.
- HTML is Basic of all Programming Languages.

❖ CSS

CSS stands for Cascading Style Sheets. This programming language dictates how the HTML elements of a website should actually appear on the frontend of the page. Whereas HTML was the basic structure of our website, CSS is what gives your entire website its style. This language affects the entire mood and tone of a web page, making it an incredibly powerful tool

and an important skill for web developers to learn. It's also what allows websites to adapt to different screen sizes and device types.

ADVANTAGES:

- Easier to maintain and update.
- Greater consistency in design.
- More formatting options.
- Lightweight code.
- Faster download times.
- Search engine optimization benefits.
- Ease of presenting different styles to different viewers.
- Greater accessibility.

❖ JAVASCRIPT

JavaScript is a logic-based programming language that can be used to modify website content and make it behave in different ways in response to a user's actions. Common uses for JavaScript include confirmation boxes, calls-to-action, and adding new identities to existing information. JavaScript is a more complicated language than HTML or CSS, and it wasn't released in beta form until 1995. Nowadays, JavaScript is supported by all modern web browsers and is used on almost every site on the web for more powerful and complex functionality. In short, JavaScript is a programming language that lets web developers design interactive sites.

ADVANTAGES:

- Speed. Client-side JavaScript is very fast because it can be run immediately within the client-side browser.
- Simplicity. JavaScript is relatively simple to learn and implement.
- Popularity.
- Interoperability.
- Server Load.
- Gives the ability to create rich interfaces.

❖ JAVA

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages.

One major advantage of developing software with Java is its portability. Once you have written code for a Java program on a notebook computer, it is very easy to move the code to a mobile device. When the language was invented in 1991 by James Gosling of Sun Microsystems (later acquired by Oracle), the primary goal was to be able to "write once, run anywhere." It's also important to understand that Java is much different from JavaScript. JavaScript does not need to be compiled, while Java code does need to be compiled. Also, JavaScript only runs on web browsers while Java can be run anywhere.

New and improved software development tools are coming to market at a remarkable pace, displacing incumbent products once thought to be indispensable. In light of this continual turnover, Java's longevity is impressive; more than two decades after its creation, Java is still the most popular language for application software development—developers continue to choose it over

languages such as Python, Ruby, PHP, Swift, C++, and others. As a result, Java remains an important requirement for competing in the job market.

ADVANTAGES:

- Java is Simple.
- Java is an Object-Oriented Programming language.
- Java is a secure language.
- Java is cheap and economical to maintain.
- Java is platform-independent.
- Java supports portability feature.
- Java provides Automatic Garbage Collection.
- Java supports Multithreading.

BACK END

❖ MYSQL

MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store. In association with a scripting language such as PHP or Perl (both offered on our hosting accounts) it is possible to create websites which will interact in real-time with a MySQL database to rapidly display categorized and searchable information to a website user.

ADVANTAGES:

-
- Data Security.
- On-Demand Scalability.
- High Performance.
- Round-the-clock Uptime.
- Comprehensive Transactional Support.
- Complete Workflow Control.
- Reduced Total Cost of Ownership.
- The Flexibility of Open Source.

Chapter 3

System design

3.1 problem definition

There is some problem associated with the existing system and they are follows:

1. Some of the existing system are so much complex.
2. The contact details are not given on the website so, it's difficult to establish communication complainants.
3. Some of the information is not given on the municipal corporation website like department details and others.
4. In the existing system, complainant is unable to understand whether his or her complaints are processed or not.
5. Each municipal corporation has different website so, complainant need to search multiple municipal corporation Instead of searching in a one website,
6. User or the complainants cannot add extra information other than complaints form.

3.2 planning and scheduling

3.2.1 Gantt chart

Gantt charts are commonly used for tracking project schedules, and they are especially useful in project management. To put it simply, they illustrate and allow you to know what needs to be done, and when it needs to be done. Gantt charts are also able to show you additional information regarding the different tasks or sections of a project, such as how far have tasks progresses, how a group of tasks might depend on other groups of tasks, how important several tasks are, and resources are being used within a project.

One of the great things about Gantt charts is that they are extremely visual. On the left side of a chart, you will find a list of tasks necessary for a project, and on the top, you will find a

time scale. Here comes the important part: each task is represented by a bar; its length represents the duration of a task. At the same time, each bar is represented by the title of the activity that needs to be completed, along with any relevant information that will help carry it to completion

Advantages of the Gantt chart

Make full use of the advantages of Gantt chart to improve understanding and communication between a project manager and the team members.

1. Visualize Project Information Clearly

Different from project plan documents made by Word, a Gantt chart shows essential information about a project and the tasks involved in a graph. Comparing with tedious words explanation, the cells and bars visualize the progress of each task at a glance, allowing the audience to see the start and finish date, duration, priority and resources allocation of a task at a glance. Whenever you come to the Gantt chart, you can see the complete percentage of tasks and check the project status to keep up with the time. By visualizing all these elements, you can thus realize the feasibility of your schedule and plan if necessary. If something unexpected happens, you are able to take appropriate action in time.

2. Improve Accountability and Communication Efficiency

As has mentioned above, a Gantt chart displays a clear visualization of a project and the tasks involved. Such visibility not only saves the project leader much time to explain a project to his team members, but also engages the team to follow the progress and prepare for the successor task at the right time. Besides, planning a project with Gantt chart makes it easy for stakeholders to understand the timeline and get clarity of the project breakdown immediately, thus establishing their accountability. If everyone gets clear about his role and partners in the project, then misunderstanding and budget overrun can be under control. It enables the team to better coordinate project activities thereby enabling the team to improve overall efficiency.

2. Make Project Planning Practical and Realistic

For most people, it is usually vague and ambiguous when they plan a project in mind. However, when you try to put your project plan into a Gantt chart which requires start and finish date in the calendar, duration, delay possibility, cost and resources allocation, you must quantify the information to make things clear. You will go deeper about the plan and consider more details that influence the priority and practicability. To ensure the practicability and achievement of your project, realistic planning helps to avoid the delays and losses of money that may result. The process of putting a project plan into a Gantt chart forces you to think about your ideas carefully and give a practical and mature plan.

Date Phase \ Date	July	August	September	October	November	December	January	February
Literature survey								
Requirement gathering								
Research								
System design								
Coding								
Testing								
Documentation								

Estimated time: 

Actual time: 

Table 1: Gantt chart

3.3 Data Dictionary

Advantage of Database Management System (DBMS):

Some of them are given as following below.

1. Better Data Transferring:

Database management creates a place where users have an advantage of more and better managed data. Thus, making it possible for end-users to have a quick look and to respond fast to any changes made in their environment.

2. Better Data Security:

As number of users increases data transferring or data sharing rate also increases thus increasing the risk of data security. It is widely used in corporation world where companies invest money, time and effort in large amount to ensure data is secure and is used properly. A Database Management System (DBMS) provide a better platform for data privacy and security policies thus, helping companies to improve Data Security.

3. Better data integration:

Due to Database Management System we have an access to well managed and synchronized form of data thus it makes data handling very easy and gives integrated view of how a particular organization is working and also helps to keep a track on how one segment of the company affects another segment.

4. Minimized Data Inconsistency:

Data inconsistency occurs between files when different versions of the same data appear in different places.

For Example, data inconsistency occurs when a student name is saved as “John Wayne” on

a main computer of school but on teacher registered system same student name is “William J. Wayne”, or when the price of a product is \$86.95 in local system of company and its National sales office system shows the same product price as \$84.95.

So if a database is properly designed then Data inconsistency can be greatly reduced hence minimizing data inconsistency.

5. Faster data Access:

The Data base management system (DBMS) helps to produce quick answers to database queries thus making data accessing faster and more accurate. For example, to read or update the data. For example, end users, when dealing with large amounts of sale data, will have enhanced access to the data, enabling faster sales cycle.

Some queries may be like:

- What is the increase of the sale in last three months?
- What is the bonus given to each of the salespeople in last five months?
- How many customers have credit score of 850 or more?

6. Better decision making:

Due to DBMS now we have Better managed data and Improved data accessing because of which we can generate better quality information hence on this basis better decisions can be made.

Better Data quality improves accuracy, validity and time it takes to read data.

DBMS does not guarantee data quality, it provides a framework to make it is easy to improve data quality.

7. Increased end-user productivity:

The data which is available with the help of combination of tools which transform data into useful information, helps end user to make quick, informative and better decisions that can make difference between success and failure in the global economy.

8. Simple:

Data base management system (DBMS) gives simple and clear logical view of data. Many operations like insertion, deletion or creation of file or data are easy to implement.

3.3.1 Schema Design

Admin details

Full name	Admin_id	Username	Password	Municipal Corporation	Address	Mobile number	Email	Gender	Age

Complainant / user details table

User_id	Name	Username	Password	Mobile number	City	State	Area	Age

Municipal corporation details table

Id	Name	Area	Address	Contact	Email	About	Commissioner name

Complaint details table

Complaint_id	Complainant_name	Type of complaint	Complainant Mobile number	Complaint Date	Response date	status

Feedback details table

Name	F_id	Mobile number	feedback

Department details table

Dept_id	Dept_name
---------	-----------

Relationship between table

Admin table

Attribute	Data type	Allow null
Admin_id	Int	Primary key
name	Varchar	Foreign key

Complainant / user table

Attribute	Data type	Allow null
User_id	Int	Primary key

Municipal corporation details table

Attribute	Data type	Allow null
id	Int	Primary key
Commissioner name(admin)	Varchar	Foreign key

Complaint details table

Attribute	Data type	Allow null
Complaint_id	Int	Primary key
Complainant_name	Varchar	Foreign key

Feedback details table

Attribute	Data type	Allow null
F_id	Int	Primary key

Department details table

Attribute	Data type	Allow null
Dept_id	Int	Primary key

3.3.2 Data integrity and constraint

Admin details

Attribute	Data type	Allows null	Description
Admin_id	Int	Primary key	Unique id of admin
Name	Varchar	Foreign key	Name of the admin
Username	Varchar	Not null	The username of admin.
Password	Varchar	Unique	Password for logging in.
Address	Varchar	Not null	This include address of the admin.
Mobile number	Numeric	Not null	The 10 digit contact number of admin.
Email	Varchar	Unique	Email id of admin.
Gender	Varchar	Not null	Gender of admin.
Age	Int	Not null	This is the age of admin.

Complainant / user details

Attribute	Data type	Allows null	Description
User_id	Int	Primary key	Unique id of user
Name	Varchar	Foreign key	Name of the user
Username	Varchar	Not null	The username of user.

Password	Varchar	Unique	Password for logging in.
Address	Varchar	Not null	This include address of the user.
Mobile number	Numeric	Not null	The 10-digit contact number of user.
Email	Varchar	Unique	Email id of user.
Gender	Varchar	Not null	Gender of user.
Age	Int	Not null	This is the age of admin.

Municipal corporation details table

Attribute	Data type	Allows null	Description
Id	Int	Primary key	Unique id of Municipal corporation
Name	Varchar	Foreign key	Name of the Municipal corporation
Area	Varchar	Not null	Area which municipal corporation belongs.
Address	Varchar	Not null	Complete address of municipal corporation.
Contact	Numeric	Not null	Contact details.
About	Varchar	Not null	Details about the municipal corporation.
Commissioner name	Varchar	Foreign key	Name of Commissioner.

Complaint details table

Attribute	Data type	Allows null	Description
Complaint_Id	Int	Primary key	Unique id of complaint
Complainant_Name	Varchar	Foreign key	Name of the complainant.
Type of complaint	Varchar	Not null	Specifies complaint type.
Complainant Mobile number	Numeric	Not null	Contact details of user.
Complaint date	Varchar	Not null	Date of registering complaint.
Response date	Varchar	Not null	Date on which response pass to complaint.
Status	Char	Not null	Status of the complaint.

Feedback details table

Attribute	Data type	Allows null	Description
F_Id	Int	Primary key	Unique id of each feedback.
Complainant_Name	Varchar	Foreign key	Name of the complainant.
Mobile number	Numeric	Not null	Contact details of user.

Feedback	Varchar	Not null	Suggestion pass by the user
----------	---------	----------	-----------------------------

Department details table

Attribute	Data type	Allows null	Description
Dept_Id	Int	Primary key	Unique id of each department
Dept_Name	Varchar	Not null	Name of the department.

3.4 procedural design

3.4.1 logic diagram

The logical architecture is considered a type of structural design that is used to provide a detailed description of the system without defining the system technology or environment. It includes all the information related logically and detailed description without using technical terms in it. It means it can be considered a diagram used to define the relation between the software components. The main purpose of designing logical architecture is to plan for the system and use it for communication before designing and implementing the system. And by this diagram, the user can get an overview of the system.

When the logical architecture diagram is created, it developed with more focused and have more concerned compare to physical architecture as this is the first step for designing the system, and if there is any relative error present in the system, it can create problems in implementing the system in the future. For a system, there is one common tier architecture is defined that contains three tiers that are used to defining the response-request cycle.

- The user uses the presentation tier to generate the request.
- The request is immediately transferred to the application tier.

- In the application tier, further processing is done, and the data is fetched from the data tier that is required in manipulation.
- The response is created by the application tier and transferred to the presentation tier.
- The response is transferred to the user using the presentation tier.

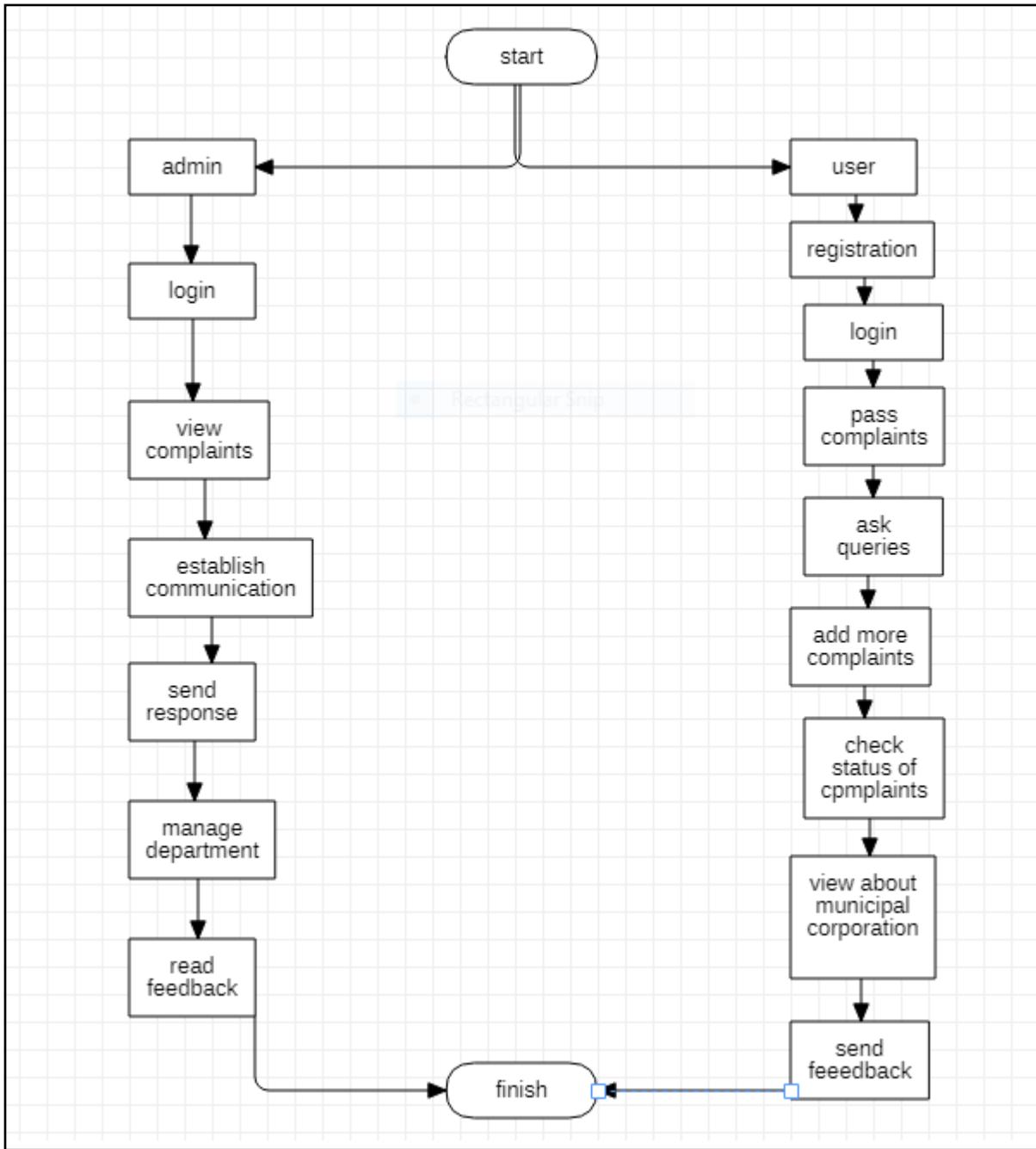


Figure 1:logic diagram

3.5 E-R Diagrams

An Entity-Relationship (ER) diagram expresses the logical structure of a database by showing the relationships between entity sets in a system. ER diagrams are widely used because they help structure information stored in a database and can be directly translated into tables.

Components

The following are the major components of an ER diagram:

- Entity

An entity is a real-world object. It may or may not have a physical existence. Examples include person John *and* his bank account.

- Entity Set

An entity set is a collection of similar entities. Examples include a group of people or a bank. An entity set is represented by a rectangle in an ER diagram.

- Attribute

An attribute is the property of an entity. Examples include name, age, etc. An attribute is represented by an oval in an ER diagram.

- Relationship

A relationship is an association between two or more entity sets. A relationship is represented by a diamond in an ER diagram.

- Cardinality

Cardinality defines the attributes of a relationship using numbers. It includes:

1. One-to-one relationship: An entity in entity set “a” is associated with, at most, one entity in entity set “B”.
2. One-to-many relationship: An entity “a” is associated with more than one entity in “B”.
3. Many-to-one relationship: Many entities in “a” are associated with an entity in “b”.
4. Many-to-many relationship: Many entities in “a” are associated with many entities in “b”.

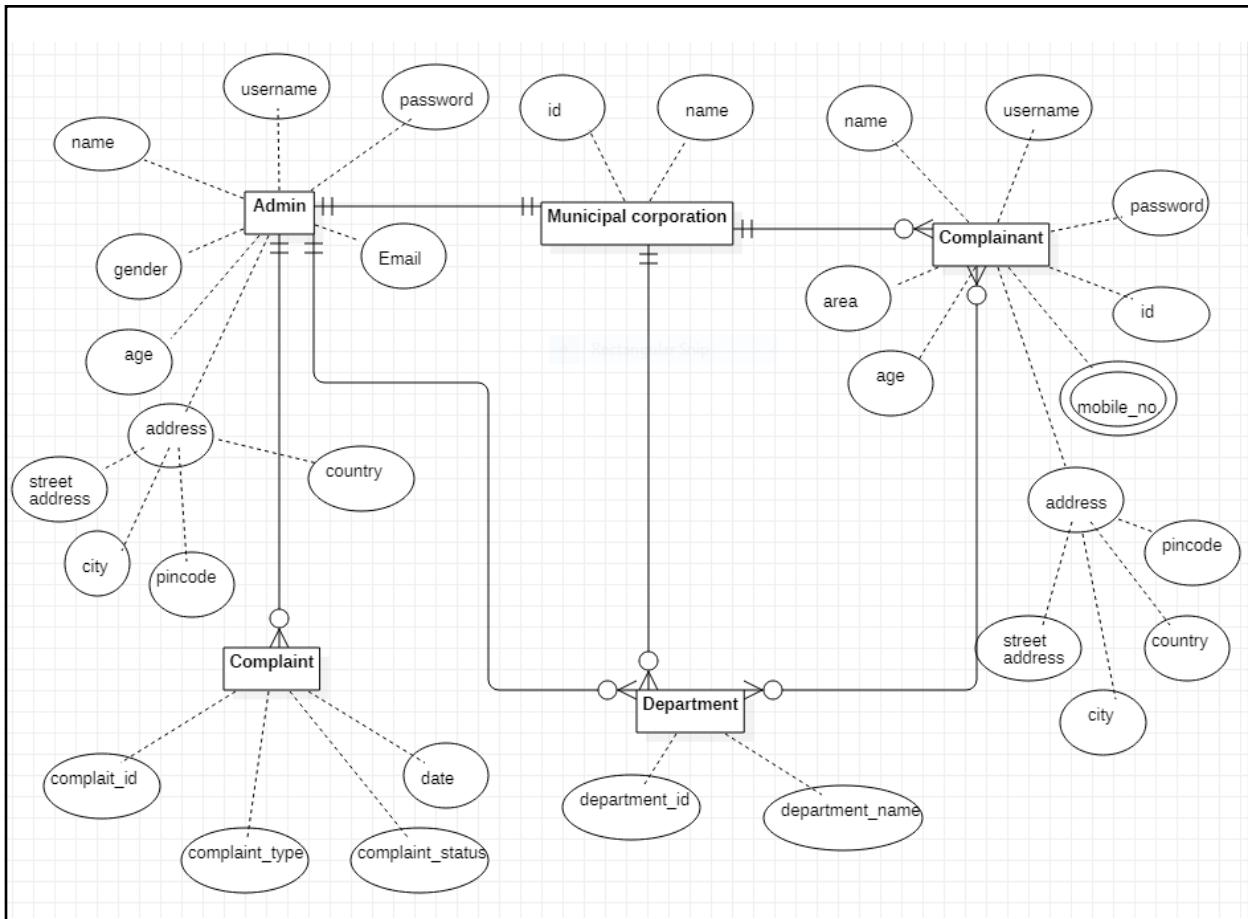


Figure 2:ER diagram

3.6 Conceptual model

3.6.1 SDLC: Waterfall model

For developing this system, the approach that we are using is waterfall model. The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows –

- No working software is produced until late during the life cycle.

- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Diagram of Waterfall Model

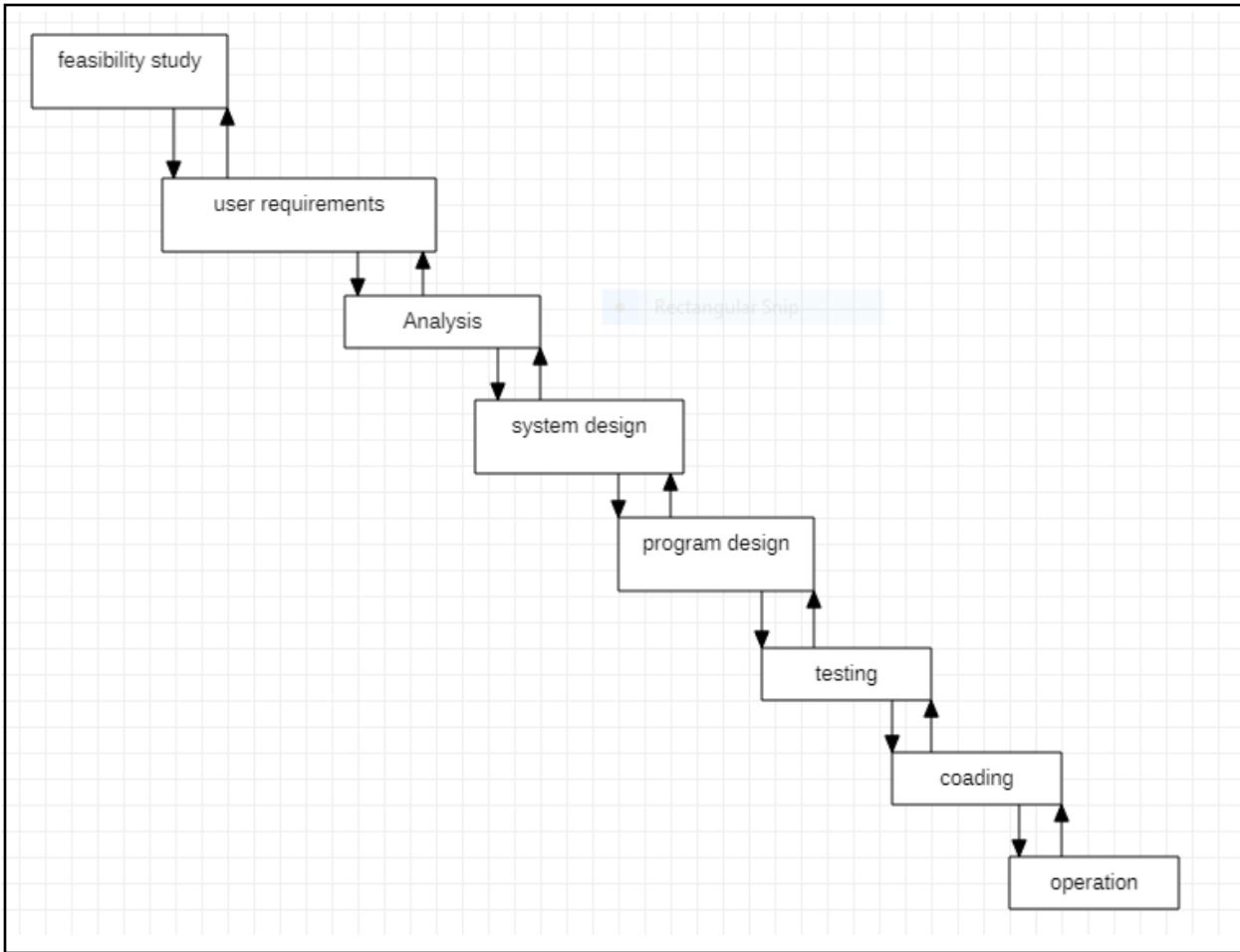


Figure 3:waterfall model

3.7 UML diagrams

3.7.1 use-case diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

A use case diagram is usually simple. It does not show the detail of the use cases:

- It only summarizes some of the relationships between use cases, actors, and systems.
- It does not show the order in which steps are performed to achieve the goals of each use case.

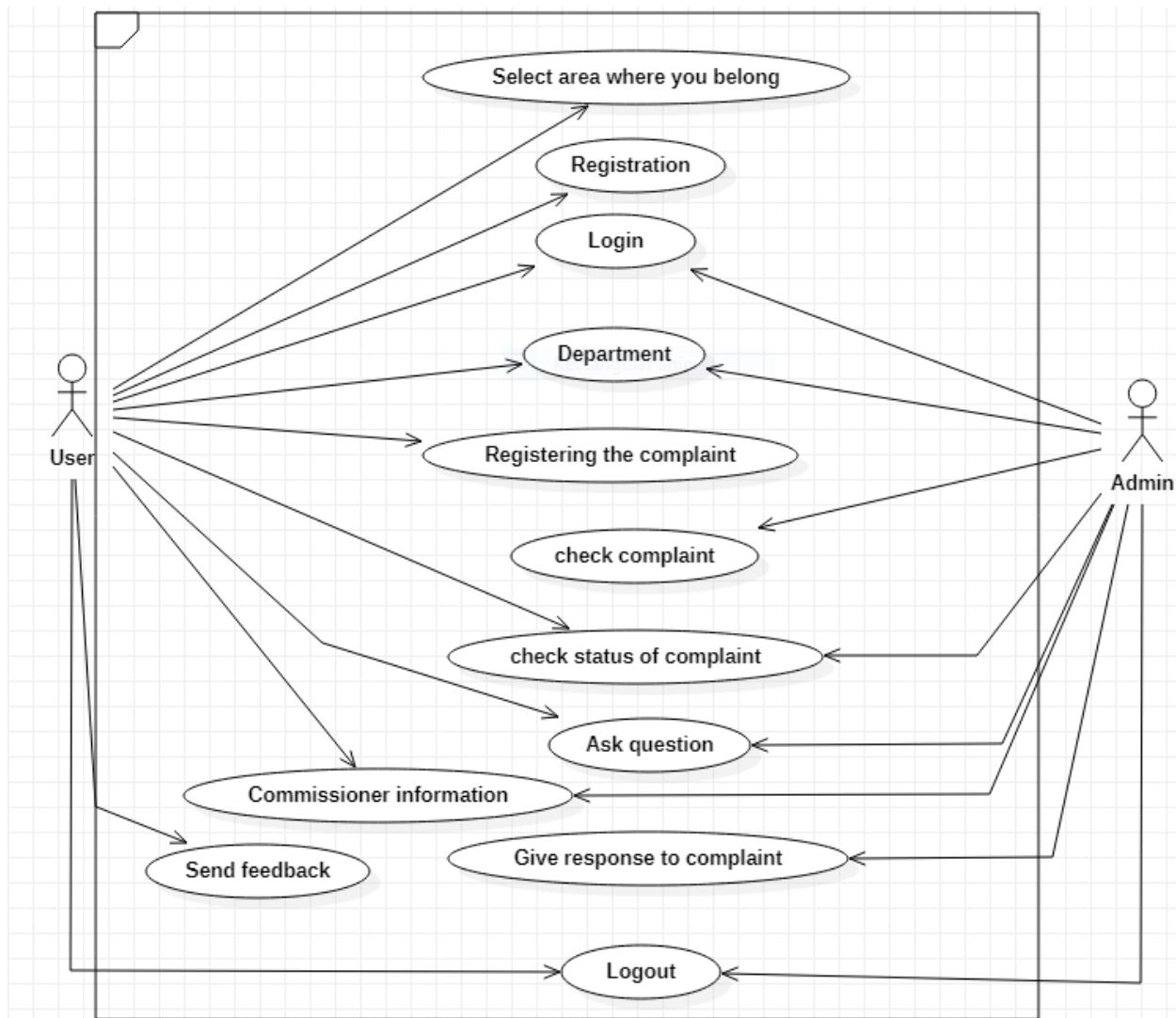


Figure 4:use case diagram

3.7.2 Activity diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

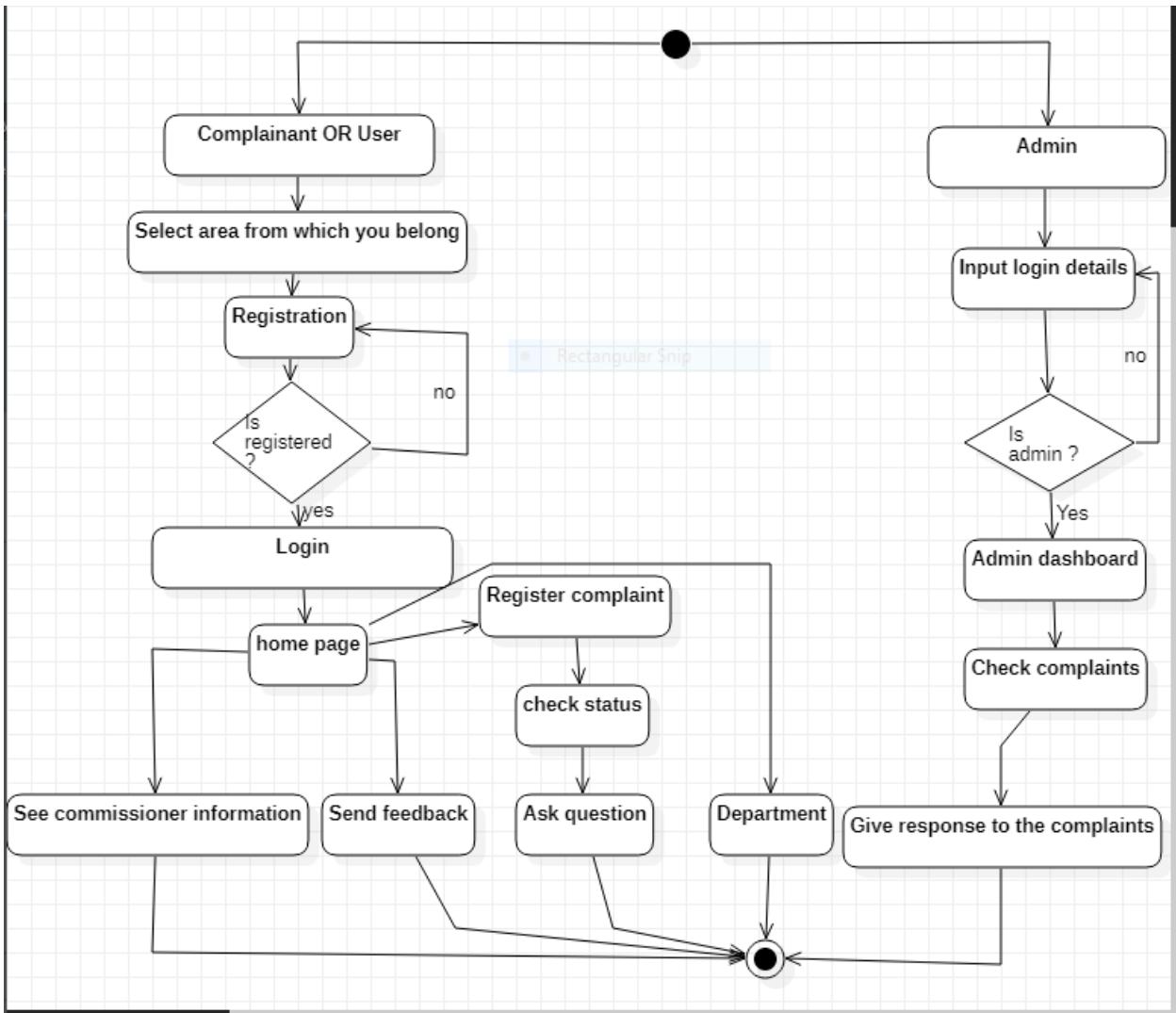


Figure 5:activity diagram

3.7.3 Sequence diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

Sequence Diagram Notation

Notation description
Actor <ul style="list-style-type: none">• a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data)• external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject).• represent roles played by human users, external hardware, or other subjects.
Lifeline <ul style="list-style-type: none">• A lifeline represents an individual participant in the Interaction.
Activations <ul style="list-style-type: none">• A thin rectangle on a lifeline) represents the period during which an element is performing an operation.• The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively
Call Message

- A message defines a particular communication between Lifelines of an Interaction.
- Call message is a kind of message that represents an invocation of operation of target lifeline.

Return Message

- A message defines a particular communication between Lifelines of an Interaction.
- Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.

Self-Message

- A message defines a particular communication between Lifelines of an Interaction.
- Self-message is a kind of message that represents the invocation of message of the same lifeline.

Recursive Message

- A message defines a particular communication between Lifelines of an Interaction.
- Recursive message is a kind of message that represents the invocation of message of the same lifeline. Its target points to an activation on top of the activation where the message was invoked from.

Create Message

- A message defines a particular communication between Lifelines of an Interaction.
- Create message is a kind of message that represents the instantiation of (target) lifeline.

Destroy Message

- A message defines a particular communication between Lifelines of an Interaction.
- Destroy message is a kind of message that represents the request of destroying the lifecycle of target lifeline.

Duration Message

- A message defines a particular communication between Lifelines of an Interaction.
- Duration message shows the distance between two-time instants for a message invocation.

Sequence diagram for user

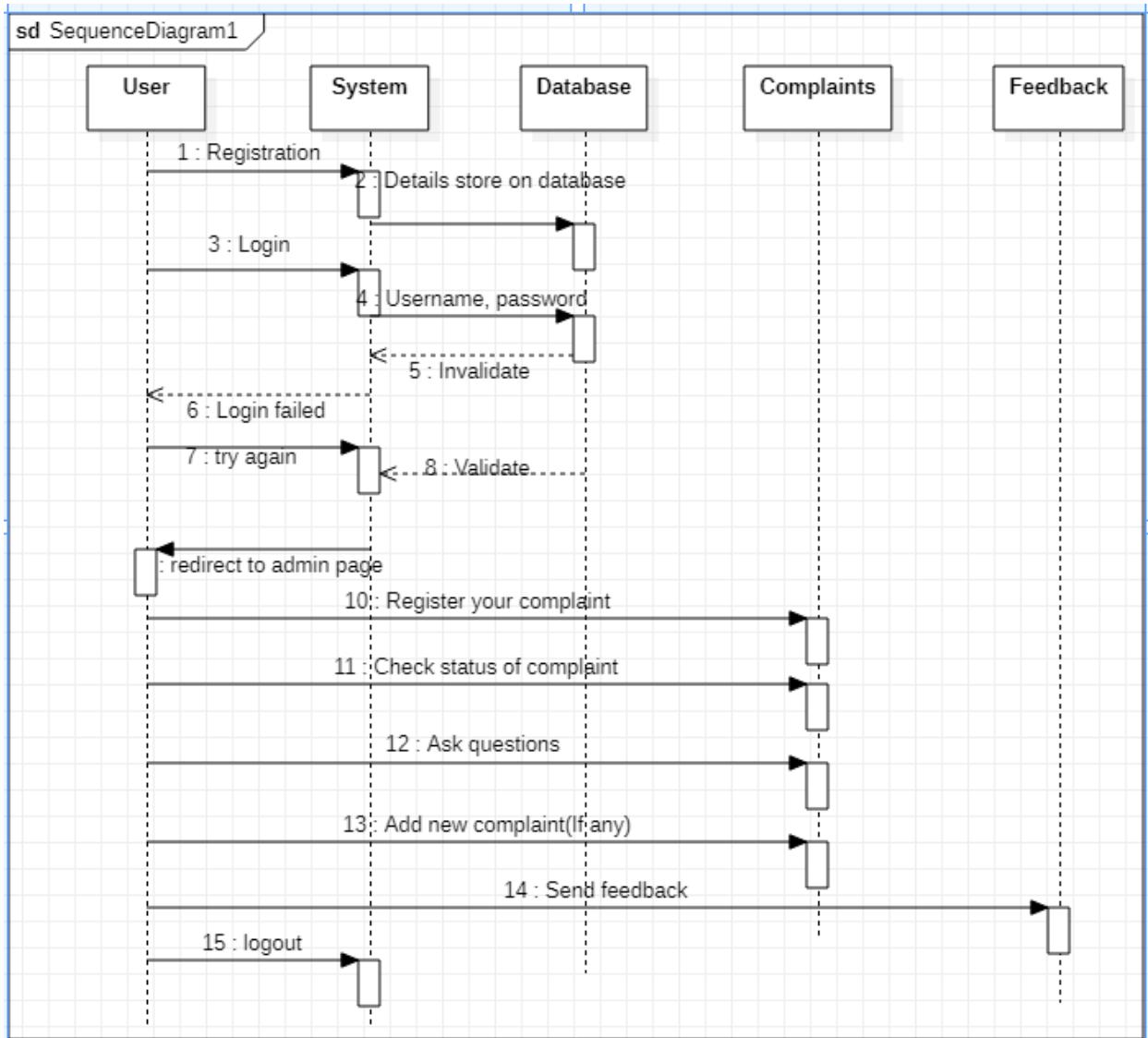


Figure 6:sequence diagram for user

Sequence diagram for admin

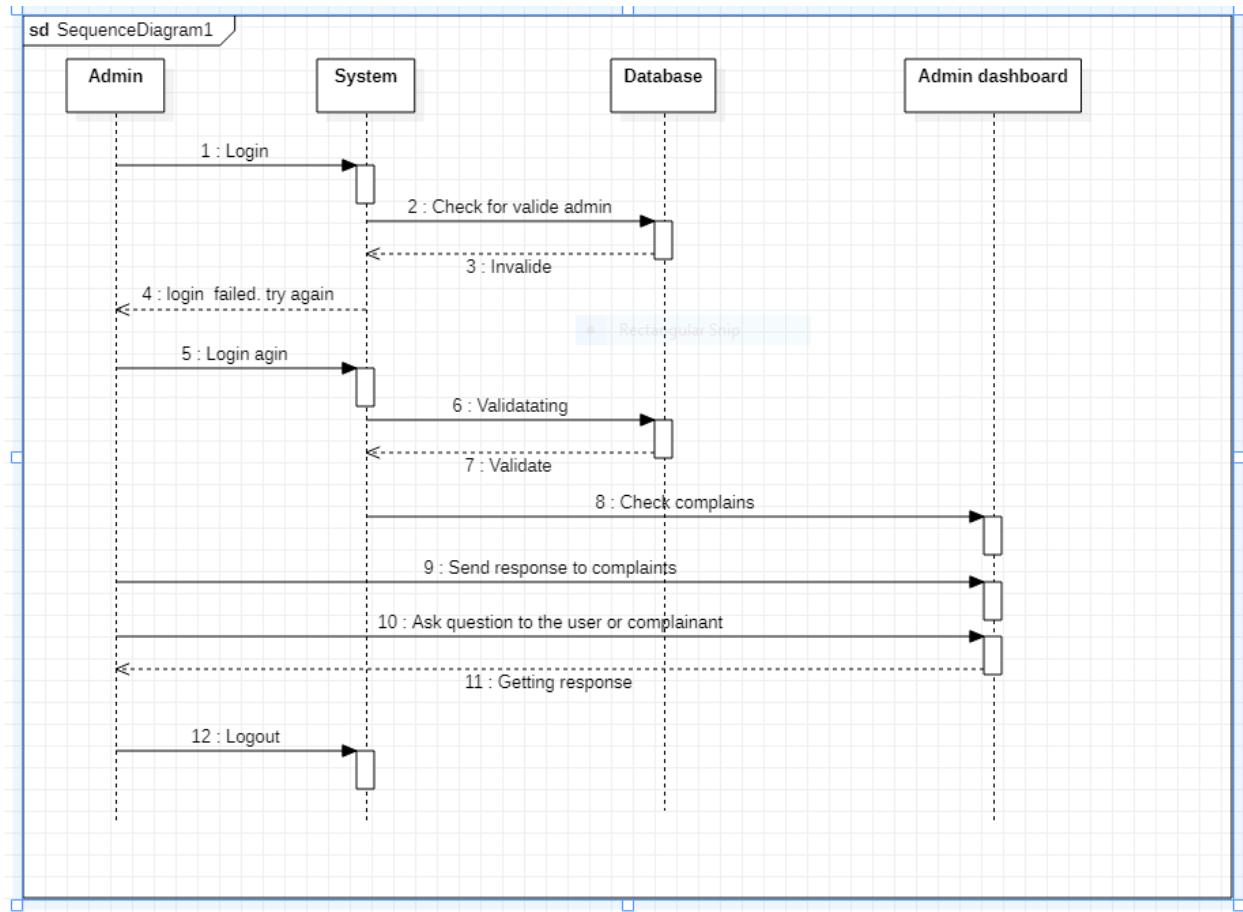


Figure 7:sequence diagram for admin

3.7.4 class diagram

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagrams

1. Shows static structure of classifiers in a system
2. Diagram provides a basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too

4. Business Analysts can use class diagrams to model systems from a business perspective

A UML class diagram is made up of:

- A set of classes and
- A set of relationships between classes

What is a Class

A description of a group of objects all with similar roles in the system, which consists of:

- Structural features (attributes) define what objects of the class "know"
 - Represent the state of an object of the class
 - Are descriptions of the structural or static features of a class
- Behavioural features (operations) define what objects of the class "can do"
 - Define the way in which objects may interact
 - Operations are descriptions of behavioural or dynamic features of a class

Class Notation

A class notation consists of three parts:

1. Class Name

- The name of the class appears in the first partition.

2. Class Attributes

- Attributes are shown in the second partition.
- The attribute type is shown after the colon.
- Attributes map onto member variables (data members) in code.

3. Class Operations (Methods)

- Operations are shown in the third partition. They are servicing the class provides.

- The return type of a method is shown after the colon at the end of the method signature.
- The return type of method parameters is shown after the colon following the parameter name.
- Operations map onto class methods in code

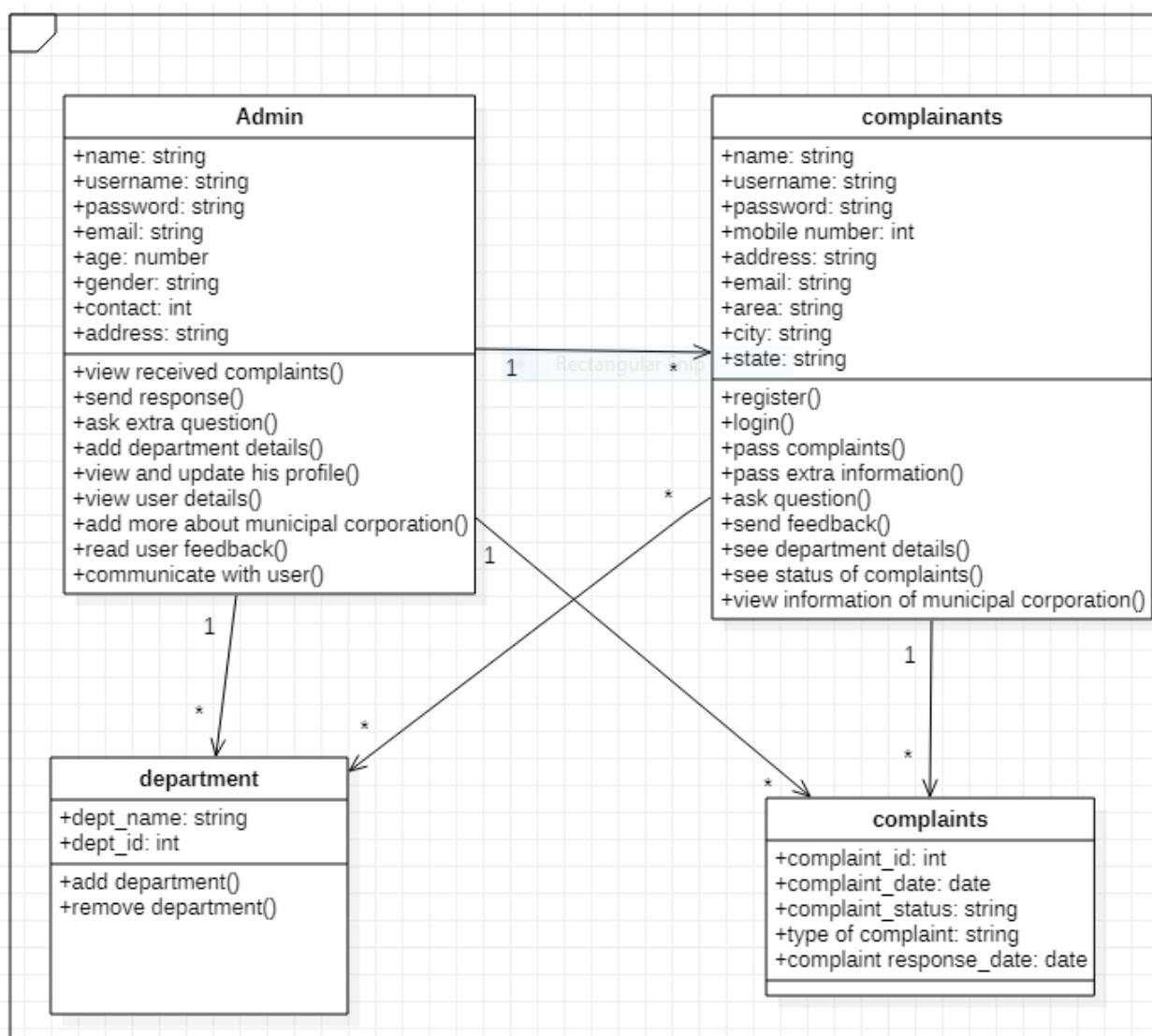


Figure 8:class diagram

3.7.5 Object diagram

Object is an instance of a class in a particular moment in runtime that can have its own state and data values. Likewise a static UML object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships which may be considered a special case of a class diagram or a communication diagram.

Purpose of Object Diagram

The use of object diagrams is fairly limited, mainly to show examples of data structures.

- During the analysis phase of a project, you might create a class diagram to describe the structure of a system and then create a set of object diagrams as test cases to verify the accuracy and completeness of the class diagram.
- Before you create a class diagram, you might create an object diagram to discover facts about specific model elements and their links, or to illustrate specific examples of the classifiers that are required.
-

Basic Object Diagram Symbols and Notations

Object Names:

- Every object is actually symbolized like a rectangle, that offers the name from the object and its class underlined as well as divided with a colon.

Object Attributes:

- Similar to classes, you are able to list object attributes inside a separate compartment. However, unlike classes, object attributes should have values assigned for them.

Links:

- Links tend to be instances associated with associations. You can draw a link while using the lines utilized in class diagrams.

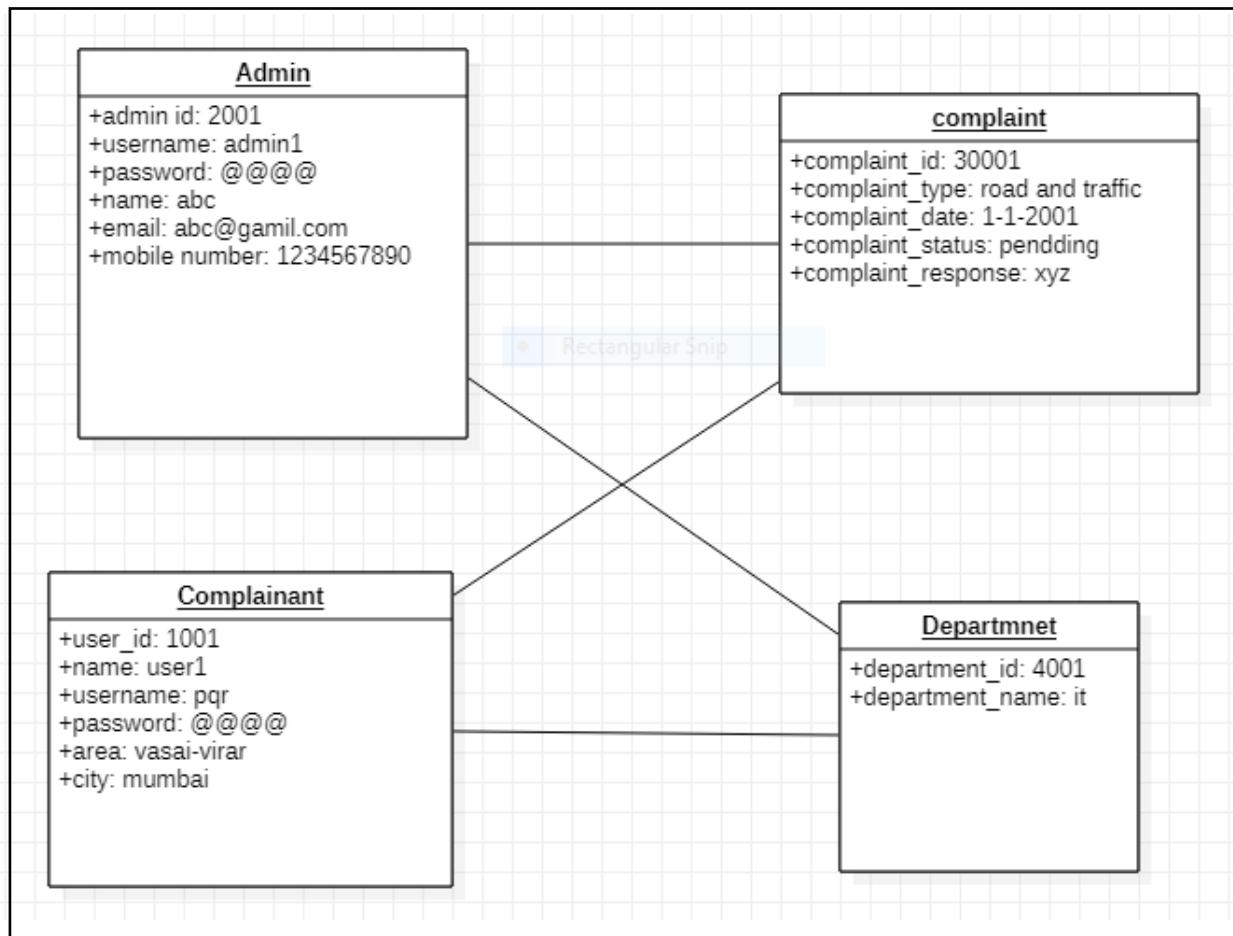


Figure 9:object diagram

3.7.6 Component diagram

UML Component diagrams are used in modelling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering.

Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

Basic Concepts of Component Diagram

A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment. In UML 2, a component is drawn as a rectangle with optional compartments stacked vertically. A high-level, abstracted view of a component in UML 2 can be modeled as:

1. A rectangle with the component's name
2. A rectangle with the component icon
3. A rectangle with the stereotype text and/or icon

Interface

Provided interface symbols with a complete circle at their end represent an interface that the component provides symbol is shorthand for a realization relationship of an interface classifier.

Required Interface symbols with only a half circle at their end represent an interface that the component requires (in both cases, the interface's name is placed near the interface symbol itself).

Subsystems

The subsystem classifier is a specialized version of a component classifier. Because of this, the subsystem notation element inherits all the same rules as the component notation element. The only difference is that a subsystem notation element has the keyword of subsystem instead of component.

Port

Ports are represented using a square along the edge of the system or a component. A port is often used to help expose required and provided interfaces of a component.

Relationships

Graphically, a component diagram is a collection of vertices and arcs and commonly contain components, interfaces and dependency, aggregation, constraint, generalization, association, and realization relationships. It may also contain notes and constraints.

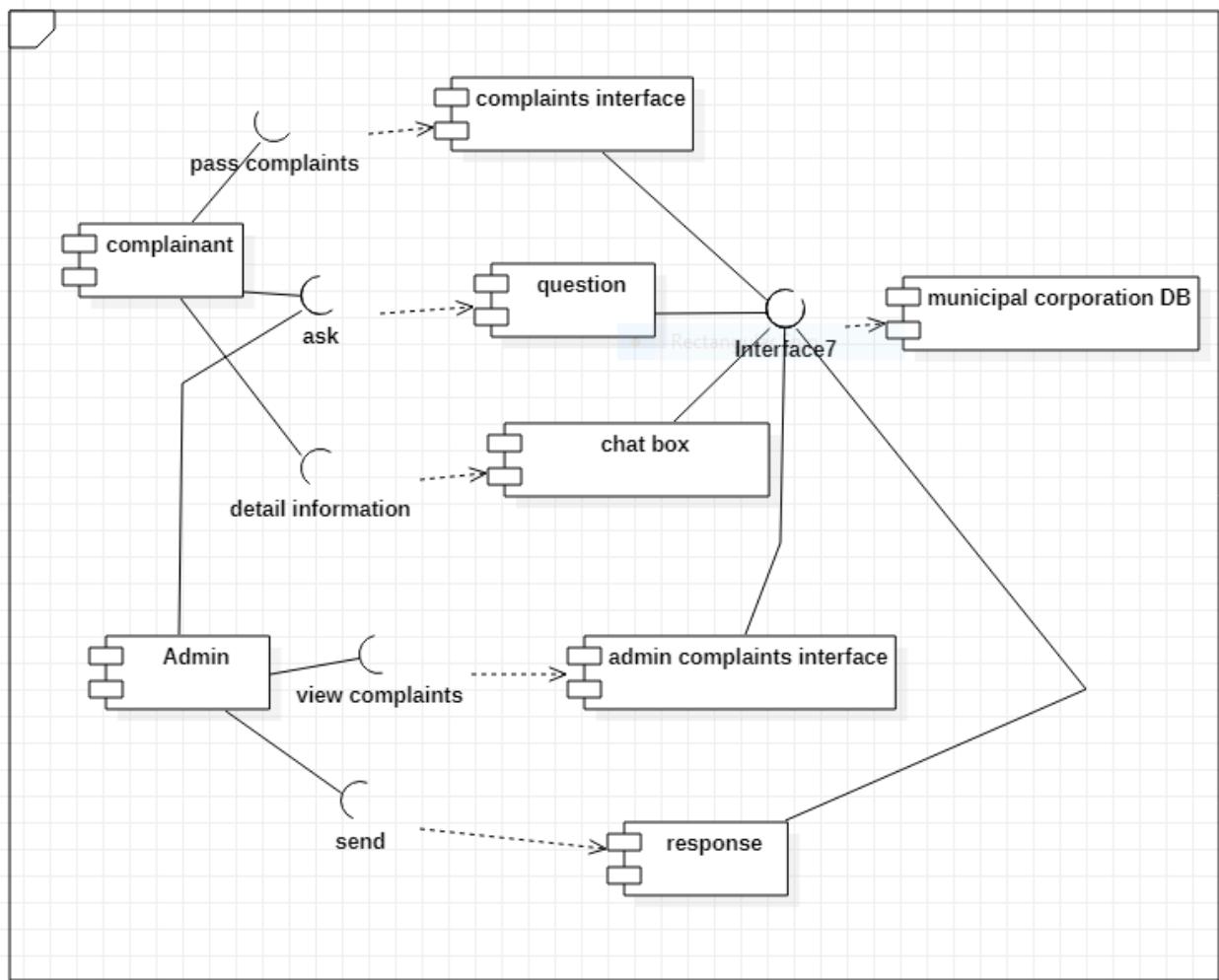


Figure 10:component diagram

3.7.7 Deployment diagram

A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of

structure diagram used in modelling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system

Purpose of Deployment Diagrams

- They show the structure of the run-time system
- They capture the hardware that will be used to implement the system and the links between different items of hardware.
- They model physical hardware elements and the communication paths between them
- They can be used to plan the architecture of a system.
- They are also useful for Document the deployment of software components or nodes

Deployment Diagram at a Glance

Deployment diagrams are important for visualizing, specifying, and documenting embedded, client/server, and distributed systems and also for managing executable systems through forward and reverse engineering.

A deployment diagram is just a special kind of class diagram, which focuses on a system's nodes. Graphically, a deployment diagram is a collection of vertices and arcs. Deployment diagrams commonly contain:

Nodes

- 3-D box represents a node, either software or hardware
- HW node can be signified with <<stereotype>>
- Connections between nodes are represented with a line, with optional <<stereotype>>
- Nodes can reside within a node

Other Notations

- Dependency

- Association relationships.
- May also contain notes and constraints.

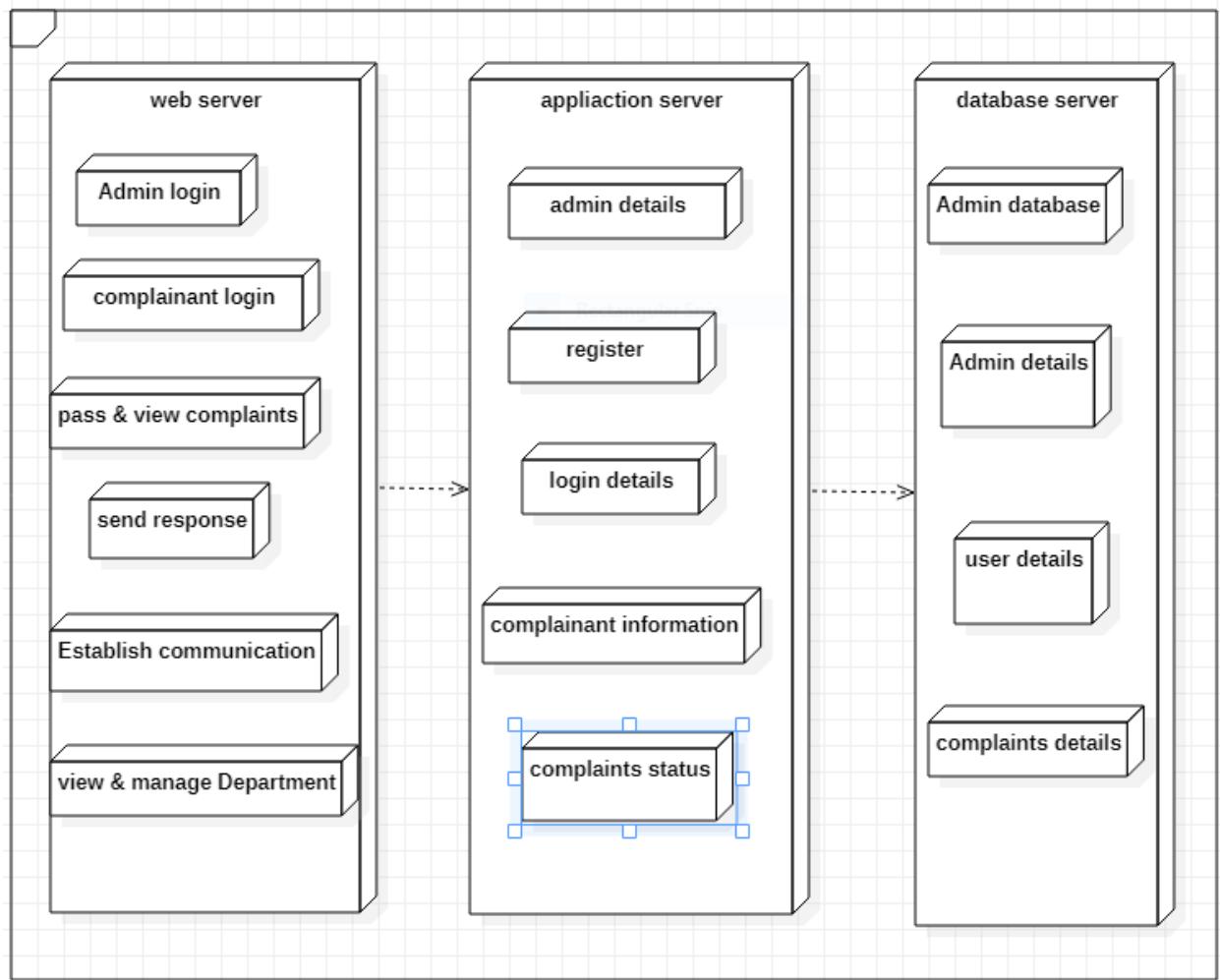


Figure 11:deployment diagram

Chapter 4

Implementation and Testing

4.1 Basic modules

A module is a software component or part of a program that contains one or more routines. One or more independently developed modules make up a program. An enterprise-level software application may contain several different modules, and each module serves unique and separate business operations. Modules make a programmer's job easy by allowing the programmer to focus on only one area of the functionality of the software application. Modules are typically incorporated into the program (software) through interfaces. Software applications include many different tasks and processes that cohesively serve all paradigms within a complete business solution. Early software versions were gradually built from an original and basic level, and development teams did not yet have the ability to use prewritten code.

The introduction of modularity allowed programmers to reuse prewritten code with new applications. Modules were created and bundled with compilers, in which each module performed a business or routine operation within the program.

Benefits of Modularization

The primary benefit of modularization is mastering the art of managing software complexity. When done well, good modular design leads to some very powerful, practical benefits:

- ❖ **understandability:** A well-modularized system is much easier to reason about, think about and communicate to others.

- ❖ improvability: Strongly encapsulated modules maximizes your ability to fix or improve individual module implementations without needing to update any other, dependent modules.
- ❖ refactorability: The less inter-dependencies in a project, the easier it is to make large changes across multiple modules.
- ❖ reusability: Modules with the best-conceived purposes are fully reusable. Whenever you have the same problem again, you can simply reuse the old solution.
- ❖ testability: Modules with good interfaces and minimal inter-dependencies are easier to test. Well-designed APIs can be easily unit-tested. Modules with minimal inter-dependencies can be tested without the need for mocks or more difficult integration-testing. Modules allow you to write tests once, ensure correctness, then reuse the module without the need for further testing.
- ❖ scalability: All of which adds up to the most important benefit of modules: They let our applications scale. It's impossible to build large applications without good modularization. Without modules, complexity will destroy your productivity.

1. Admin module:

Login

In this module, the admin will be allowed to access their account only if they have correct username and password of their existing account.

Logout:

In this module, admin can log out from his or her account.

2. User module:

Register:

If the user is visited first time to this website he or she have to first register on this website by giving some necessary information.

Login:

In this module, the user will be allowed to access their account only if they have correct username and password of their existing account.

Logout:

In this module, user can log out from his or her account.

3. Home screen module:

This screen will appear to the type of users, the one who are logged in this website. In this module user can interact with the multiple option from which he or she can perform multiple activity on this website.

4.Complaint module:

In this module, users can register their complaints regarding to their problems. As well as they can see the status of their complaints whether its approved or still pending. Also, they can communicate with the admin through the chat box and pass the extra information as well as ask some queries to the admin.

5. department module:

In this module, user can see the all department details of each and every department.

6.feedback module:

In this module, user can send some feedback to admin whatever user want to send the suggestion.

4.2 Event table

Sr no .	Event	Trigger	Source	Activity	Response	Destination

1.	Admin visit the website	Request to enter login details	Admin	Verify admin details	Login page of system	Database/system
2.	Admin request for login info the system	Request for login	Admin	Verify username and password	Home page of admin	Database
3.	Admin can view complaints	Show the details of complaints	Admin	Type of complaint, date, complainant name and other details	Complaint page for admin	Database
4.	Admin can establish the communication for more details	Admin communicate with the user	Admin	Communicating with each other	Admin chat with the user	Database
5.	Admin can send the response to the complainant	Show complaint response report	Admin	Type of complaint, date, complainant name and other details	Response will pass to the user	Database
6.	Admin can manage other	Add/ update details of	Admin	Name and other details of user	Admin will see account of	Database

	complainants in the system	user in the system			another user	
7.	Admin can manage departments	Add/ update details of Departmen t	Admin	Department id , name and other details	Departme nt informatio n will update	Database
8.	Admin can read feedback of complainant	Get a user feedback detail	Admin	User feedback details	Admin will get user sugges tion	Database
9.	Admin request for logout	Request for logout	Admin	Admin logged out from system	Login page of system	Admin
10 .	Complainant request for login	Request for login	Complainant/u ser	Verify username and password	Home page of complaina nt or user	Database
11 .	User can pass the complaints	Give the details of complaint	Complainant/u ser	Type of complaint, date, complainant name and other details	Complaint registered	Database
12 .	User can ask question	Getting clear with all the	Complainant/u ser	communicati ng with admin	User will receive response	Database

		doubt of user				
13 .	User can view status of the complaints	Show whether complaint is approved or not	Complainant/user	Approval status of complaints	Approval status will appear	Database
14 .	User can view department information	Show department details	Complainant/user	Departments id, name and other details	Department details will appear	Database
15 .	User can add photos with the complaints form	Provide extra information	Complainant/user	Adding 2 or 3 photos from different angel	Extra details attach with the form	Database
16 .	User can send feedback	User pass his suggestion s to admin	Complainant/user	sending user suggestion to the admin	Feedback will be submitted	Database
17 .	User request for logout	Request for logout	Complainant/user	User logged out from system	User login of system	User

Table 2: Event table

4.3 User interface design

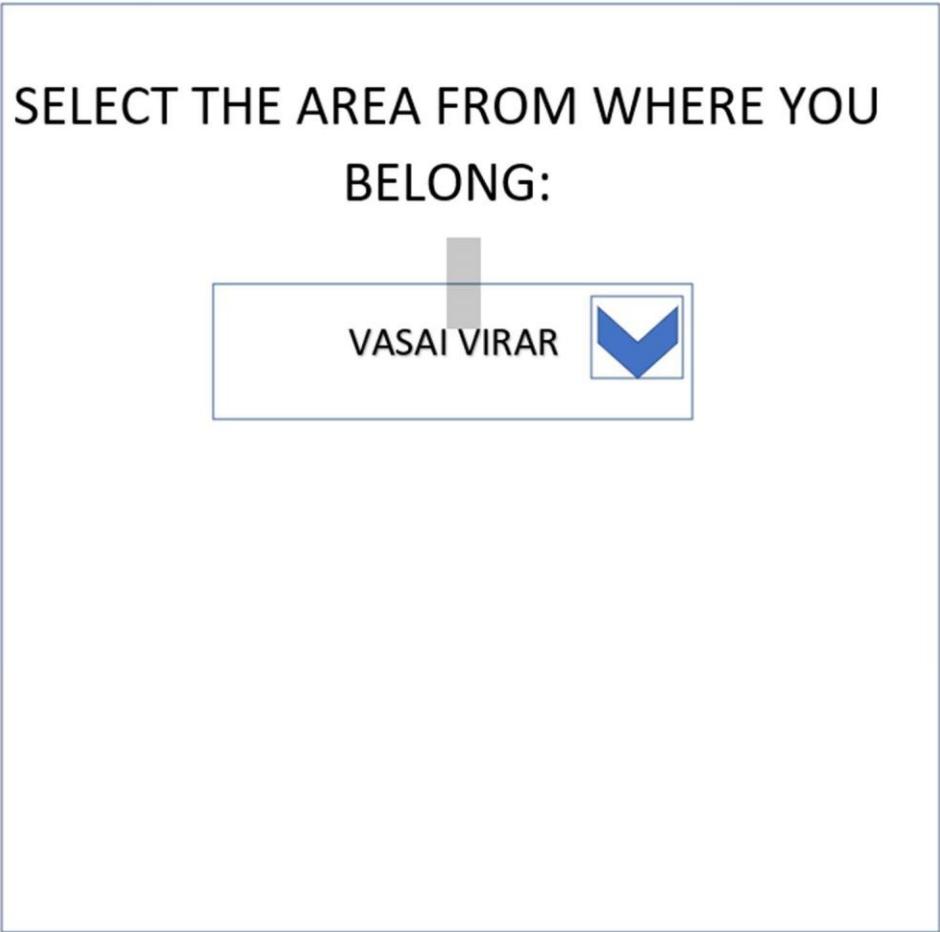


Figure 12:start page

User will select the area, from where he/she wants to complaint to municipal corporation. Once he/ she selected the area, interface of the municipal corporation belonging to that area will be appear on next page.

E.g. If he/ she select Vasai Virar as an area then Vasai Virar municipal corporation webpage will appear.

Home page

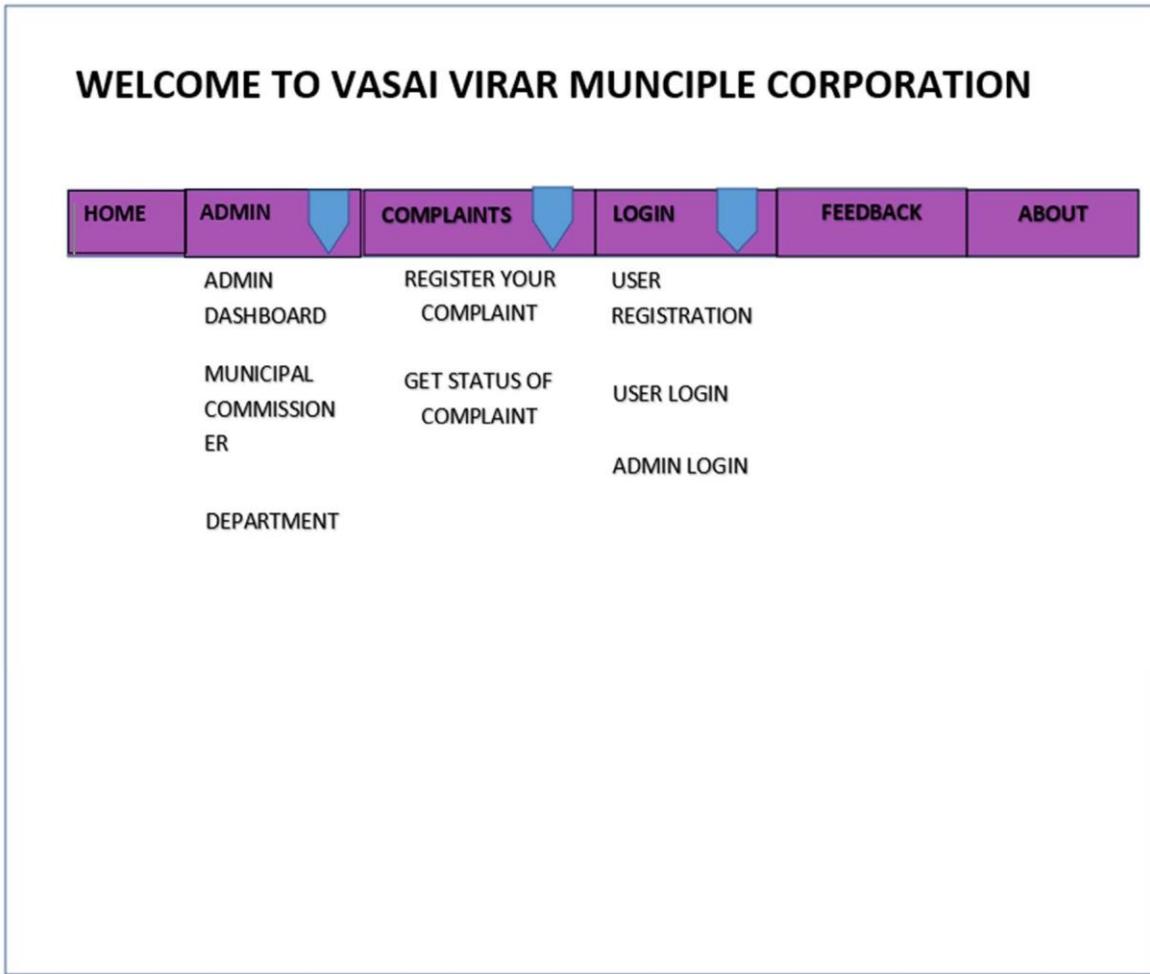


Figure 13:home page

This is the home page of that Vasai Virar municipal corporation. In this window complainant can complaint about particular issues by selecting the ‘register your complaint’.

REGISTRATION PAGE

NAME:	PQR
USERNAME:	ABSJP
PASSWORD:	#####
MOBILE NO.:	1234567890
CITY:	VASAI VIRAR
STATE:	MAHARASHTRA
AREA:	VASAI

Figure 14:registration page

This is the registration page for the complainant. User have to just fill the information and click on the register to move forward.

LOGIN PAGE

USERNAME: XYZ

PASSWORD: @@@@@"

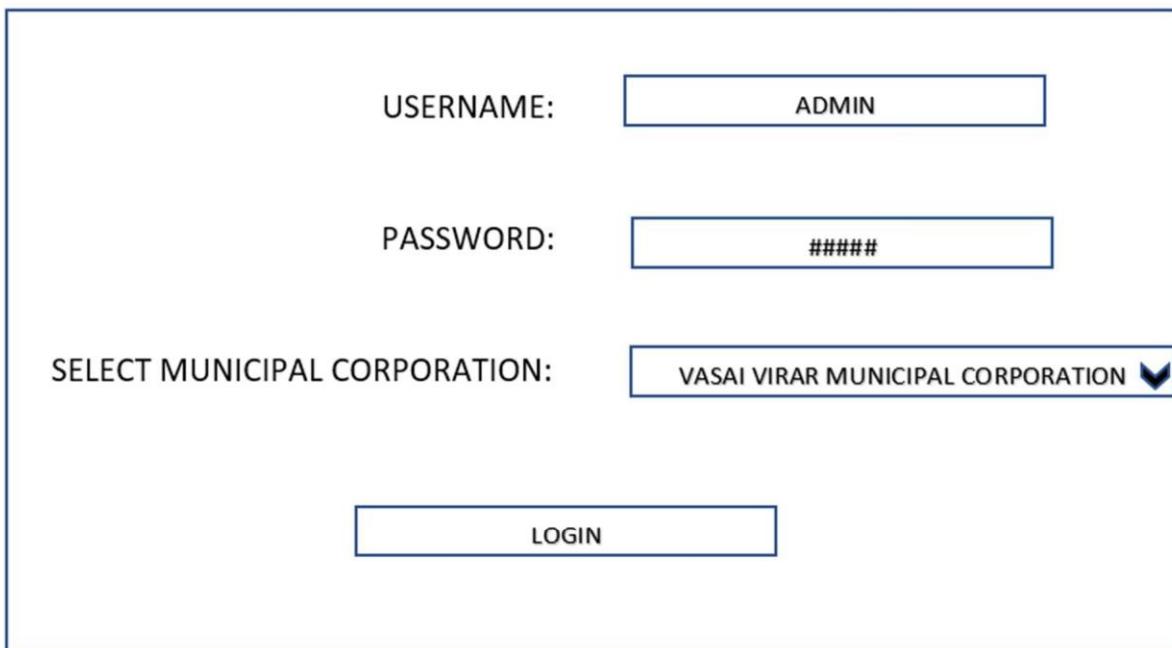
CONFIRM PASSWORD: @@@@@"

LOGIN

Figure 15:login page for user

Login page for user. To complaint about particular issue, he / she must login first.

ADMIN LOGIN



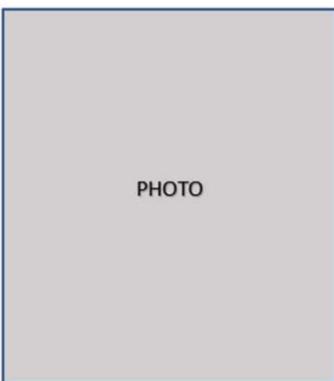
The image shows a login form titled "ADMIN LOGIN". It contains three input fields: "USERNAME" with the value "ADMIN", "PASSWORD" with the value "#####", and "SELECT MUNICIPAL CORPORATION" with the value "VASAI VIRAR MUNICIPAL CORPORATION" and a small blue logo. Below these fields is a "LOGIN" button.

USERNAME:	ADMIN
PASSWORD:	#####
SELECT MUNICIPAL CORPORATION:	VASAI VIRAR MUNICIPAL CORPORATION 
LOGIN	

Figure 16:admin login

This one is the admin side page where admin of particular municipal corporation can login by just selecting the municipal corporation.

MUNICIPAL COMMISSIONER



PHOTO

SHREE. XYZ. (I.A.S)

MUNICIPAL COMMISSIONER

FAX: - 0000-00000000

EMAIL: vasaivirarcorporation@yahoo.com

Welcome to the official web portal of Vasai Virar City Municipal Corporation. We aim to provide services to the residents of Vasai-Virar in an efficient and user friendly manner and reduce inconveniences caused to the public for getting information from various departments.

Figure 17:commissioner profile

This is the details of municipal commissioner. When user click on the municipal commissioner from home page he / she will redirect to this page.

Department

DEPARTMENT

LIST OF DEPARMENT:

- † IT DEPARTMENT
- † LICENSE
- † TOWN PLANING
- † MARKET
- † FIRE
- † GARDEN AND TREES

Figure 18:department

This is the department page where the list of departments is shown.

ADMIN DASHBOARD

ADMIN PAGE

COMPLAINT NO.	COMPLAINANT NAME	TYPE OF COMPLAINT	MOBILE NO.	COMPLAINT DATE	RESPONSE DATE	APPROVAL STATUS	ACTION
01	ABC	ROAD AND TRAFIC	1234567890	15 AUG. 2012	-	PENDDING	RESPONSE

Figure 19:admin dashboard

This is another admin page. Where, when admin login, he / she will redirect to this page and he /she can give response to the received complaint by clicking on response button.

Registration of complaints

REGISTER YOUR COMPLAINTS

SELECT COMPLAINT TYPE:	<input type="text" value="ROADS AND TRAFIC ISSUES"/>
WARD:	<input type="text" value="SELECT"/>
NAME OF COMPLAINANT:	<input type="text" value="ABC"/>
DATE OF COMPLAINT:	<input type="text" value="15 MAR. 2015"/>
COMPLAINANT MOBILE NUMBER:	<input type="text" value="1478523695"/>
UPLOAD SOME PHOTOS(OPTIONAL):	<input type="button" value="UPLOAD"/>
<input type="button" value="GET STATUS"/>	

Figure 20:complaint registration

Here, complainant can complaint by selecting the ‘type of complaint’ and some other information which is needed to fill and just submit.

GET STATUS OF YOUR COMPLAINT

NAME	TYPE OF COMPLAINT	MOBILE NO.	COMPLAINT DATE	RESPONSE DATE	APPROVAL STATUS
ABC	ROAD AND TRAFIC	1234567890	15 AUG. 2012		PENDDING

ADD COMPLAINT

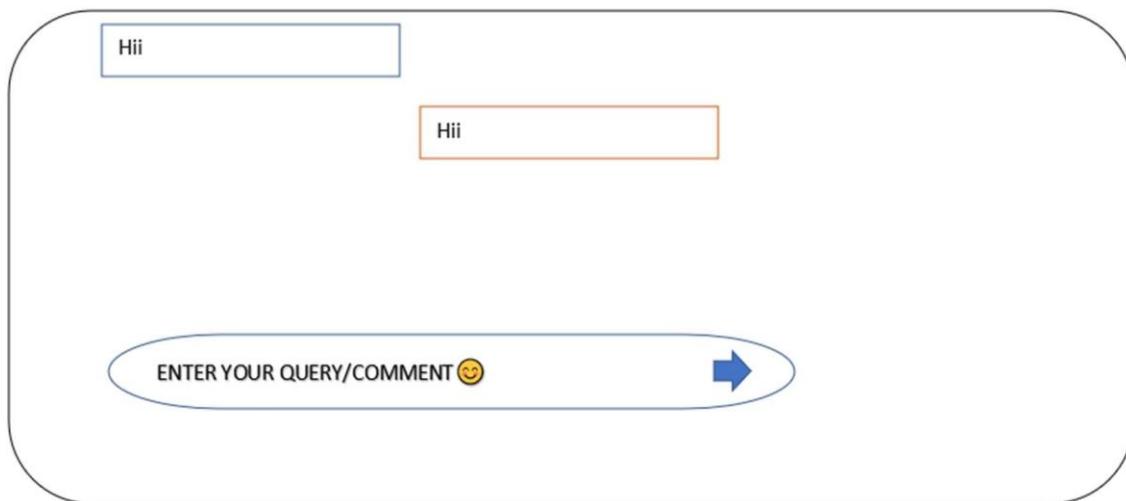
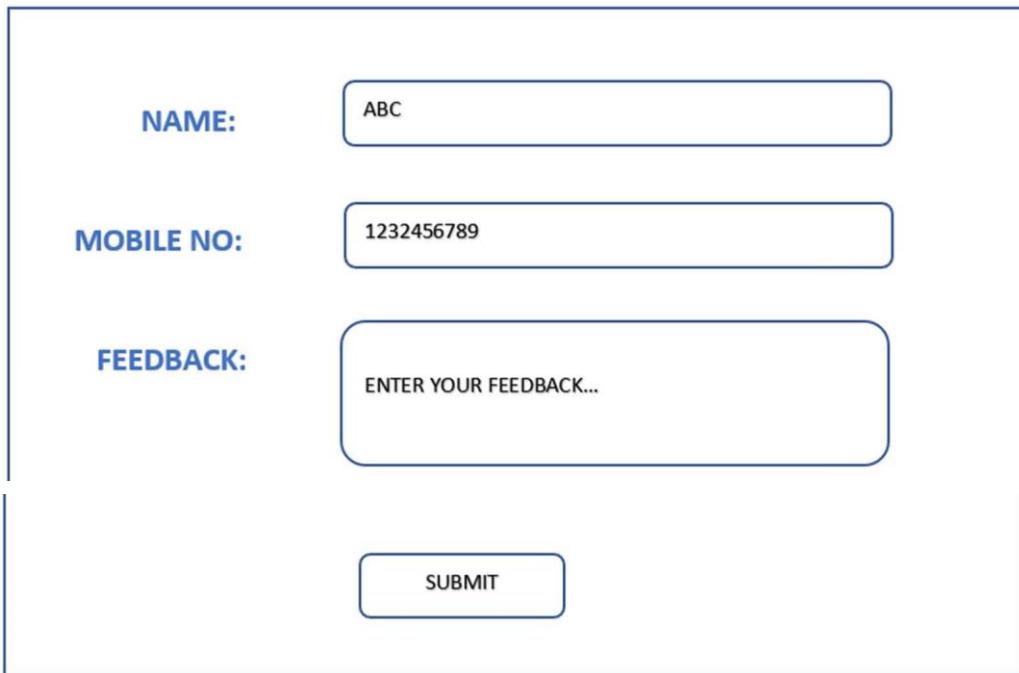


Figure 21:status interface

Here, complainant can see the details of their complaint whether the complaint is approved or not, Also, user can establish the communication with admin, as well as he can add more complaint by click on ‘add complaint’.

FEEDBACK



A wireframe diagram of a feedback form. It features a large rectangular container with a thin blue border. Inside, there are three input fields: a top field for 'NAME' containing 'ABC', a middle field for 'MOBILE NO.' containing '1232456789', and a bottom field for 'FEEDBACK' with the placeholder 'ENTER YOUR FEEDBACK...'. Below these fields is a single button labeled 'SUBMIT'.

NAME:	ABC
MOBILE NO.:	1232456789
FEEDBACK:	ENTER YOUR FEEDBACK...
SUBMIT	

Figure 22:feedback

This is the feedback page where user can give their suggestion to that municipal corporation.

About corporation interface

ABOUT CORPORATION

DEMOGRAPHICS • Rectangular Snip

- VVCMC Constituted on 3rd July 2009 comprising 4 Municipal Councils and 53 villages.
- VVCMC is declared as "C" Class Municipal Corporation
- Geographical Area of VVCMC -311 Sq. Kms.
- Population as per census 2011 -12.22 Lakhs
- Projected Current population -19.85 Lakhs
- Declared as Special Planning Authority for 21 Villages in Vasai Virar Sub Region .
- Area including SPA :-380 sq kms.
- Total number of electoral wards -115
- Total Number of administrative wards -9(A to I)
- Location-located at 19deg. 28min . north-90 deg. 47 min. north and 72 deg. 48 min. east-72 deg. 8 min. east

The Vasai-Virar City Municipal Corporation (VVCMC) was established on 3rd July 2009. Vasai Virar corporation is located between 19 deg. 28 min. North-90 deg. 47 min. North latitude and 72 deg. 48 min. East-72 deg. 8 min. East

Figure 23: about corporation

This is about the municipal corporation.

4.4 security issues

Issues	Solution
Complainant data	Data will be placed in encrypted form in the database and accessible only by user and admin.

User password	Access will be given only to the valid user / complainants and admin.
Admin / user privilege	User can access only his account, but admin can access all the account.

Table 3: security table

4.5 Testing Approach

Implementation of test strategy for a particular project is known as "test approach". The test approach is usually defined in all test plans and test designs. Test approach refers to the commencement of various project activities such as planning the testing process, selecting the designs, defining the entry and exit criteria etc.

There are two approaches in testing:

1. Preventative approach.
2. Reactive approach.

Preventative approach

In preventative approach, tests are designed at an early stage, i.e. before the commencement of software development.

Reactive approach

In Reactive approach, tests are designed after software development.

Different Test approaches:

There are many strategies that a project can adopt depending on the context and some of them are:

- Dynamic and heuristic approaches
- Consultative approaches
- Model-based approach that uses statistical information about failure rates.

- Approaches based on risk-based testing where the entire development takes place based on the risk
- Methodical approach, which is based on failures.
- Standard-compliant approach specified by industry-specific standards.

Before selecting a 'test approach', we should consider the following factors:

- Analyse the various risks associated with failure of the project - to the people the environment, and the company.
- Analyse the expertise and efficiency of the people in the proposed tools and techniques.
- The nature of product and the business.

4.5.1 Unit Testing

UNIT TESTING is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object. In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a Whitebox testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

Why Unit Testing?

Unit Testing is important because software developers sometimes try saving time doing minimal unit testing and this is myth because inappropriate unit testing leads to high cost Defect fixing during System Testing, Integration Testing and even Beta Testing after

application is built. If proper unit testing is done in early development, then it saves time and money in the end.

Here, are the key reasons to perform unit testing in software engineering:

1. Unit tests help to fix bugs early in the development cycle and save costs.
2. It helps the developers to understand the testing code base and enables them to make changes quickly
3. Good unit tests serve as project documentation
4. Unit tests help with code re-use. Migrate both your code and your tests to your new project. Tweak the code until the tests run again.

How to do Unit Testing

In order to do Unit Testing, developers write a section of code to test a specific function in software application. Developers can also isolate this function to test more rigorously which reveals unnecessary dependencies between function being tested and other units so the dependencies can be eliminated. Developers generally use UnitTest framework to develop automated test cases for unit testing.

Unit Testing is of two types

- Manual
- Automated

Unit testing is commonly automated but may still be performed manually. Software Engineering does not favor one over the other but automation is preferred. A manual approach to unit testing may employ a step-by-step instructional document.

Under the automated approach-

- A developer writes a section of code in the application just to test the function. They would later comment out and finally remove the test code when the application is deployed.
- A developer could also isolate the function to test it more rigorously. This is a more thorough unit testing practice that involves copy and paste of code to its own testing environment than its natural environment. Isolating the code helps in revealing unnecessary dependencies between the code being tested and other units or data spaces in the product. These dependencies can then be eliminated.
- A coder generally uses a UnitTest Framework to develop automated test cases. Using an automation framework, the developer codes criteria into the test to verify the correctness of the code. During execution of the test cases, the framework logs failing test cases. Many frameworks will also automatically flag and report, in summary, these failed test cases. Depending on the severity of a failure, the framework may halt subsequent testing.
- The workflow of Unit Testing is 1) Create Test Cases 2) Review/Rework 3) Baseline 4) Execute Test Cases.

Unit Testing Techniques

The Unit Testing Techniques are mainly categorized into three parts which are Black box testing that involves testing of user interface along with input and output, White box testing that involves testing the functional behavior of the software application and Gray box testing that is used to execute test suites, test methods, test cases and performing risk analysis.

Code coverage techniques used in Unit Testing are listed below:

- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Finite State Machine Coverage

Unit Testing Advantage

- Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API.
- Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly (i.e. Regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.
- Due to the modular nature of the unit testing, we can test parts of the project without waiting for others to be completed.

Unit Testing Disadvantages

- Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs
- Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.

4.5.2 Integration Testing

INTEGRATION TESTING is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as ‘I & T’ (Integration and Testing), ‘String Testing’ and sometimes ‘Thread Testing’.

Why do Integration Testing?

Although each software module is unit tested, defects still exist for various reasons like

- A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity
- At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary.
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues.

Approaches, Strategies, Methodologies of Integration Testing

Software Engineering defines variety of strategies to execute Integration testing, viz.

- Big Bang Approach:
- Incremental Approach: which is further divided into the following
 - Top Down Approach
 - Bottom Up Approach
 - Sandwich Approach – Combination of Top Down and Bottom Up

Big Bang Testing

Big Bang Testing is an Integration testing approach in which all the components or modules are integrated together at once and then tested as a unit. This combined set of components is considered as an entity while testing. If all of the components in the unit are not completed, the integration process will not execute.

Advantages:

- Convenient for small systems.

Disadvantages:

- Fault Localization is difficult.
- Given the sheer number of interfaces that need to be tested in this approach, some interfaces link to be tested could be missed easily.
- Since the Integration testing can commence only after “all” the modules are designed, the testing team will have less time for execution in the testing phase.
- Since all modules are tested at once, high-risk critical modules are not isolated and tested on priority. Peripheral modules which deal with user interfaces are also not isolated and tested on priority.

Incremental Testing

In the Incremental Testing approach, testing is done by integrating two or more modules that are logically related to each other and then tested for proper functioning of the application. Then the other related modules are integrated incrementally and the process continues until all the logically related modules are integrated and tested successfully.

Incremental Approach, in turn, is carried out by two different Methods:

- Bottom Up
- Top Down

Stubs and Drivers

Stubs and Drivers are the dummy programs in Integration testing used to facilitate the software testing activity. These programs act as a substitute for the missing models in the testing. They do not implement the entire programming logic of the software module but they simulate data communication with the calling module while testing.

Stub: Is called by the Module under Test.

Driver: Calls the Module to be tested.

Bottom-up Integration Testing

Bottom-up Integration Testing is a strategy in which the lower level modules are tested first. These tested modules are then further used to facilitate the testing of higher level modules. The process continues until all modules at top level are tested. Once the lower level modules are tested and integrated, then the next level of modules are formed.

Advantages:

- Fault localization is easier.
- No time is wasted waiting for all modules to be developed unlike Big-bang approach

Disadvantages:

- Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects.
- An early prototype is not possible

Top-down Integration Testing

Top Down Integration Testing is a method in which integration testing takes place from top to bottom following the control flow of software system. The higher-level modules are tested first and then lower level modules are tested and integrated in order to check the software functionality. Stubs are used for testing if some modules are not ready.

Advantages:

- Fault Localization is easier.
- Possibility to obtain an early prototype.
- Critical Modules are tested on priority; major design flaws could be found and fixed first.

Disadvantages:

- Needs many Stubs.

- Modules at a lower level are tested inadequately.

How to do Integration Testing?

The Integration test procedure irrespective of the Software testing strategies (discussed above):

1. Prepare the Integration Tests Plan
2. Design the Test Scenarios, Cases, and Scripts.
3. Executing the test Cases followed by reporting the defects.
4. Tracking & re-testing the defects.
5. Steps 3 and 4 are repeated until the completion of Integration is successful.

4.5.3 Test cases design

Index	Test case	Test data	State	Test input value	Expected data
1	The username should be only in characters(A-Z) and (a-z) and special characters	Number or any other special characters	Invalid	pqrstu	Enter valid name

		Alphabet A-Z and a-z	Valid	Abc@78	Name accepted
2	The password should be minimum 8 characters or more including some special characters	Only number or alphabets	Invalid	abcd	Password must be minimum 8 characters including some special characters.
		Alphabets special characters and number	Valid	gfhk@6789	Confirm password
3	Mobile number should be 10 digits only	Characters or numbers less than 10 or more than 10 digits	Invalid	123445hj	Mobile number should be 10 digits only
		Number between 0-9	Valid	1234567890	Mobile number accepted

4	Id should be in proper format	Id with special characters alphabets	Invalid	45hgf	Enter proper id
		Only digit	Valid	123	Accepted
5	The address should be in proper format	Address without alphabets and digit	Invalid	vbn	Enter correct address
		Should be consist of alphabets A-Z or a-z and digit	Valid	Parera road, Mumbai 453787	Accepted
6	Email id should be in proper format	Id without special characters and alphabets	Invalid	6678gmail	Enter proper email id
		Alphabets and special characters	Valid	pqr@gmail.com	Accepted

Table 4: Test case

