



Daffodil
International
University

Complain Management Platform

Submitted by

Omer Faruq Nadim

171-35-1934

Department of Software Engineering

Daffodil International University

Supervised by

S A M Matiur Rahman

Associate Professor

Department of Software Engineering

Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of
Bachelor of Science in Software Engineering.

APPROVAL

This project titled on “Complain Management Platform”, submitted by Omer Faruq Nadim (ID: 171-35-1934) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

DECLARATION

I hereby declare that project titled “Complain Management Platform” has been completed by me under the supervision of Mr. S A M Matiur Rahman, Associate Professor, Department of Software Engineering, Daffodil International University for the purpose of achieving degree of Bachelor of Science from Daffodil International University. This is also declared by me that neither this project nor any part of this project has been used or submitted elsewhere for any kind of degree or awards.

ACKNOWLEDGEMENT

First and foremost, I want to praise Almighty God for all of His blessings during my project work, which gave me perseverance and helped me to finish on time. So, I consider myself quite privileged to have completed this project.

All of them are the product of such observations and assistance, for which I am grateful. I also like to thank and acknowledge my supervisor, Mr. S A M Matiur Rahman, Associate Professor, Department of Software Engineering, Daffodil International University, Dhaka.

He knew a lot about the software industry, and his solid judgment, supervision, and high competence level aided me much in my project. When I need to oversee, he is always available.

I would like to convey my gratitude and gratitude to Dr. Imran Mahmud, Associate Professor and Head of the Department of Software Engineering, for his thoughtful review of my project, which helped me to actually accomplish it.

I would also want to thank the other academic and staff members of Daffodil International University's Department of Software Engineering for their warmth and helpful attitude toward us.

I'd want to thank everyone of my Daffodil International University students that participated in this debate while taking the course. I would also like to thank other faculty and staff members of Daffodil International University's Software Engineering department for their friendliness and helpful attitude toward us.

I'd want to thank everyone of my Daffodil International University students who participated in this debate while doing school work.

Finally, and most significantly, I must express my deepest gratitude for their constant support and patience.

ABSTRACT

The "Complaint Management Platform" project is intended to digitally gather university student complaint resolution surveys. The project is built on a web application that provides a major way to organize, monitor, manage, and handle student grievances.

Providing the University with a powerful tool for identifying and targeting specific problems, tracking complaint handling performance, and improving resolution procedures.

Complain Management Platform (CMP) is an online complaint management system that provides students with instant access to a variety of information and gathers legitimate concerns that students confront on their campus.

Table of Content

APPROVAL	i
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT.....	iv
CHAPTER 01	1
Introduction.....	1
1.1 Project Overview	1
1.2 Project Purpose	1
Benefits	2
Beneficiaries.....	2
1.3 Proposed System Model	2
CHAPTER 02	5
SOFTWARE REQUIREMENT SPECIFICATION.....	5
2.1 Functional Requirement	5
2.2 Performance Requirement	7
2.3 Dependability Requirement.....	7
2.4 Maintainability and Supportability Requirement	8
• Provide a user guide.....	8
• Provide FAQ for a better user experience.	8
2.5 Security Requirement	8
CHAPTER 03	9
SYSTEM ANALYSIS	9
Introduction:	9
3.1 Use Case	10
3.1.2 Use Case for Admin.....	15
3.1.3 Use Case for Authority	19
3.4 ERD Diagram	55
CHAPTER 04	56
DEVELOPMENT TOOLS & TECHNOLOGY.....	56
4.1 User Interface Technologies	56
4.2 Implementation Technologies	56
4.3 Platform & Environment	56

CHAPTER 05	58
SYSTEM TESTING	58
5.1 Introduction	58
The purpose of system testing is to find flaws. This activity's major goal is to verify that the procedure is effectively conveyed and that all requirements are satisfied. System testing is important for assessing whether or not a function is functioning properly.....	58
5.2 Test Case	58
Test Case No -1	58
Testing Case, No-2	58
Testing Case, No-3	59
Testing Case, No-4	60
CHAPTER 06	61
USER MANUAL.....	61
6.1 User manual	61
FOR User	61
For Admin.....	68
For Authority.....	73
CHAPTER 07	77
PROJECT SUMMARY	77
7.1 GitHub Link:.....	77
7.2 Limitations:.....	77
7.3 Obstacle & Achievement.....	77
Obstacle:	77
Achievement:	77
7.4 Future Work	77
References	78

List of Figures

1.3.1 Agile Model	11
1.3.2 Gannt Chart	12
2.1.1 Functional Requirement for User	13
2.1.2 Functional Requirement for Admin	14
2.1.3 Functional Requirement for Authority	15
3.1.1 Use Case	17
3.2.1.1 User Login	29
3.2.1.2 User Registration	30
3.2.1.3 User Password Change	31
3.2.1.4 User Assign Complain	32
3.2.1.5 User View Complain.....	33
3.2.1.6 User Edit Complain	34
3.2.1.7 User Assign Suggestion	35
3.2.1.8 User View Suggestion.....	36
3.2.1.9 User Edit Suggestion.....	37
3.2.1.10 User Complain Details	38
3.2.1.11 User Complain Status.....	39
3.2.1.12 User Get Help.....	39
3.2.2.1 Admin Login.....	40
3.2.2.2 Admin Complain.....	41
3.2.2.3 Admin Update Complain	42
3.2.2.4 Admin Complain Approve.....	43
3.2.2.5 Admin User Details.....	44
3.2.2.6 Admin User Help	45
3.2.2.7 Admin User Suggestion Details	46
3.2.2.8 Admin Generate Report	47
3.2.3.1 Authority Login.....	49
3.2.3.2 Authority View User	50
3.2.3.3 Authority User Complain.....	51
3.2.3.4 Authority User Suggestion.....	52
3.2.3.5 Authority Report Check	53
3.3.1.1 Sequence Diagram User login	54
3.3.1.2 Sequence Diagram User Registration	55

3.3.1.3 Sequence Diagram User Complain and Suggestion	56
3.3.1.4 Sequence Diagram User Main Options	57
3.3.2.1 Sequence Diagram Admin login	58
3.3.2.1 Sequence Diagram Admin Main Options	59
3.3.2.1 Sequence Diagram Admin Notice Update	60
3.3.3.1 Sequence Diagram Authority Login	61
3.3.2.2 Sequence Diagram Authority Main options	62
3.4.1 ERD Diagram	63
Fig 6.1.1 User Login Page	70
Fig 6.1.2 User Registration Page	70
Fig 6.1.3.1 User Dashboard 1 Page	71
Fig 6.1.3.2 User Dashboard 2 Page	71
Fig 6.1.3.3 User Dashboard 3 Page	72
Fig 6.1.4 User Complain	72
Fig 6.1.5 User Complain View	73
Fig 6.1.6 User Suggestion	73
Fig 6.1.7 User Suggestion View	74
Fig 6.1.8 User Notice Board	74
Fig 6.1.9 User Helpline	75
Fig 6.1.10 User Profile	75
Fig 6.1.11 User Notification	76
Fig 6.2.1 Admin Login	76
Fig 6.2.2 Admin Dashboard	77
Fig 6.2.3 Admin User List	77
Fig 6.2.4 Admin User Complain	78
Fig 6.2.5 Admin User Suggestion	78
Fig 6.2.6 Admin Notice Add.....	79
Fig 6.2.7 Admin Notice Show	79
Fig 6.2.8 Admin User Complain Approve	80
Fig 6.2.9 Admin Profile Update	80
Fig 6.3.1 Authority Login	81
Fig 6.3.2 Authority Dashboard 1	81
Fig 6.3.2 Authority Dashboard 2	82
Fig 6.3.3 Authority User List	82
Fig 6.3.4 Authority User Complain	83

Fig 6.3.5 Authority User Suggestion 84

Fig 6.3.6 Authority Complain Approve 84

Fig 6.3.7 Authority Profile Update 85

CHAPTER 01

Introduction

1.1 Project Overview

Complain Management System provides a web-based solution to student problems that saves time, minimizes annoyance, and provides a quick solution.

The project's purpose is to make it quicker to organize, monitor, manage, and deal with complaints, as well as to provide the campus with an effective tool for identifying and targeting problem areas, tracking complaints handling performance, and improving business processes.

Complain Management System is an online complaint management system that is intended to respond to student issues as quickly as possible that saves time by providing students with an online alternative to resolve their complaints while being linked to the varsity administrator.

The Complaint Management System intends to enhance the ability to monitor, track, and resolve issues, offering the University with an efficient tool for identifying issues, monitoring the effectiveness of complaint management, and enhancing the campus.

Students may employ this strategy to save time and reduce misconduct in the classroom.

The proposed technique eliminates the need for students to go to the coordinator's office to resolve issues. He can get his problem fixed by publishing it in this suggested system, and he can also provide a possible solution to the problems that have been reported on the system.

In a private university, there is no direct communication between the administration (coordinator) and the students in an efficient manner for problem solving, i.e., to get a problem solved in our campus, we have to bribe the officials and get it solved in two months when it can be solved in one month.

1.2 Project Purpose

1.2.1 Project Background

In the digital Bangladesh century, every problem must be tackled digitally. We have several obstacles at times in managing a big varsity campus that is ideal for students. We can readily communicate our concern to our varsity authorities using this digital campaign "Complaint Management System," and we can easily obtain a solution to that problem.

1.2.2 Benefits & Beneficiaries

Benefits

- Users may simply submit ideas to authorities using the system.
- Users can also submit complaints to authorities via the system.
- Users can receive alerts if they accept or reject complaints.
- Users can seek assistance from the help line via live chat.
- University notifications are available in the dashboard notice area.

Beneficiaries

- Student (User)
- Admin
- Super Admin

1.2.3 Goals

- Offer a digital solution
- Simple automated system
- Provide the report anonymously.
- Improve the campus.
- Improve student convenience on campus.
- Organizations Providing all university notifications.

1.3 Proposed System Model

A software model is an organized representation of a software process. Each Model illustrates a process from a unique perspective.

1.3.1 Agile Model

Agile seems to be the model we propose. In software, the Agile Model terminology signifies the ability to adjust to technological changes and people as required. Engineering terminology performed by functional team. [1]

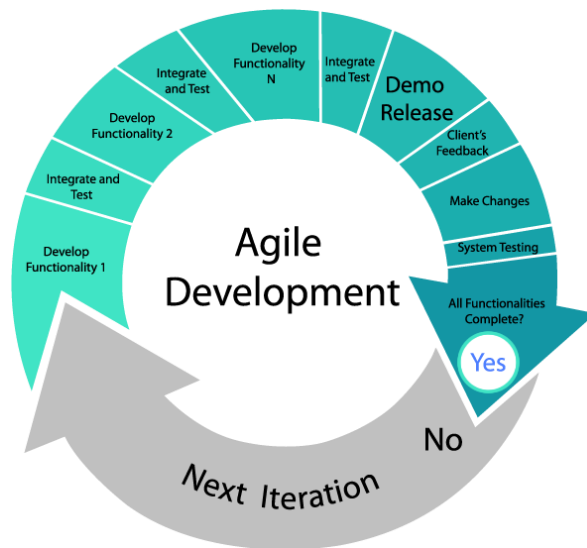


Fig 1.3.1 Agile Model

1.3.2 Gantt Chart

Activities		W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15
Planning	Ideas															
	Problem definition															
	Proposal planning															
Requirements	Requirement Specification															
	Requirement analysis															
QA -1	Quality assurance															
System Design	Design specification															
	Interface design															
	Database design															
Development	Development system modules															
	Integrate system modules															
QA -2	Test Cases															
Testing	Black box testing															
Resolve Issues	Resolve issues found															
Release	Software release															

Fig 1.3.2 Gantt Chart

CHAPTER 02

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional Requirement

For User

RQ ID	RQ Name	Description	RQ Type	Priority
1	Login	Users can log in with own username and password.	Functional	High
2	Registration	Users can register on the system by providing all the necessary information.	Functional	High
3	Assign Complain	Users can post any type of complain with a detailed explanation.	Functional	High
4	Assign Suggestions	Users can submit Specific recommendations of any category with an explanation.	Functional	Mid
5	View Complain	The user can access their assigned review.	Functional	Mid
6	Update Complain	The user can make changes to their complaint.	Functional	Mid
7	View Suggestion	The user can view the Suggestion that has been assigned to them.	Functional	Mid
8	Get Help	Users can get assistance from the Helpline through Live Chat.	Functional	Mid
9	Get Notification	Every notice may be obtained from the university notice board.	Functional	Mid

Fig 2.1.1 Functional Requirement for user

For Admin:

RQ ID	RQ Name	Description	RQ Type	Priority
1	Login	Admin can log in with own username and password.	Functional	High
2	See All Complain	Admin has access to all User Complaint Submissions.	Functional	High

3	Solve the problem	Admin can manually solve the problem and assign this to the system as a way to solve.	Functional	High
4	Change Status	Admin can accept user suggestions and complaints.	Functional	High
5	Generate Report	If an administrator approves a complaint, an email is sent to the admin.	Functional	High
6	Manage Account	The user account can be updated or deleted by the administrator.	Functional	High
7	View User list	The administrator can view the user list.	Functional	High
8	Give Solution	Admin may provide emergency support to users via live Chat.	Functional	High
9	Give Notice	The administrator has the ability to upload notices to the system.	Functional	High
10	View Notice	Admin has access to the Notice that has been uploaded.	Functional	Mid

Fig 2.1.2 Functional Requirement for Admin

For Authority

RQ ID	RQ Name	Description	RQ Type	Priority
FC.A.RQ 1	Login	Authority can log in with own username and password.	Functional	High
FC.A.RQ 2	View Complains	Authority has access to all User Complaint Submissions	Functional	High
FC.A.RQ 3	View Suggestions	Authority Admin has access to all User Suggestions Submissions	Functional	High
FC.A.RQ 4	User complain approve	The authority has the power to approve a user's complaint request.	Functional	High
FC.A.RQ 5	View all user list	The Authority can view the user list.	Functional	High
FC.A.RQ 6	Check Report	The administrator's report can be evaluated or verified by the authority.	Functional	High

2.2 Performance Requirement

2.2.1 Speed and Latency Requirement

- **Data would be inserted into the MySQL database in a number of seconds.**
- **Database queries would respond quickly and provide results in a matter of seconds.**
- **The system GUI should load in 3 seconds.**

However, it is subject to machine and internet speed.

2.2.2 Precisions Requirements

- Following the successful login, the user should display the correct result.
- All complain and application data should be correct and validly stored in the database.
- A particular complaint and application would be viewed by a certain user.
- Only registered users can make changes to their complaint information.

2.3 Dependability Requirement

2.3.1 Reliability Requirements

- The user registration should create a new user and update the database with the provided information.
- Only accurate information can be used to get access to the system.
- If a person forgets their password, they can change it.
- Every new user must validate their email address on their initial login.
- Every user has the ability to amend their complaint information.
- This system should be accessible via a web browser.
- Upon user request, this system should perform operations promptly.
- This system's design is responsive, making it suitable for any device (Mobile, Tab, PC).

2.4 Maintainability and Supportability Requirement

2.4.1 Maintenance Requirement

- **Modify the system as the application environment changes.**
- **Fix a problem when the system is corrupted.**
- **Correct user-inadvertent data errors.**

2.4.2 Supportability Requirements

- Provide a user guide.
- Provide FAQ for a better user experience.

2.5 Security Requirement

2.5.1 Access Requirement

- Only registered and email-verified users can access the system.
- Only the administrator may authorize a user complaint.
- Only particular email addresses and passwords will be accepted for Authority/Admin login.
- Admin and Authority can investigate user complaints.
- The administrator can send an email notification to the authority.

CHAPTER 03

SYSTEM ANALYSIS

Introduction:

In general, system development is divided into two parts: data integration and computer engineering. Software architecture refers to the process of designing a new system or expanding or augmenting an existing system. But, before we begin planning, we must thoroughly examine the existing system and identify how technology might best be leveraged to increase production. Network analysis is the method of collecting and analyzing data, detecting issues, and using the findings to propose changes happen. The systems analyst is in charge of this. It is the process of creating a new or improved business system and describing its components to meet specific needs.

3.1 Use Case

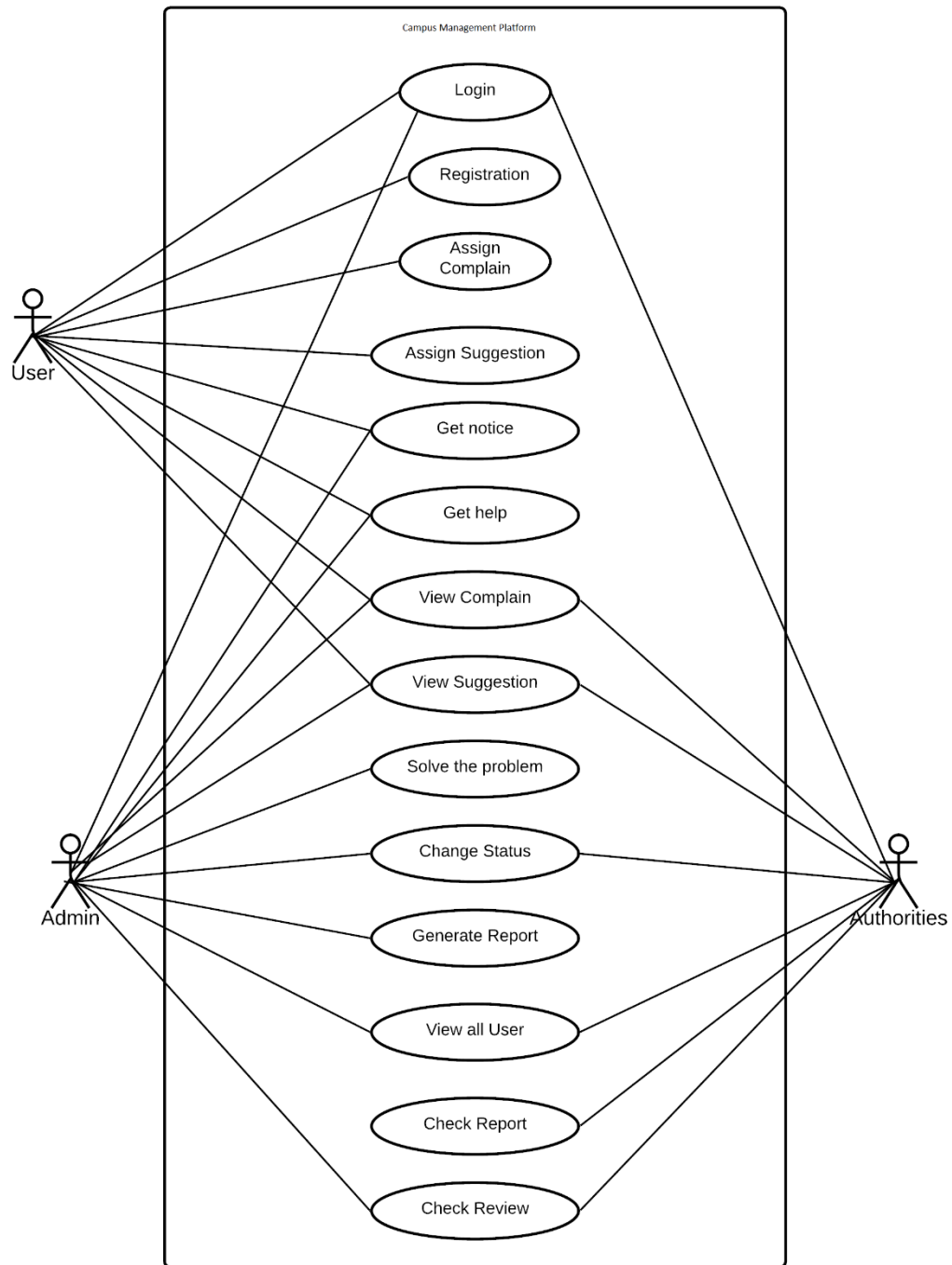


Fig 3.1.1: Use case [2]

3.1.1 Use Case for User

Use Case name	Login	
Scenario	User	
Brief Description	Users must use their own authentic email id to log in..	
Actor	Student	
Pre-condition	Authentic email and login type must be used.	
Post-Condition	Before logging in, users must confirm their email address.	
Flow of Condition	User	System
	<ul style="list-style-type: none"> Before logging in, users must validate their email address. 	<ul style="list-style-type: none"> Verify a genuine email address. Confirm a valid password Verify your email address Allow access to the system.
Exception Condition	Give permission for account access.	

Use Case name	Registration	
Scenario	User (Student) Registration to the system	
Brief Description	Users should register themselves by their own email address.	
Actor	User (Student)	
Pre-condition	Must be login in to the system.	
Post-Condition	Users must confirm registration with email verification	
Flow of Condition	User	System

	<ul style="list-style-type: none"> • Must provide Valid email. 	<ul style="list-style-type: none"> • Must fill up all required Field • Confirm valid email Address • Confirm unique email address. • Verify confirm password
Exception Condition	Get permission to create an account.	

Use Case name	Assign Suggestion	
Scenario	Users (Student) can create a suggestion if any problem occurs in campus	
Brief Description	Users should assign a valid suggestion, what admin should change	
Actor	User (Student)	
Pre-condition	Must be login in to the system.	
Post-Condition	Users must confirm the suggestion will be a big change for the campus.	
Flow of Condition	User	System
	<ul style="list-style-type: none"> • Must provide Valid reason. 	<ul style="list-style-type: none"> • Must fill up all required Field • Make sure the reason is valid.
Exception Condition	Get permission to add a suggestion to the system.	

Use Case name	View Complain	
Scenario	Users (Student) can check their added complain list on the system.	
Brief Description	Users can view their complain status, and added complain list.	
Actor	User (Student)	

Pre-condition	Must be login in to the system	
Post-Condition	Users must be at least one complain in our system.	
Flow of Condition	User	System
	<ul style="list-style-type: none"> • Must add a complaint before to check the complaint list. • Must add valid information 	<ul style="list-style-type: none"> • Provide specific User's complaint list from system
Exception Condition	View complaint list of users.	

Use Case name	View Suggestion	
Scenario	Users (Student) can check their added Suggestion list on the system.	
Brief Description	Users can view their suggestion status, and added suggestion list.	
Actor	User (Student)	
Pre-condition	Must be login in to the system	
Post-Condition	Users must be at least one suggestion in our system.	
Flow of Condition	User	System
	<ul style="list-style-type: none"> • Must add a suggestion before to check the suggestion list. • Must add valid information 	<ul style="list-style-type: none"> • Provide specific User's suggestion list from system
Exception Condition	View suggestion list of users.	

Use Case name	View Notice Board
----------------------	-------------------

Scenario	Users (Student) can check notice list on the system.	
Brief Description	Users can view university any published notice immediately.	
Actor	User (Student)	
Pre-condition	Must be login in to the system	
Post-Condition	Users should check all notice board.	
Flow of Condition	User	System
	<ul style="list-style-type: none"> Must check all notice in notice board. 	<ul style="list-style-type: none"> Provide specific User's notice list from system.
Exception Condition	View notice list of users.	

Use Case name	Get help	
Scenario	Users (Student) can get help on the system.	
Brief Description	Users can get help by chat with helpline.	
Actor	User (Student)	
Pre-condition	Must be login in to the system	
Post-Condition	Users will chat first with helpline.	
Flow of Condition	User	System
	<ul style="list-style-type: none"> User will message first 	<ul style="list-style-type: none"> Provide specific User's solution by helpline.

Exception Condition	Get help from helpline.
----------------------------	-------------------------

3.1.2 Use Case for Admin

Use Case name	Login	
Scenario	Admin logged on to the system	
Brief Description	Admin have a specific account defined by system, then he will be able to access the system	
Actor	Admin	
Pre-condition	Must be use define valid email and password	
Post-Condition	Admin must have power to use system.	
Flow of Condition	Admin	System
	<ul style="list-style-type: none"> Admin must have power to use system 	<ul style="list-style-type: none"> Confirm valid email address. Fetch user's data.
Exception Condition	Access to manage accounts and others access.	

Use Case name	View Complain	
Scenario	Admin will show the user Complain list.	
Brief Description	Admin will see the complaint list that provide by user.	
Actor	Admin	
Pre-condition	Admin must be login his/her Admin Dashboard.	
Post-Condition	Admin must have a good ethics.	
Flow of Condition	Admin	System

	<ul style="list-style-type: none"> • Check all data • Isolate suspicious data. 	<ul style="list-style-type: none"> • Fetch user's complain list.
Exception Condition	Access to see user's complain list.	

Use Case name	View Suggestion	
Scenario	Admin will show the user suggestion list.	
Brief Description	Admin will see the suggestion list that provide by user.	
Actor	Admin	
Pre-condition	Admin must be login his/her Admin Dashboard.	
Post-Condition	Admin must have a good ethics.	
Flow of Condition	Admin	System
	<ul style="list-style-type: none"> • Check all data • Isolate suspicious data. 	<ul style="list-style-type: none"> • Fetch user's suggestion list.
Exception Condition	Access to see user's Suggestion list.	

Use Case name	Solve the problem
Scenario	Admin will solve the problem.
Brief Description	Admin will solve the problem which is complain by user physically
Actor	Admin

Pre-condition	Admin must be login his/her Admin Dashboard.	
Post-Condition	Admin must have a good ethics.	
Flow of Condition	Admin	System
	<ul style="list-style-type: none"> Solve the problem which is in list. 	<ul style="list-style-type: none"> Fetch user's complain list.
Exception Condition	Admin will solve the problem.	

Use Case name	Approve Complain/Suggestion	
Scenario	Admin will approve the valid complain/suggestion	
Brief Description	Admin will check the complaint was valid reason or not, then he/she will accept the complain and try to solve it.	
Actor	Admin	
Pre-condition	Admin must be login his/her Admin Dashboard.	
Post-Condition	Admin must have power to use system.	
Flow of Condition	Admin	System
	<ul style="list-style-type: none"> Admin must accept only the valid complain 	<ul style="list-style-type: none"> Get access to approve the complain. Fetch user's data.
Exception Condition	Access to change the status of approve complain.	

Use Case name	Solve the problem	
Scenario	Admin will solve the problem.	
Brief Description	Admin will solve the problem which is complain by user physically	
Actor	Admin	
Pre-condition	Admin must be login his/her Admin Dashboard.	
Post-Condition	Admin must have a good ethics.	
Flow of Condition	Admin	System
	<ul style="list-style-type: none"> Solve the problem which is in list. 	<ul style="list-style-type: none"> Fetch user's complain list.
Exception Condition	Admin will solve the problem.	

Use Case name	Generate Report	
Scenario	Admin will send a report which complain is accept by him.	
Brief Description	Admin will mail the authority which complain was accepted.	
Actor	Admin	
Pre-condition	Admin must be login his/her Admin Dashboard.	
Post-Condition	Admin should have a strong internet.	
Flow of Condition	Admin	System

	<ul style="list-style-type: none"> • Send a mail to authority. 	<ul style="list-style-type: none"> • Send the mail immediately to the authorities.
Exception Condition	Admin will success to mail a report to the authority.	

Use Case name	View User List	
Scenario	Admin will view all user details	
Brief Description	Admin will check all user details and manage.	
Actor	Admin	
Pre-condition	Admin must be login his/her Admin Dashboard.	
Post-Condition	Admin should have a strong ethics.	
Flow of Condition	Admin	System
	<ul style="list-style-type: none"> • Check valid user. 	<ul style="list-style-type: none"> • Fetch user data. • Manage operation.
Exception Condition	Admin will view and manage user data.	

3.1.3 Use Case for Authority

Use Case name	Login
Scenario	Authority logged on to the system
Brief Description	Authority has a specific account defined by system, then he will be able to access the system

Actor	Authority	
Pre-condition	Must be use define valid email and password	
Post-Condition	Authority must have power to use system.	
Flow of Condition	Authority	System
	<ul style="list-style-type: none"> Authority must have power to use system 	<ul style="list-style-type: none"> Confirm valid email address. Fetch user's data.
Exception Condition	Access to manage accounts and others access.	

Use Case name	View Complain	
Scenario	Authority will show the user Complain list and manage.	
Brief Description	Authority will see the complaint list that provide by user.	
Actor	Authority	
Pre-condition	Authority must be login his/her Admin Dashboard.	
Post-Condition	Authority must have a good ethics.	
Flow of Condition	Authority	System
	<ul style="list-style-type: none"> Check all data Isolate suspicious data. 	<ul style="list-style-type: none"> Fetch user's complain list.
Exception Condition	Access to see user's complain list.	

Use Case name	View Suggestion	
Scenario	Authority will show the user suggestion list.	
Brief Description	Authority will see the suggestion list that provide by user.	
Actor	Authority	
Pre-condition	Authority must be login his/her Admin Dashboard.	
Post-Condition	Authority must have a good ethics.	
Flow of Condition	Authority	System
	<ul style="list-style-type: none"> • Check all data • Isolate suspicious data. 	<ul style="list-style-type: none"> • Fetch user's suggestion list.
Exception Condition	Access to see user's Suggestion list.	

Use Case name	Check Report	
Scenario	Authority will check a report which complain is accept by admin.	
Brief Description	Authority will check mail the which is accept by admin.	
Actor	Authority	
Pre-condition	Authority must be login his/her Admin Dashboard.	
Post-Condition	Authority should have a strong internet.	
Flow of Condition	Authority	System

	<ul style="list-style-type: none"> • Check the mail send by admin. 	<ul style="list-style-type: none"> • Show the mail immediately to the authorities.
Exception Condition	Authority will success to check the mail.	

3.2 Activity Diagram

3.2.1 Activity Diagram for User

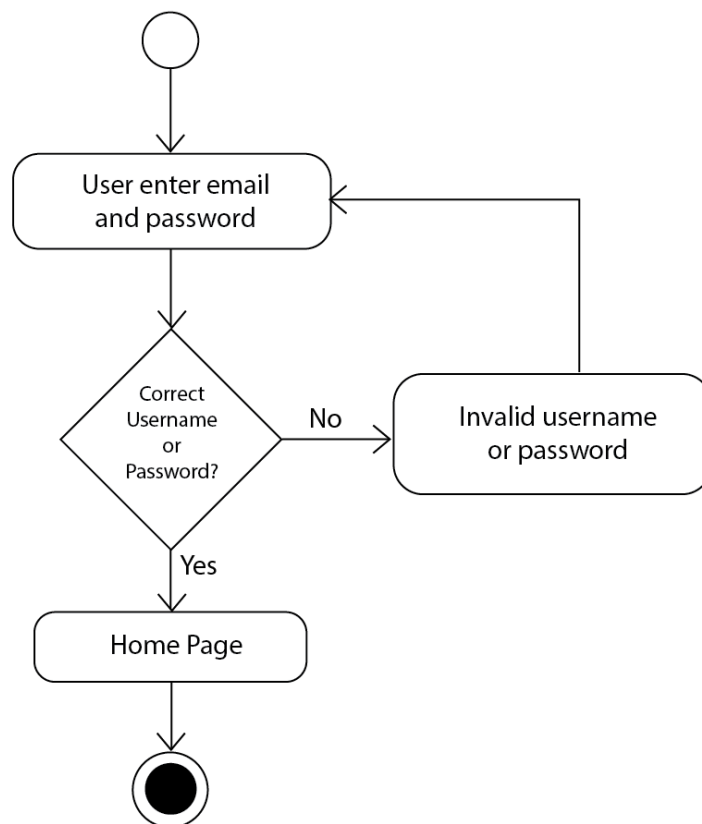


Fig 3.2.1.1 User Login

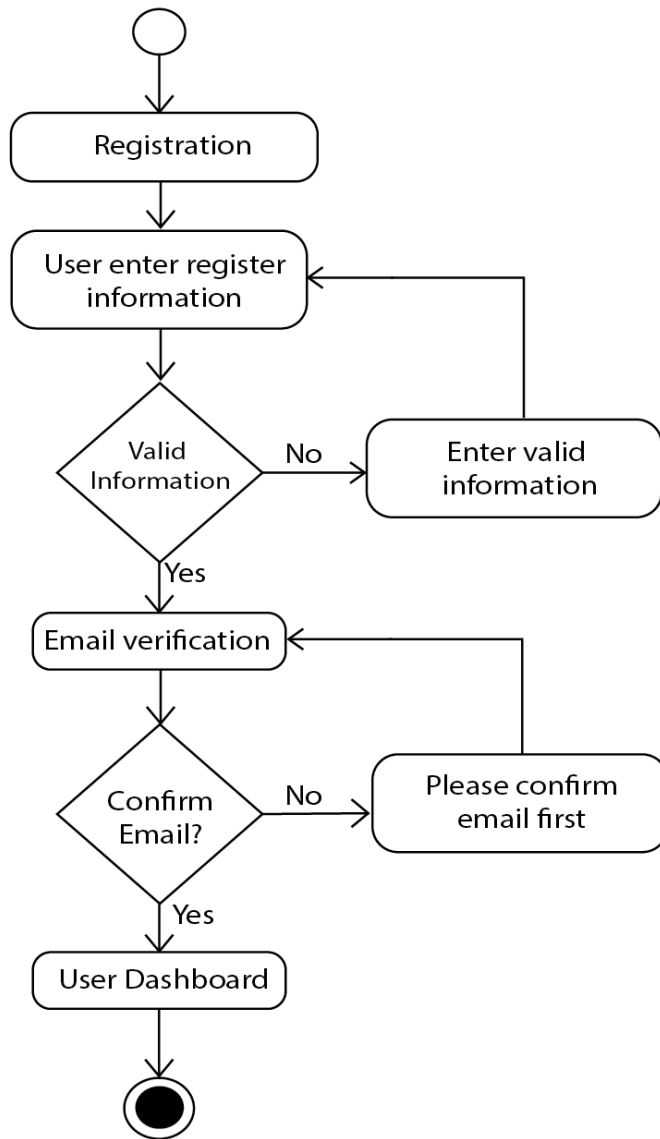


Fig 3.2.1.2 User Registration

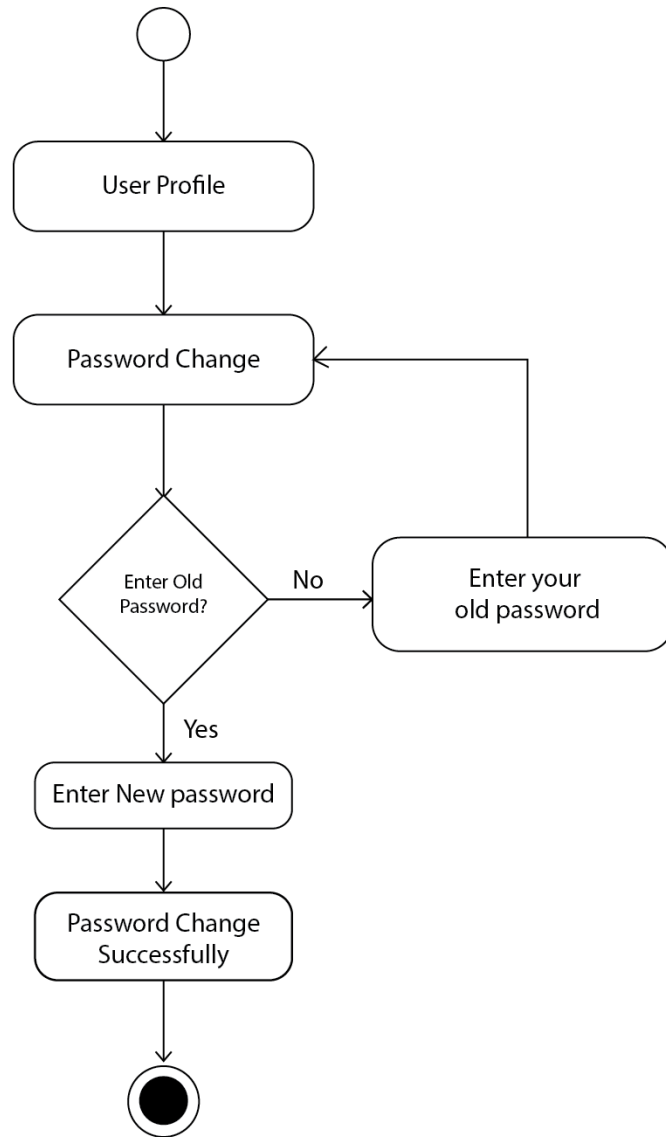


Fig 3.2.1.3 User Password Change

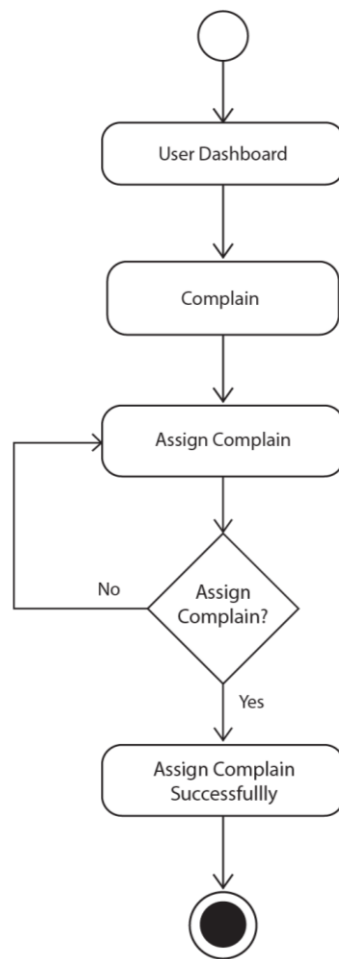


Fig 3.2.1.4 User Assign Complain

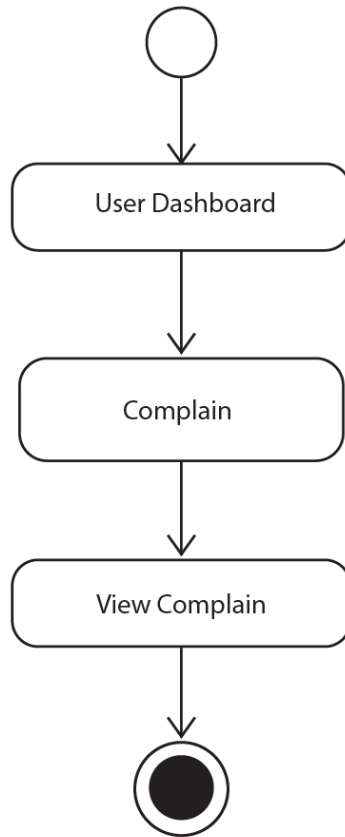


Fig 3.2.1.5 User View Complain

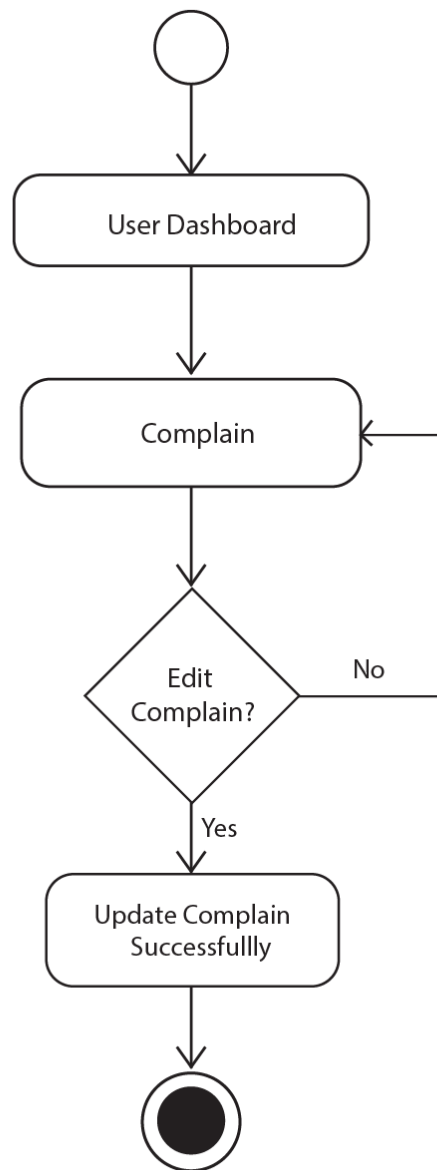


Fig 3.2.1.6 User Edit Complain

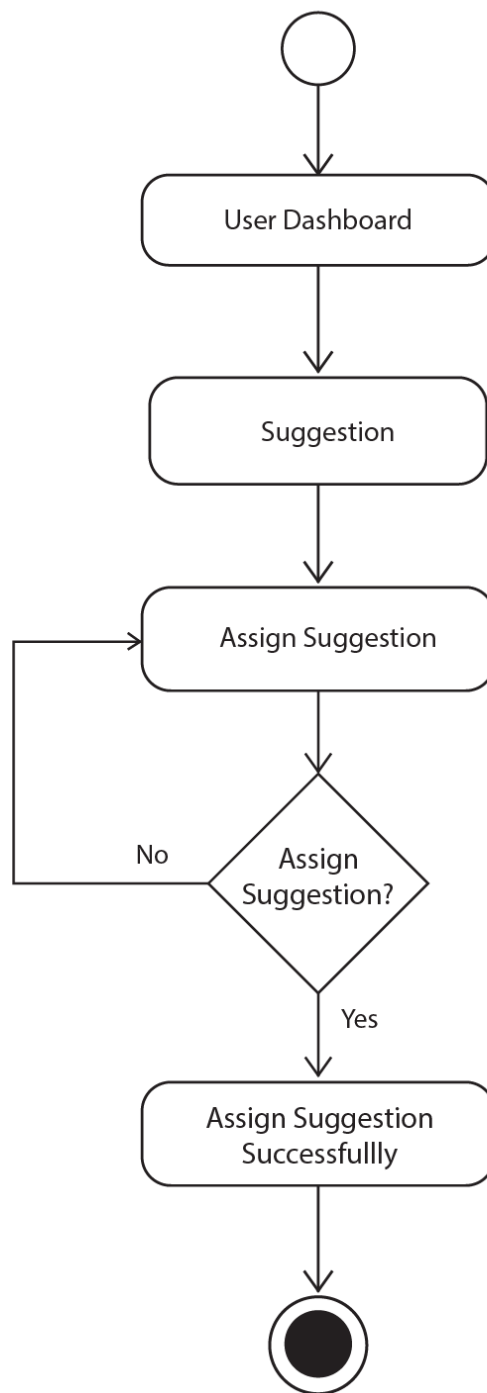


Fig 3.2.1.7 User Assign Suggestion

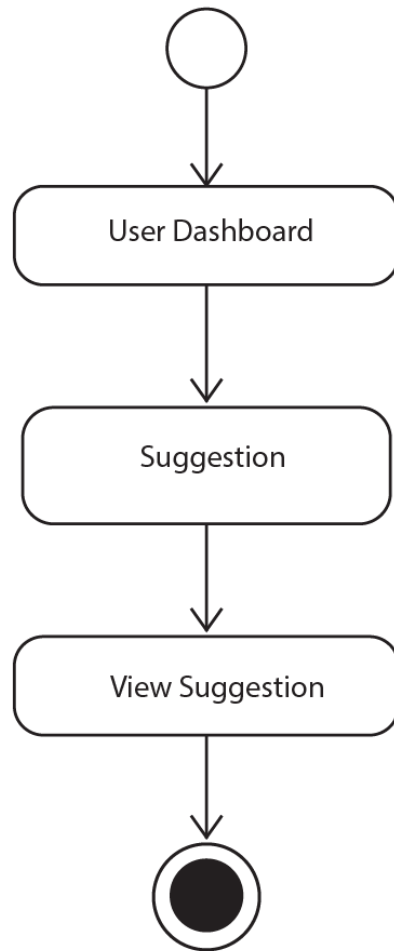


Fig 3.2.1.8 User View Suggestion

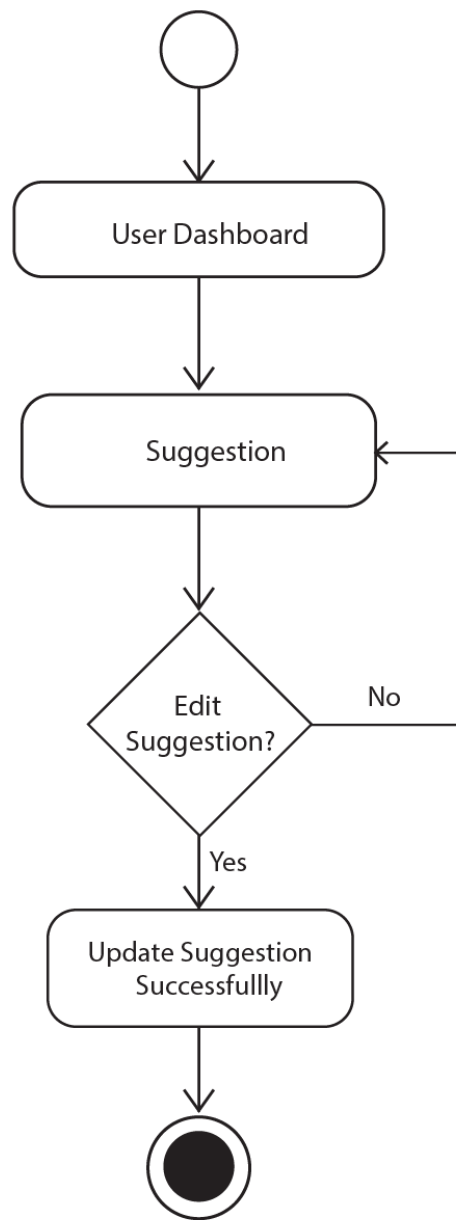


Fig 3.2.1.9 User Edit Suggestion

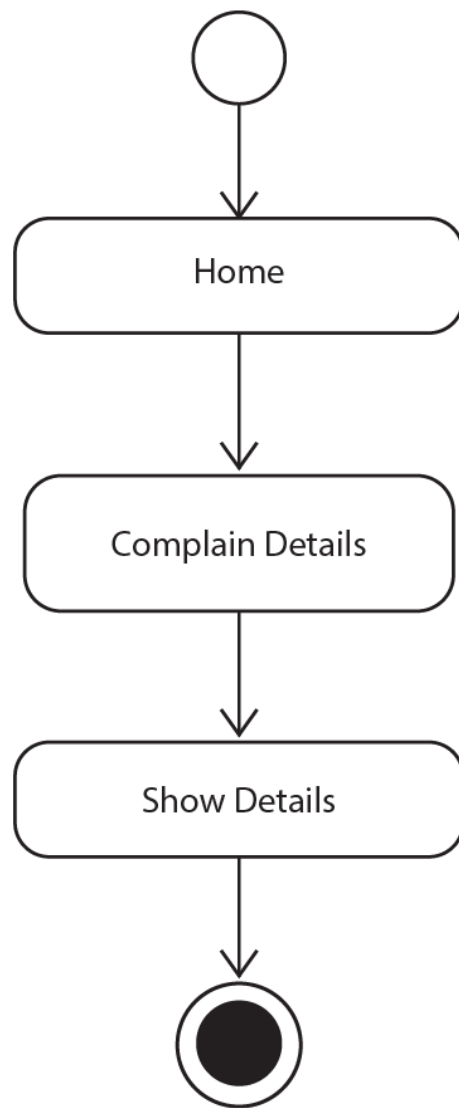


Fig 3.2.1.10 User Complain Details

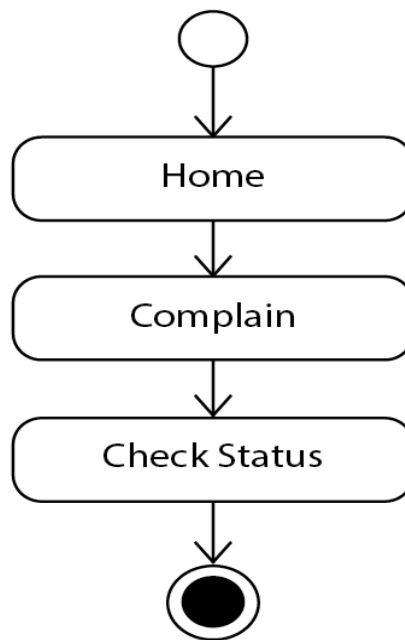


Fig 3.2.1.11 User Complain Status

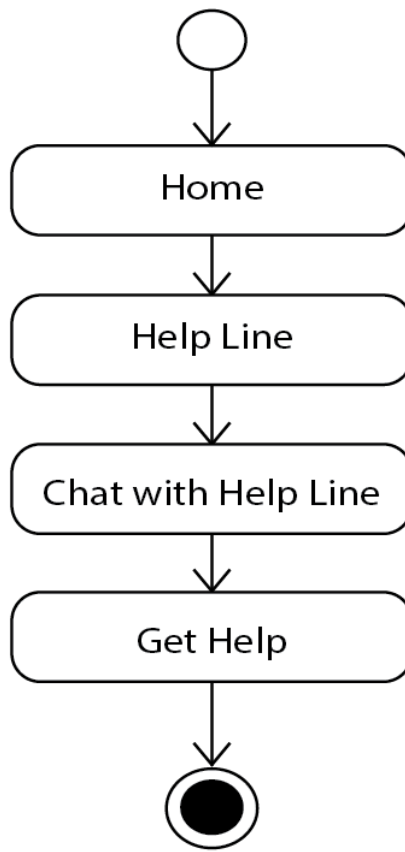


Fig 3.2.1.12 User Get Help

3.2.2 Activity Diagram for Admin

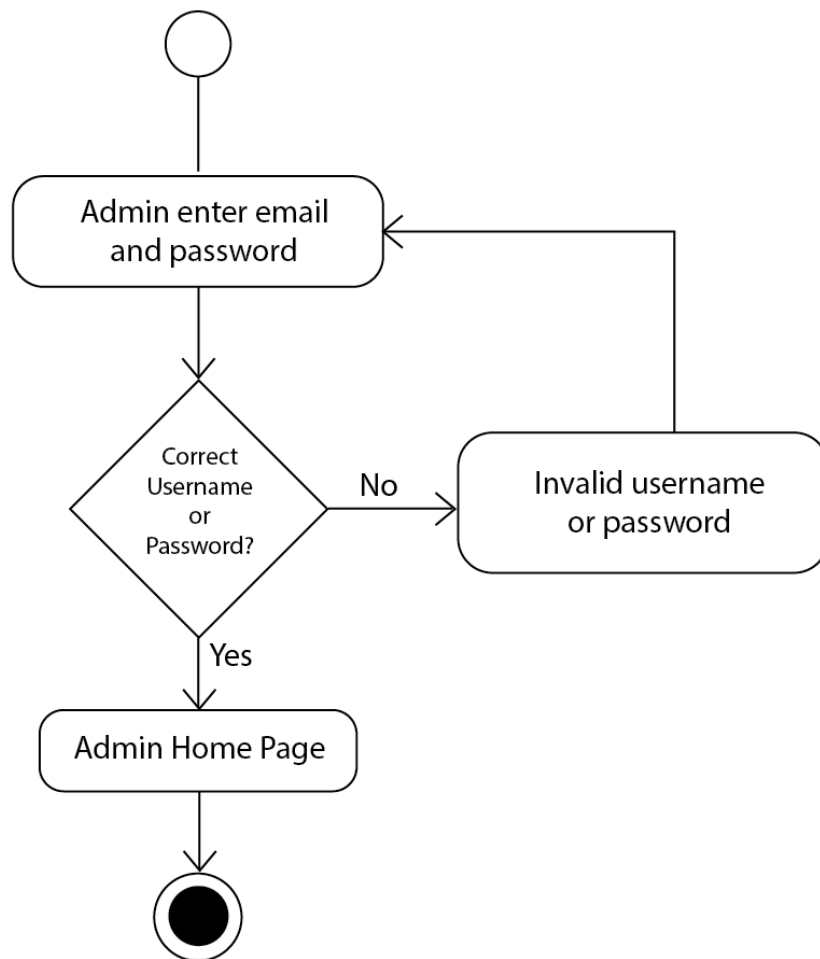


Fig 3.2.2.1 Admin Login

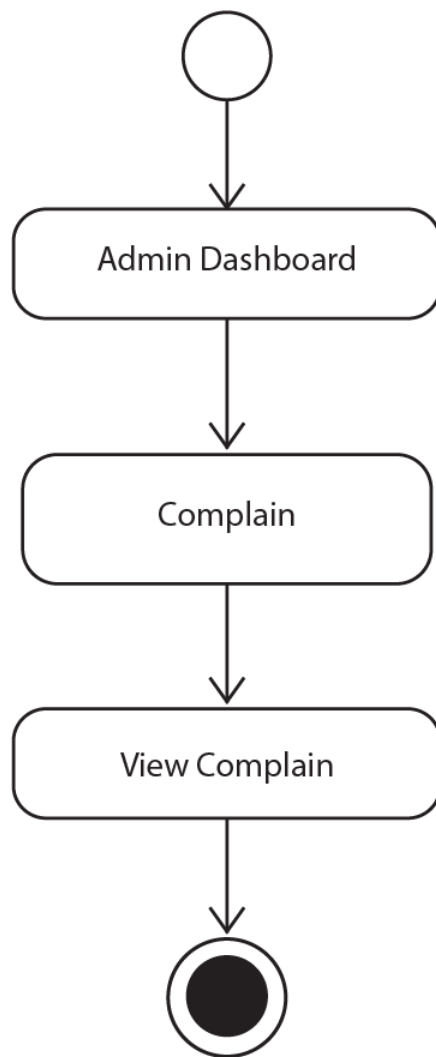


Fig 3.2.2.2 Admin Complain

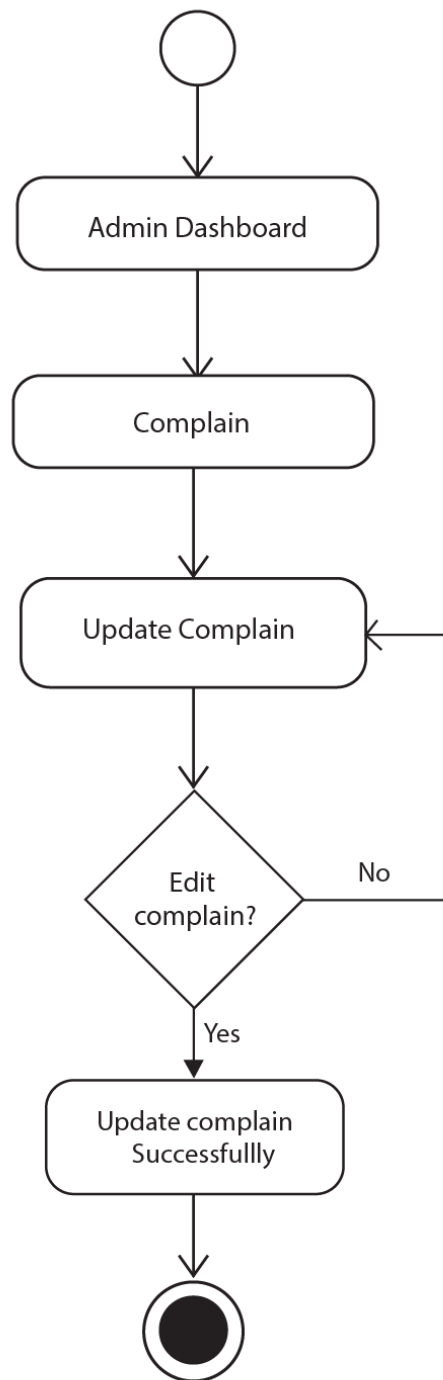


Fig 3.2.2.3 Admin Update Complain

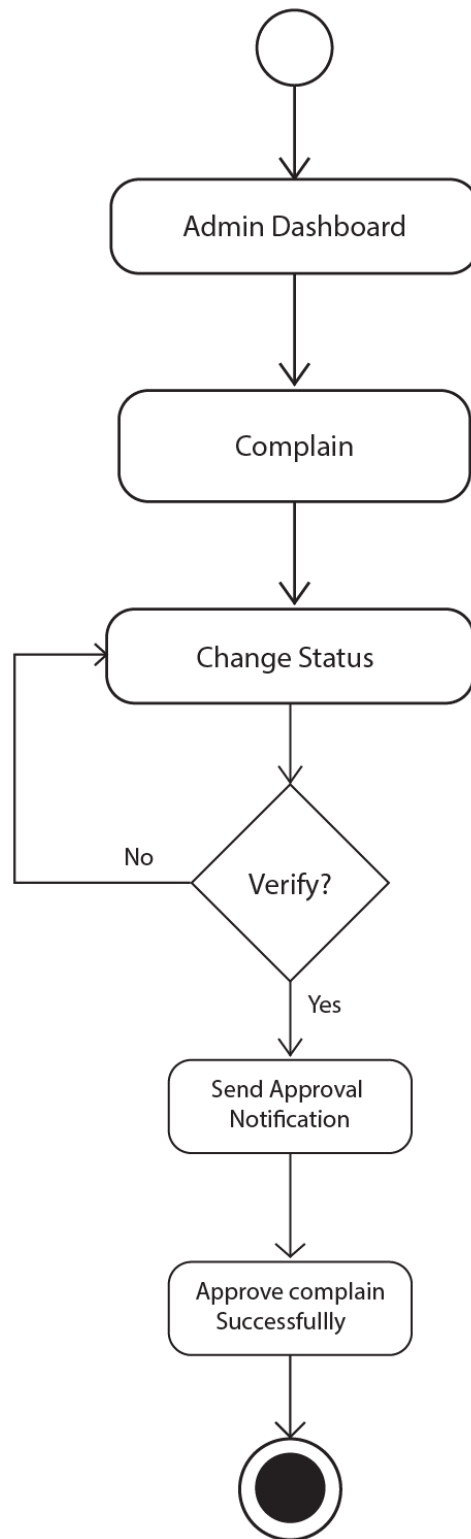


Fig 3.2.2.3 Admin Complain Approve

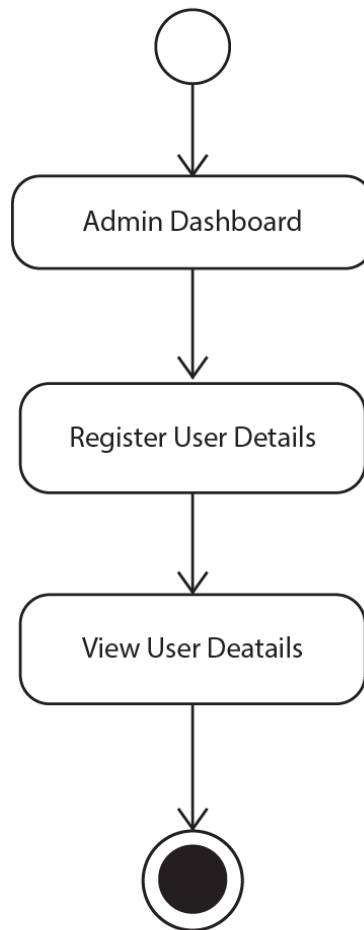


Fig 3.2.2.4 Admin User Details

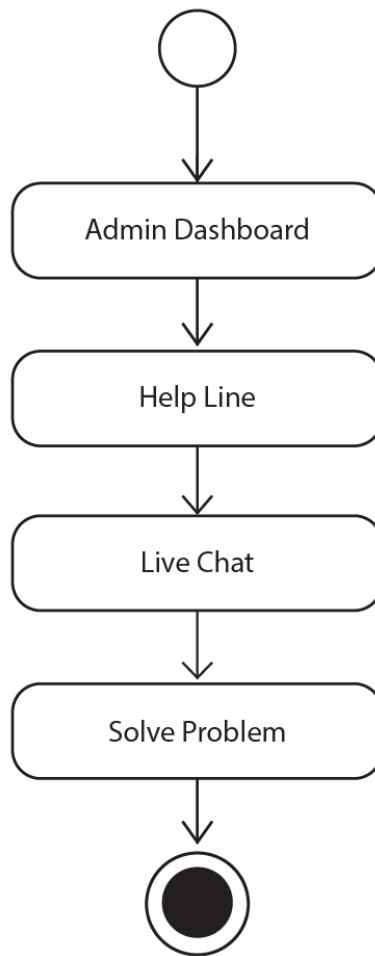


Fig 3.2.2.5 Admin User Help

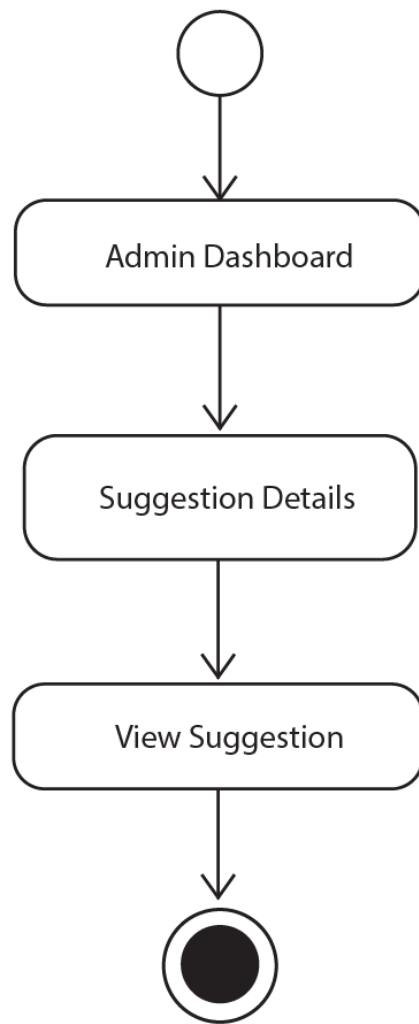


Fig 3.2.2.5 Admin User Suggestion Details

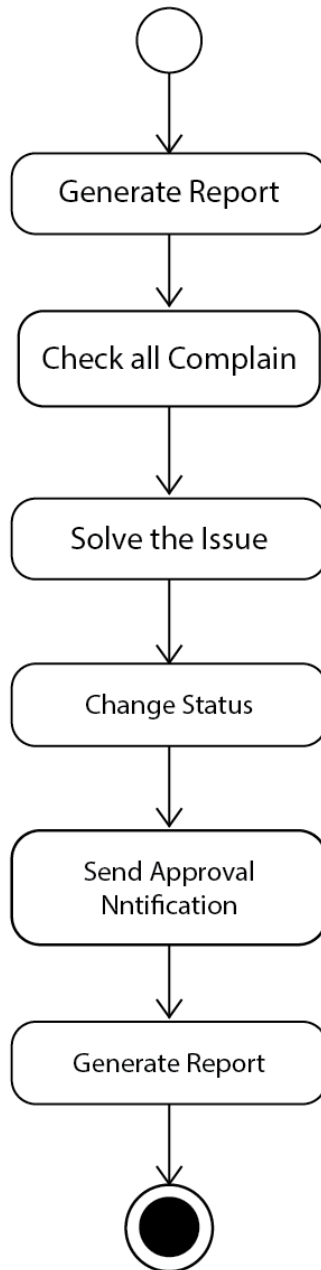


Fig 3.2.2.6 Admin Generate Report

3.2.3 Activity Diagram for Authority

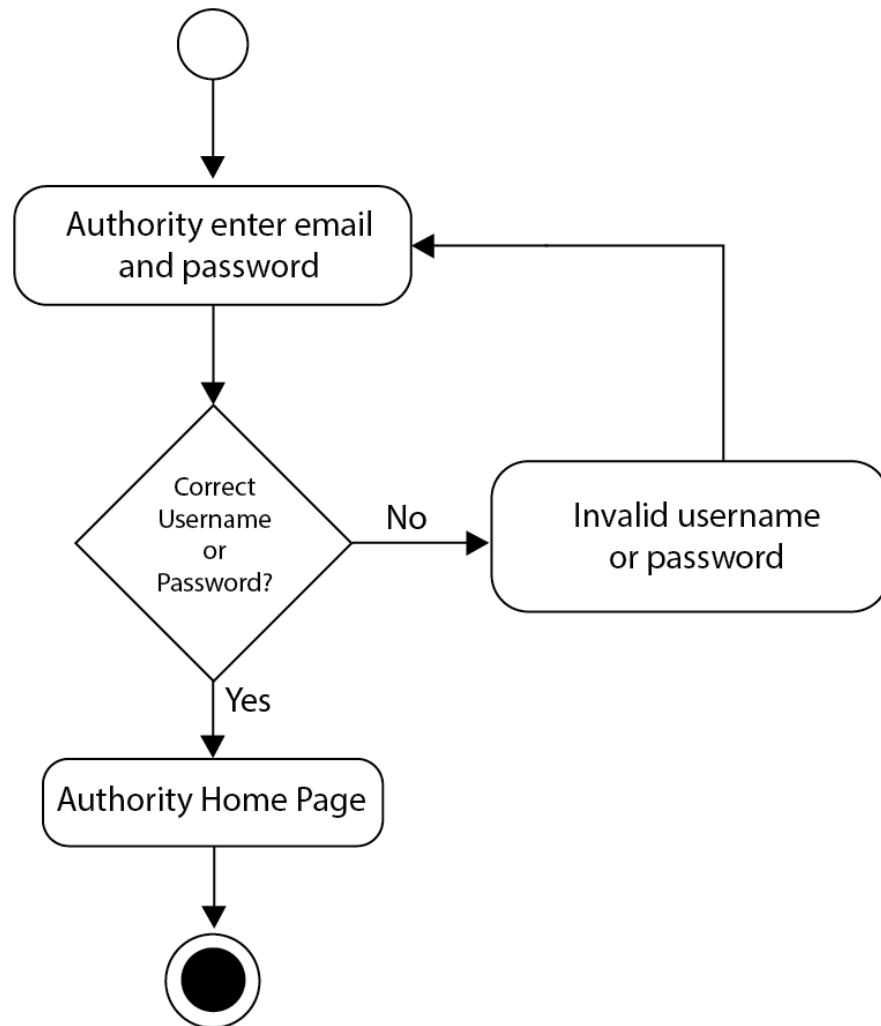


Fig 3.2.3.1 Authority Login

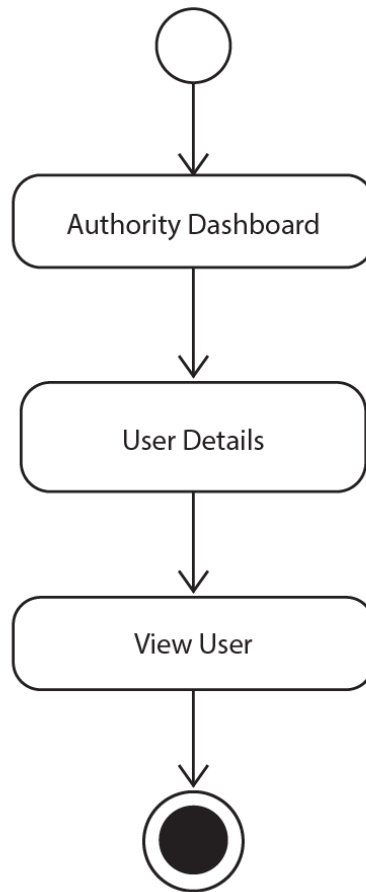


Fig 3.2.3.2 Authority View User

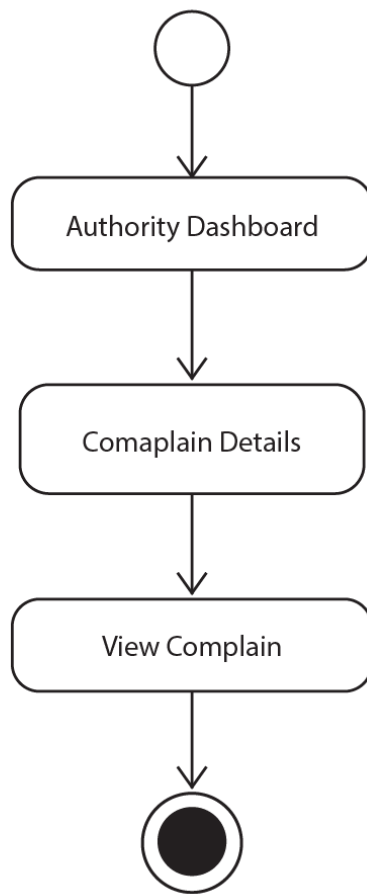


Fig 3.2.3.2 Authority User Complain

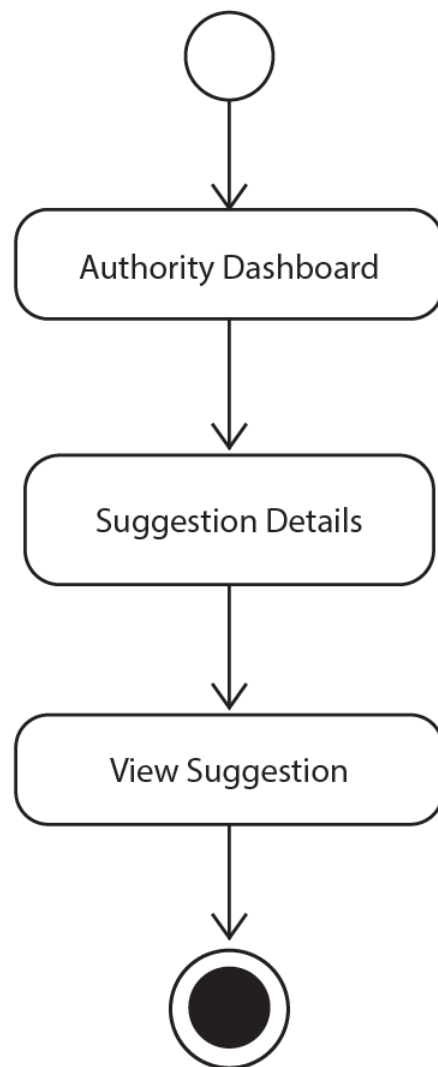


Fig 3.2.3.3 Authority User Suggestion

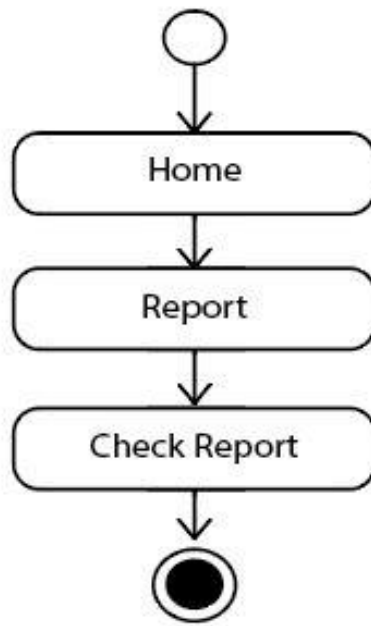


Fig 3.2.3.4 Authority Report check

3.3 Sequence Diagram

3.3.1 Sequence Diagram for User

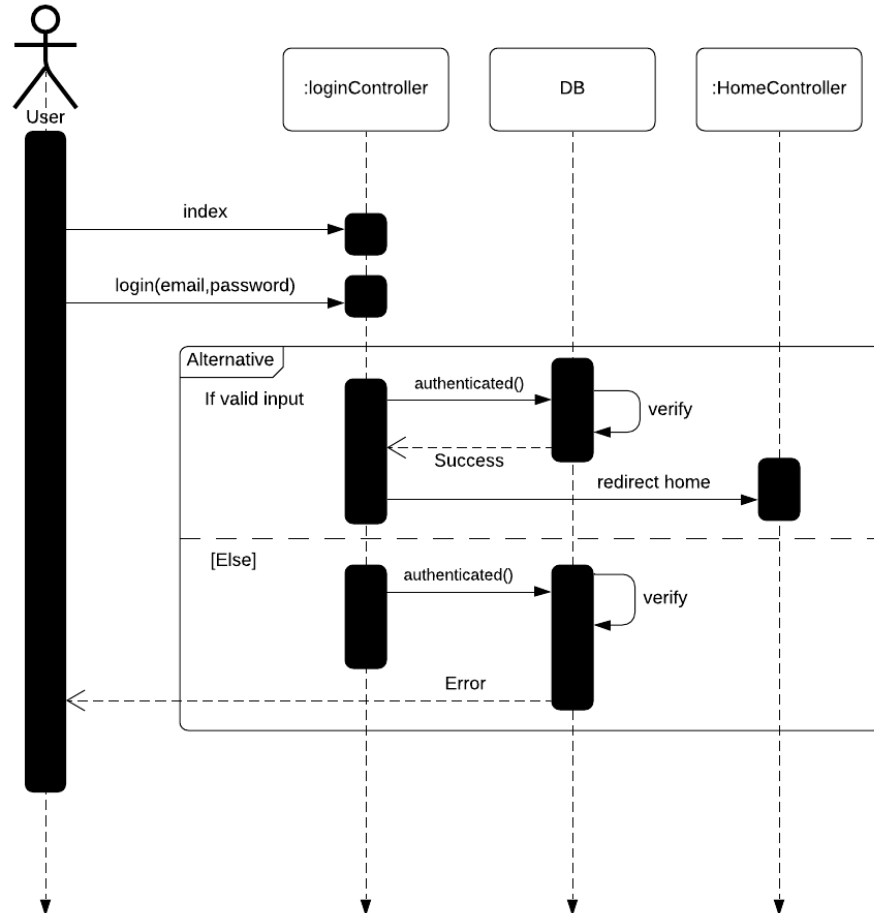


Fig 3.3.1.1 User login

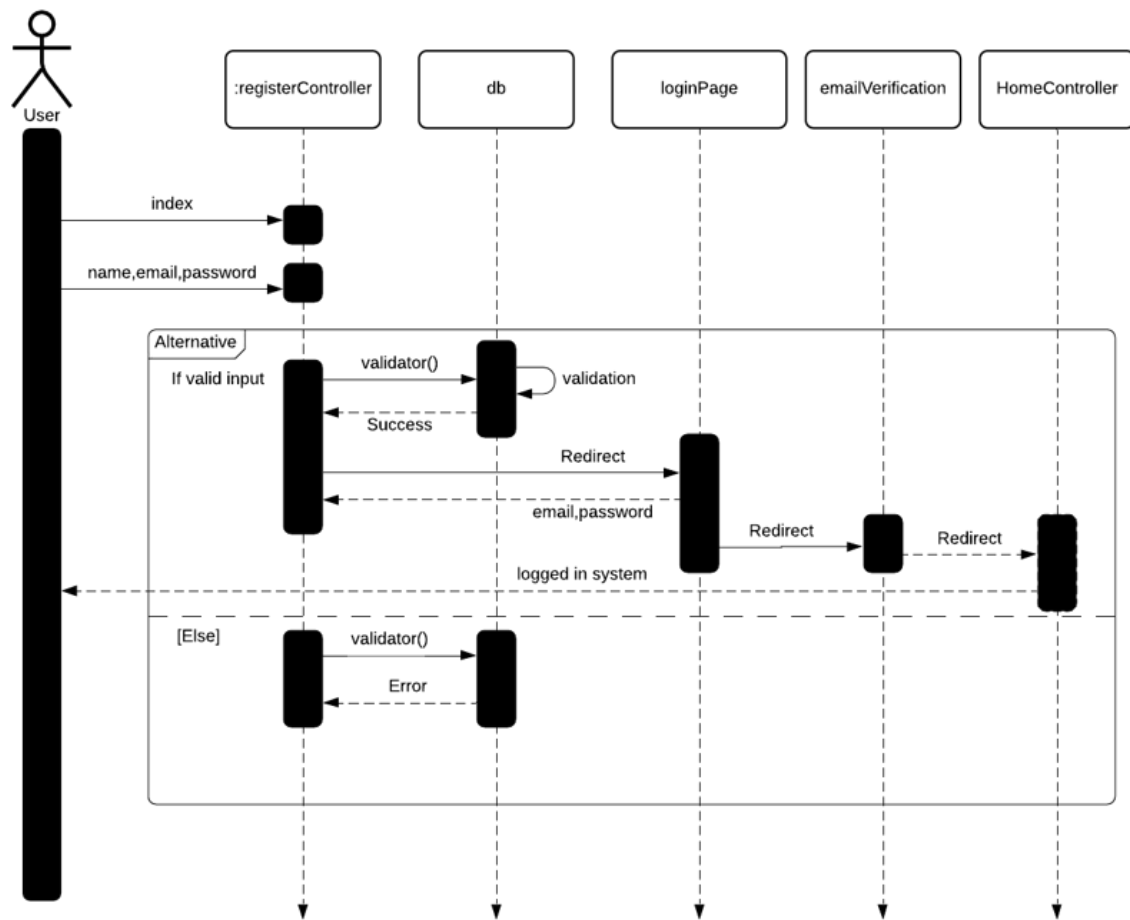


Fig 3.3.1.2 User Registration

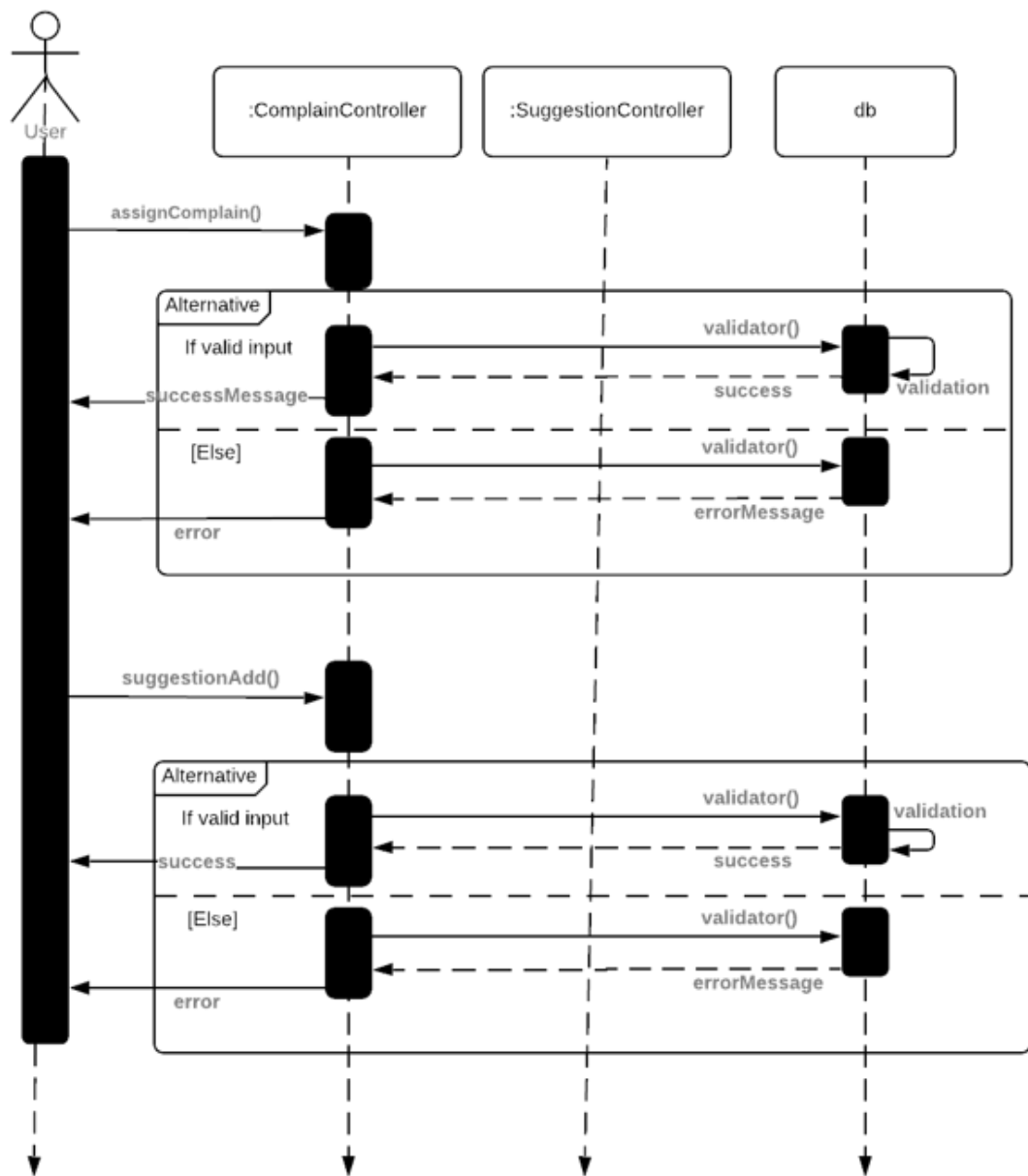


Fig 3.3.1.2 User Complain and Suggestion

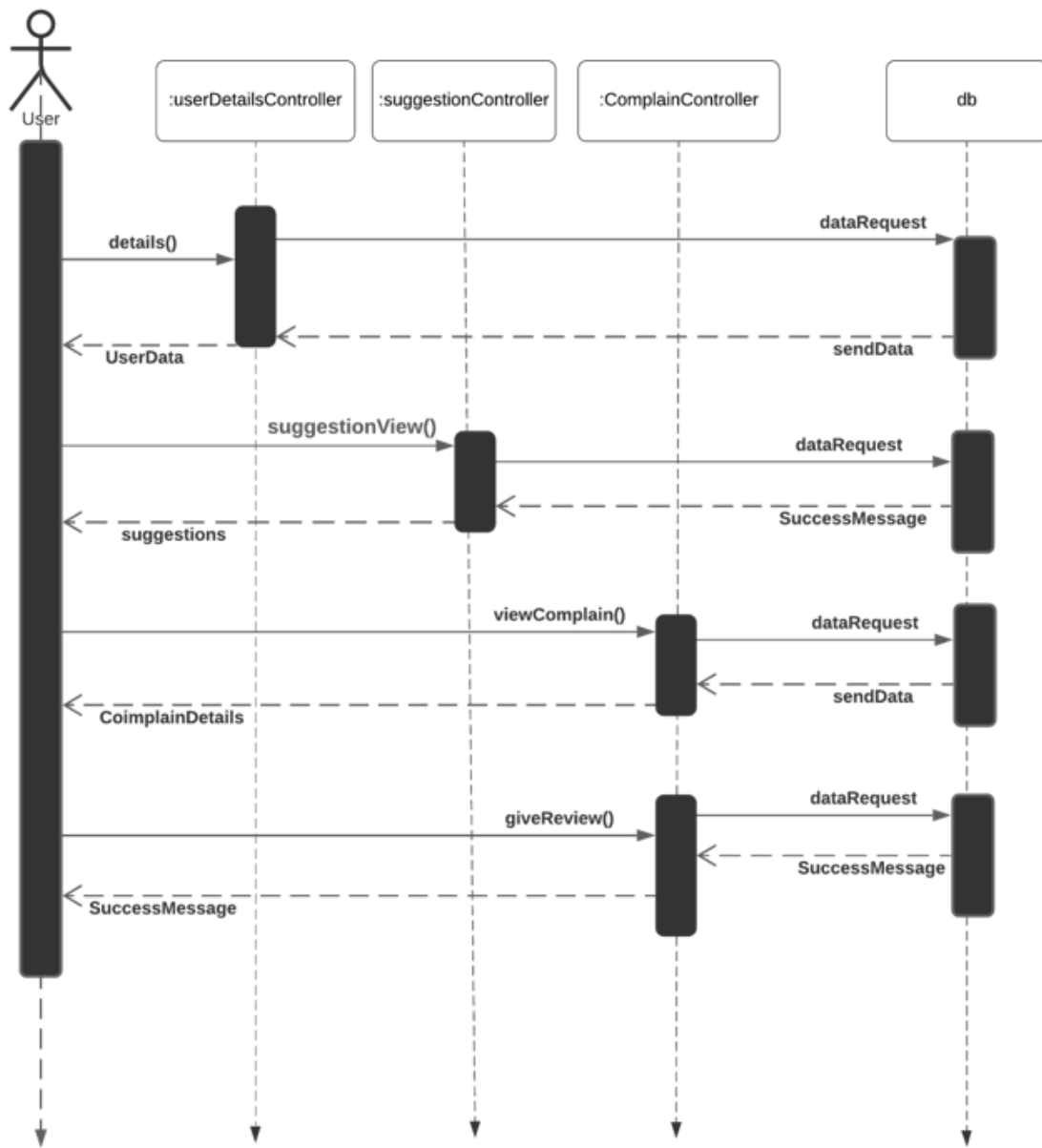


Fig 3.3.1.3 User Main Sequence

3.3.2 Sequence Diagram for Admin

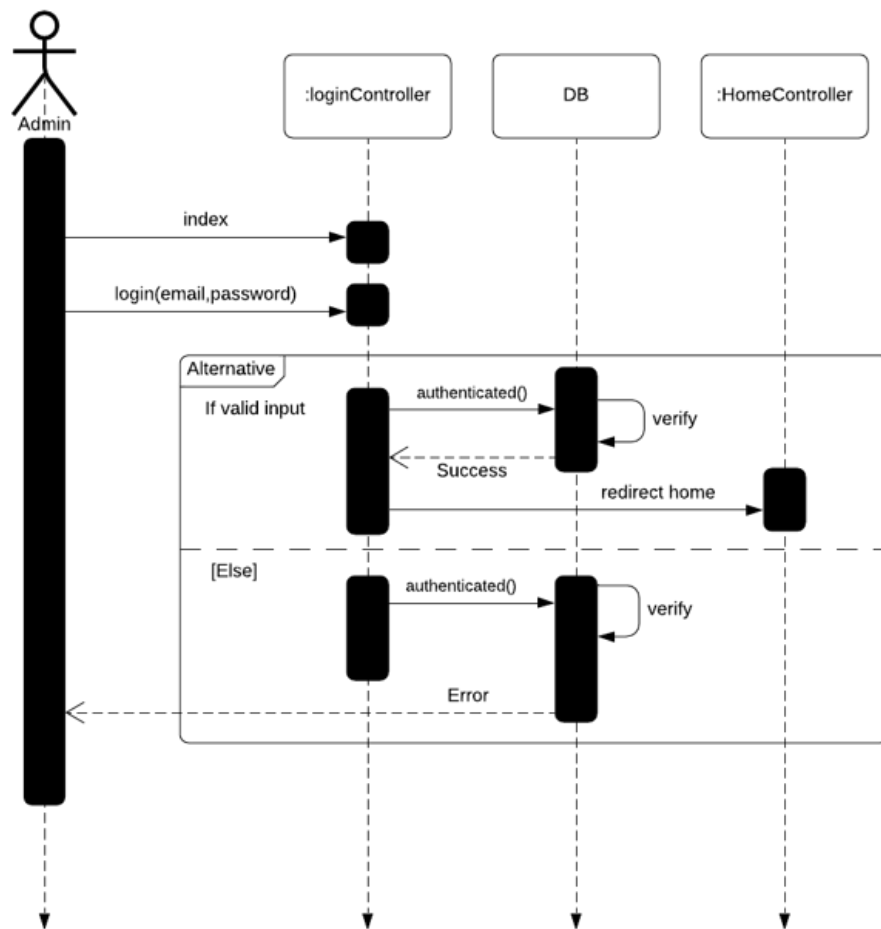


Fig 3.3.2.1 Admin login

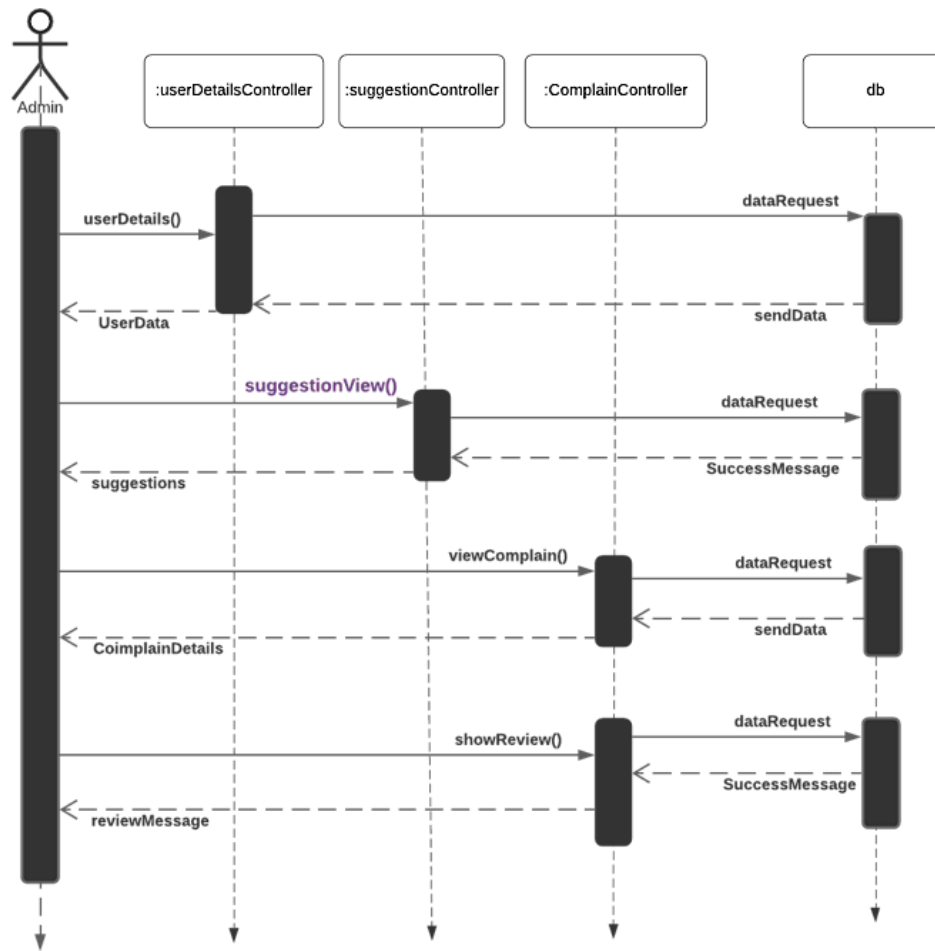


Fig 3.3.2.2 Admin Main Options

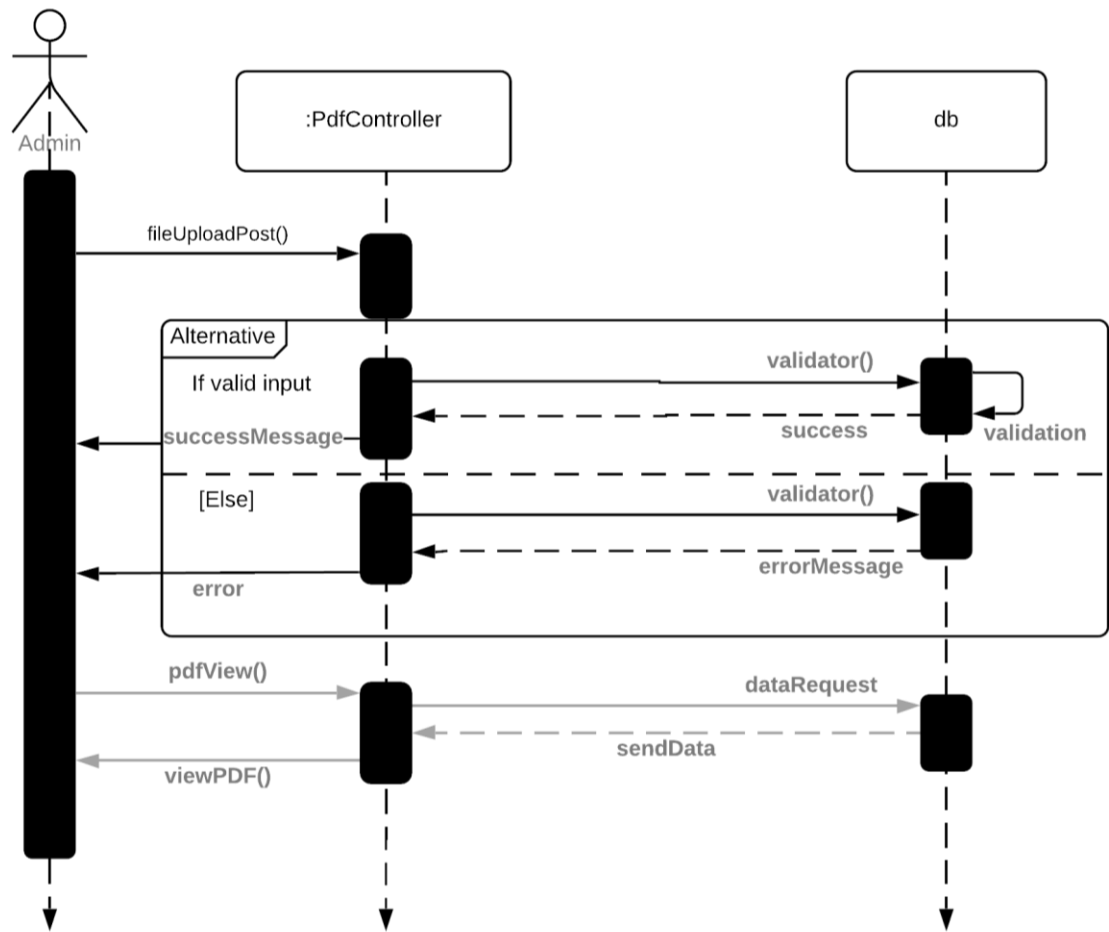


Fig 3.3.2.3 Admin Notice Update

3.3.3 Sequence Diagram for Authority

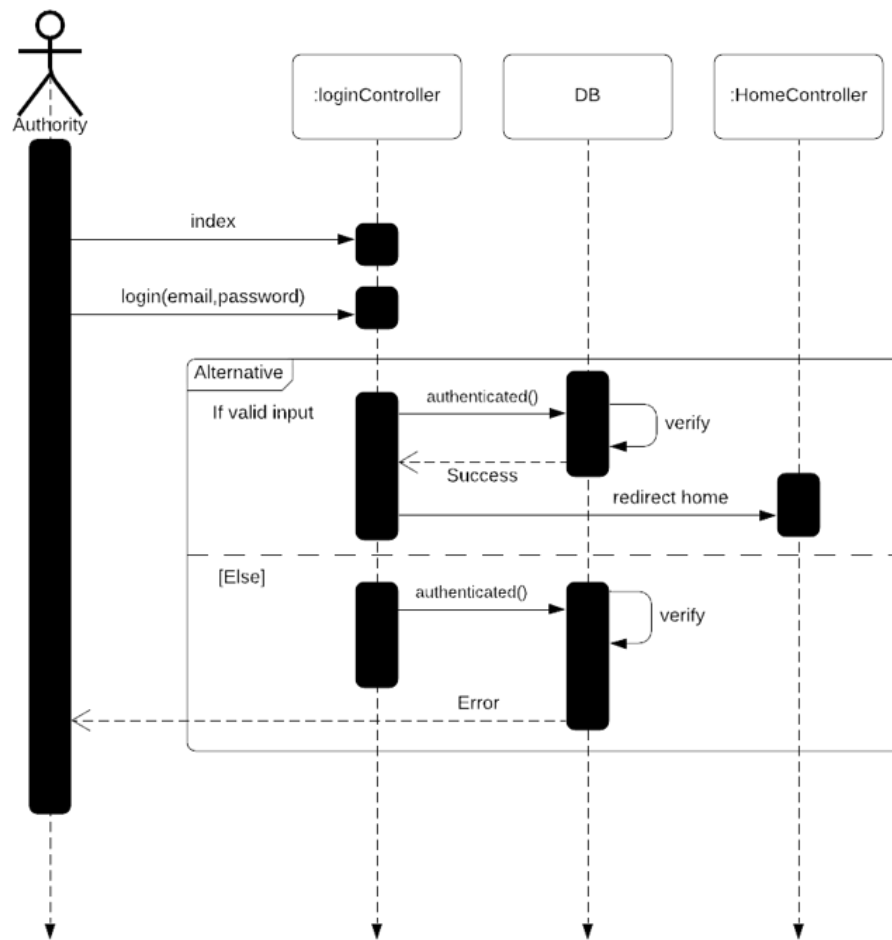


Fig 3.3.3.1 Authority Login

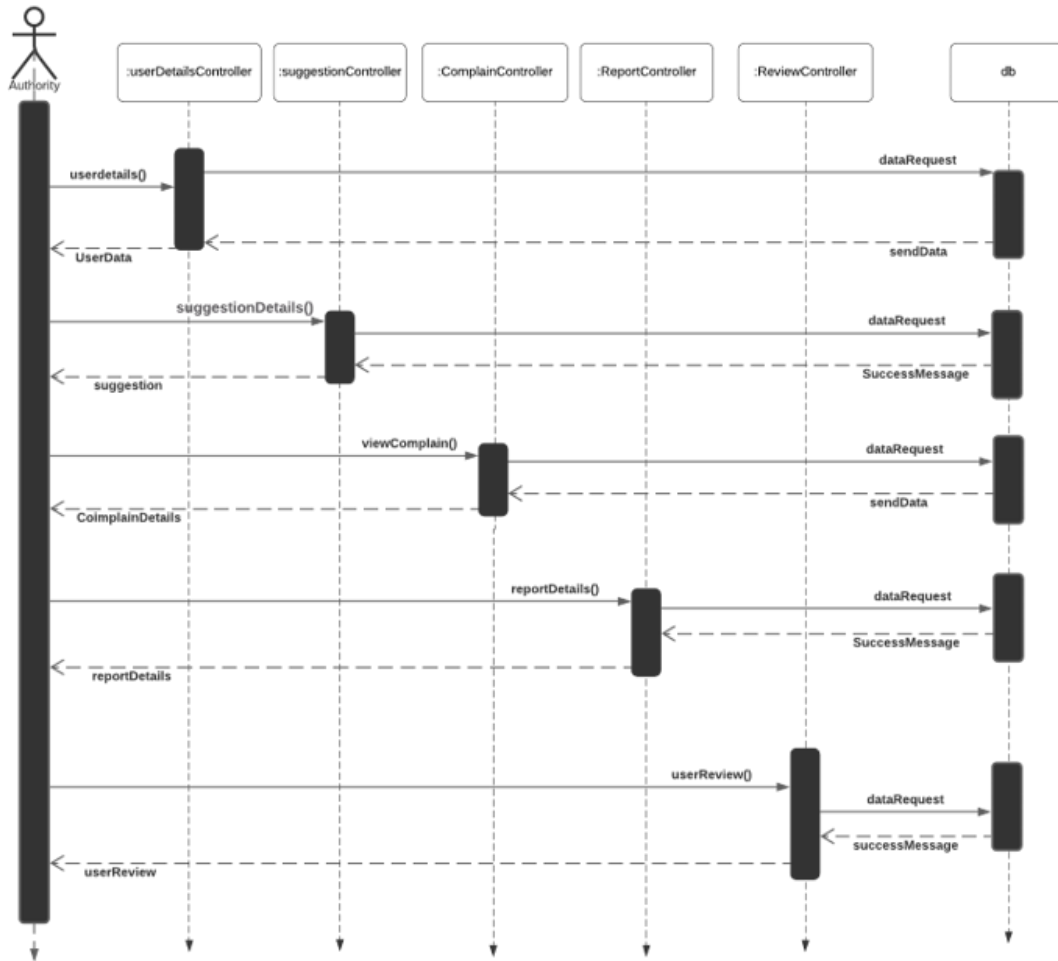


Fig 3.3.3.2 Authority Main Options

3.4 ERD Diagram

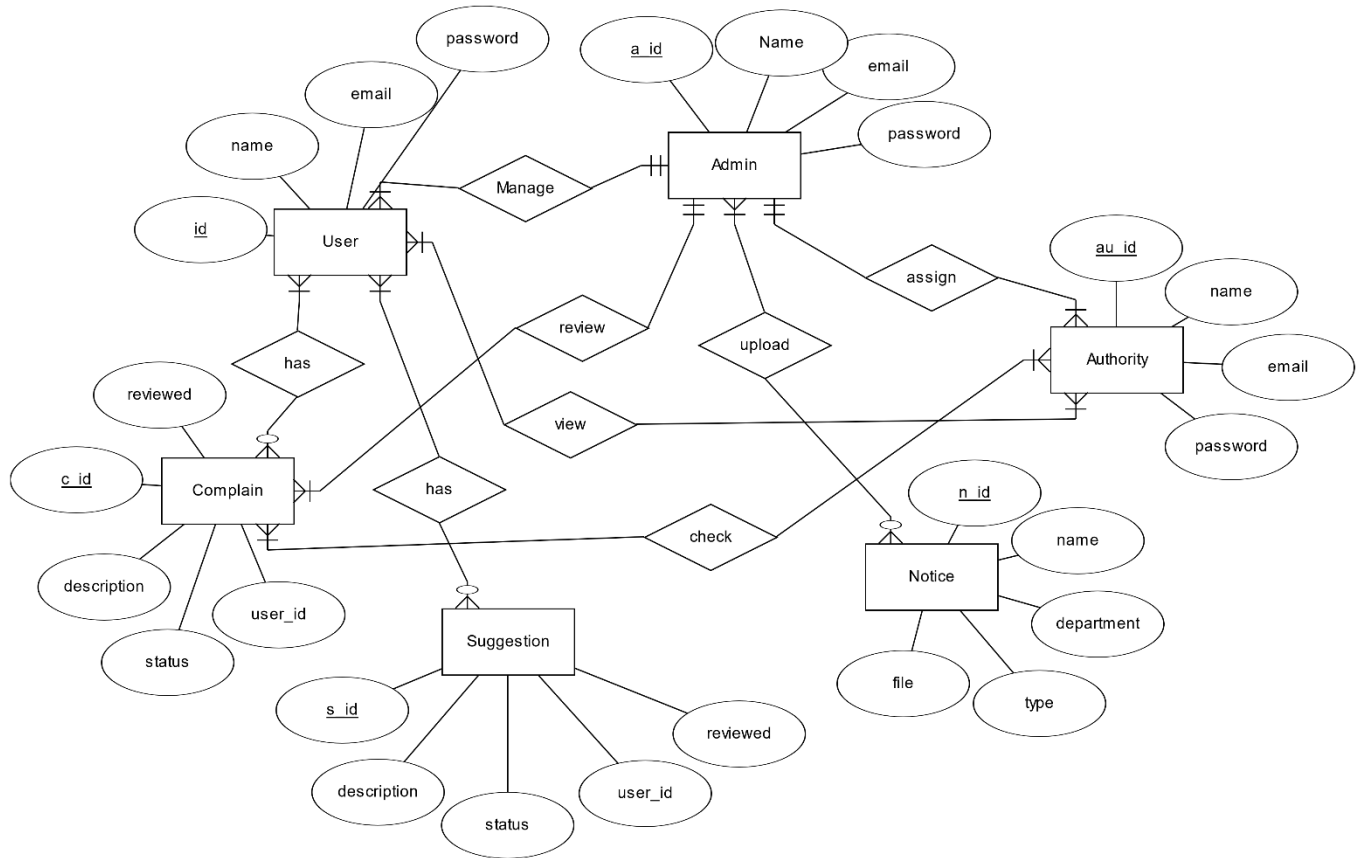


Fig 3.4.1 ERD Diagram [3]

CHAPTER 04

DEVELOPMENT TOOLS & TECHNOLOGY

4.1 User Interface Technologies

- HTML5, CSS, JavaScript, Bootstrap4, PHP, Laravel (Framework)
- jQuery 3.5.1
- Bootstrap
- Font Awesome, Material Icons

4.2 Implementation Technologies

4.2.1 Xampp (7.4.16)

With its core components being the Apache HTTP Server, MariaDB database, and script interpreters for the PHP and Perl programming languages, Xampp is a free open-source web server solution built on the Apache platform. [4]

4.2.2 NPM(6.X)

NPM is the biggest software registry and a package management for JavaScript. Find reusable code packages, then mix them in novel ways. [5]

4.2.3 PHP 7.3.27

PHP employs hypertext. A general-purpose language for Web development is known as Preprocessor. [6]



4.3 Platform & Environment

4.3.1 Hardware

- Processor: Intel core i3 (Minimum)
- RAM:4GB (Minimum)
- Hard Drive:1TB
- Ubuntu 14.04./ Windows 8.1/Windows 10

4.3.2 Tools

- IDE: PhpStorm
- Command Prompt
- Server: LocalHost:8000

4.3.3 Version Control

- Git

CHAPTER 05

SYSTEM TESTING

5.1 Introduction

The purpose of system testing is to find flaws. This activity's major goal is to verify that the procedure is effectively conveyed and that all requirements are satisfied. System testing is important for assessing whether or not a function is functioning properly.

5.2 Test Case

A test case is a sequence of situations or depend on the particular will decide whether or not the system under test satisfied the requirement or worked properly. The developing process test case can assist in identifying an issue in an application's requirement or design.

Test Case No -1

Test Case ID.1	Module name: User Login
Test Priority: High	Test Date:07.12.2021
Test Title: User login	Test executed by: Omer Faruq Nadim
Description: Test User login page	Test executed date: 07.12.2021
Pre-condition:	Users must have valid email and password.
Test steps:	<ol style="list-style-type: none">1. Go to login page2. Provide valid email & password3. Click Login button
Test Data:	User: Email: Nadim@gmail.com Password: Test12345
Expected Results:	User should able to login
Actual Result:	User logged in successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully Logged in.

Testing Case, No-2

Test Case ID.2	Module name: User Register
Test Priority: High	Test Date:22.10.2022
Test Title: User Register	Test executed by: Omer Faruq Nadim
Description: Test User Register	Test executed date: 22.10.2022
Pre-condition:	Users must have valid email and password.
Test steps:	<ol style="list-style-type: none"> 1. Go to Register page 2. Provide valid information 3. Click Register button
Test Data:	User: Email: Nadim@gmail.com Password: Test12345
Expected Results:	User should confirm email verification and logged on system
Actual Result:	User Register successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully Register.

Testing Case, No-3

Test Case ID.3	Module name: User Complain
Test Priority: High	Test Date:22.10.2022
Test Title: User Complain Section	Test executed by: Omer Faruq Nadim
Description: Test User create complain page	Test executed date: 24.10.2022
Pre-condition:	Users must enter valid information.
Test steps:	<ol style="list-style-type: none"> 1. Go to create complain page 2. Provide valid information and image. 3. Click submit button
Test Data:	Require Information

Expected Results:	User should able to create complain.
Actual Result:	User create complain successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully create complain.

Testing Case, No-4

Test Case ID.3	Module name: User Suggestion
Test Priority: High	Test Date:22.10.2022
Test Title: User suggestion Section	Test executed by: Omer Faruq Nadim
Description: Test User create suggestion page	Test executed date: 24.10.2022
Pre-condition:	Users must enter valid information.
Test steps:	<ol style="list-style-type: none"> 1. Go to create suggestion page 2. Provide valid information. 3. Click submit button
Test Data:	Require Information
Expected Results:	User should able to create suggestion.
Actual Result:	User create suggestion successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully create suggestion.

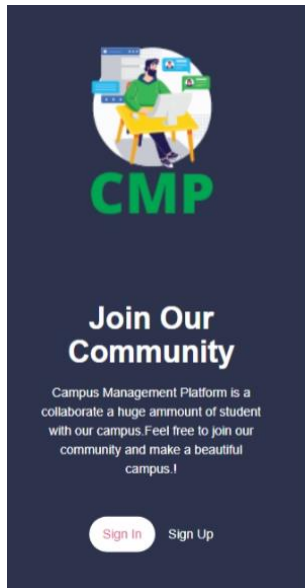
CHAPTER 06

USER MANUAL

6.1 User manual

FOR User

6.1.1 User login page [7]



Sign In To Campus Management Platform

Email

Password

☐

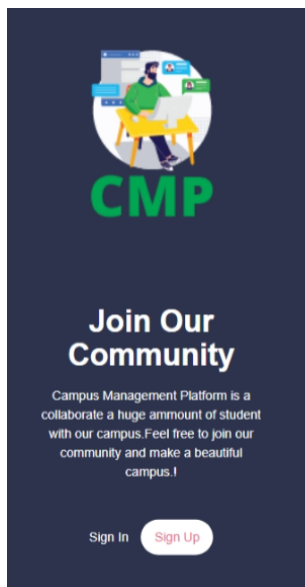
Remember me

[Forgot Your Password? ?](#)

Login

Fig 6.1.1 User Login page

6.1.2 User Registration page



Create An Account

Name

Email

Password

Confirm Password

☐

I Accept Terms and Conditions

Register

Fig 6.1.2 User Registration page

6.1.3 User Dashboard

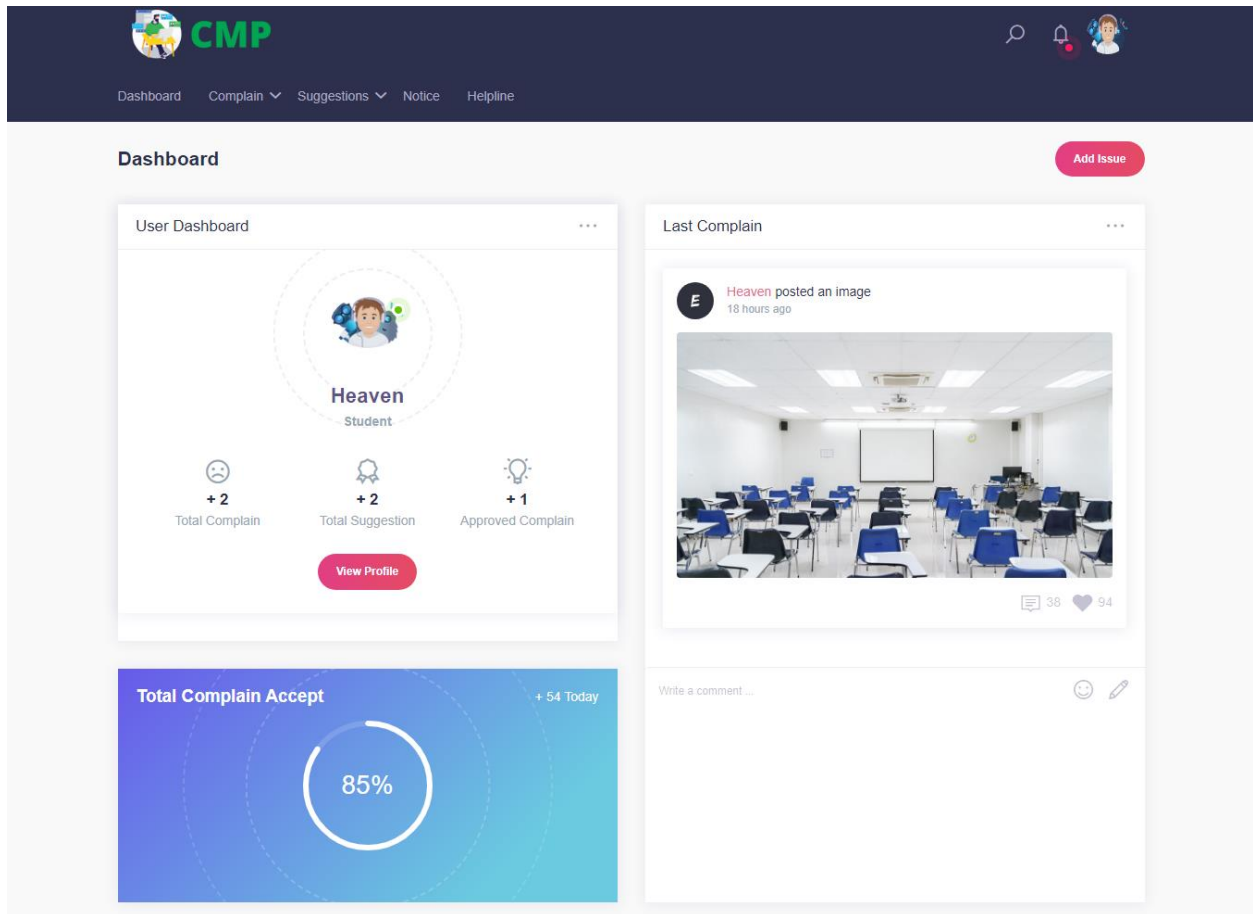


Fig 6.1.3.1 User Dashboard 1

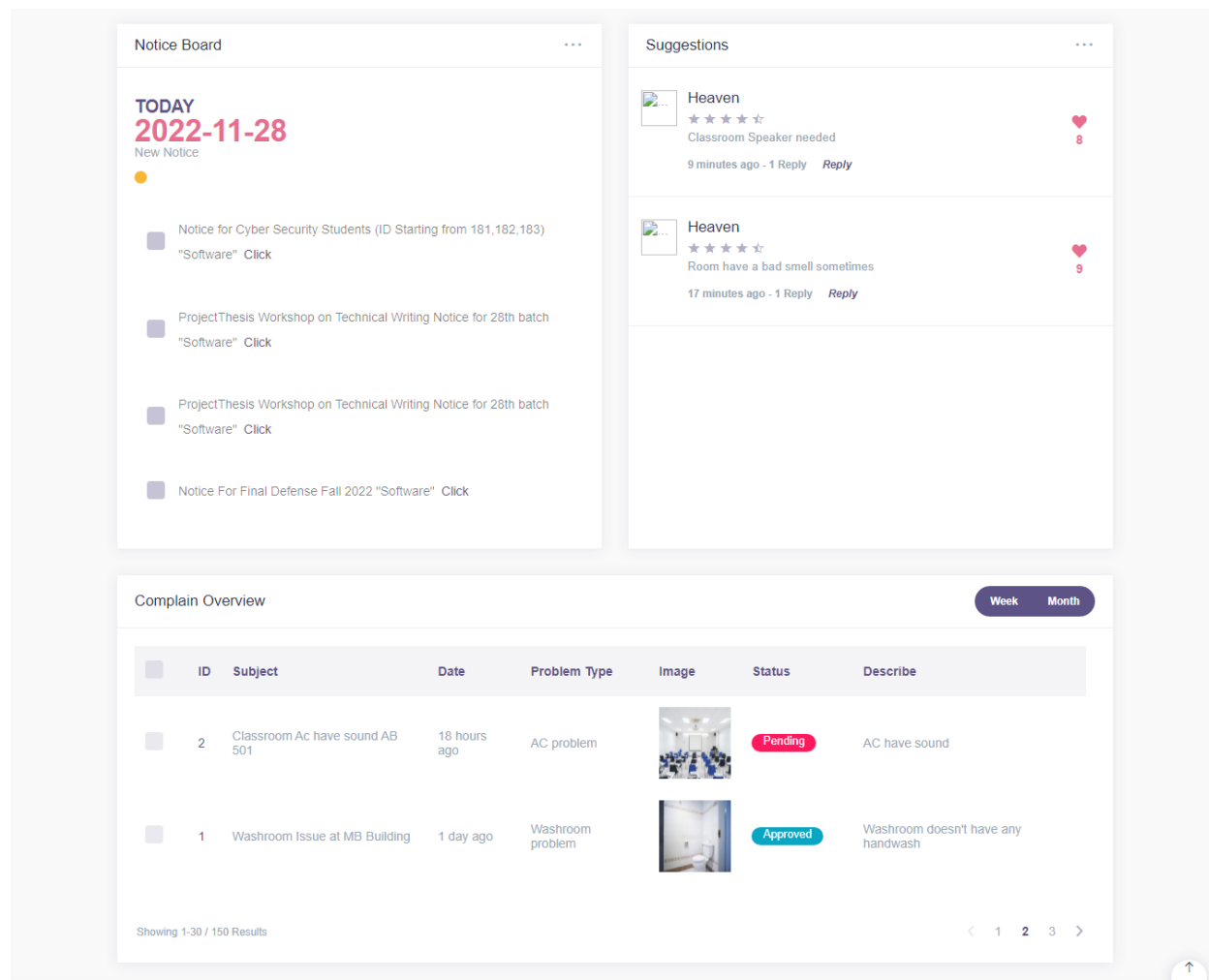
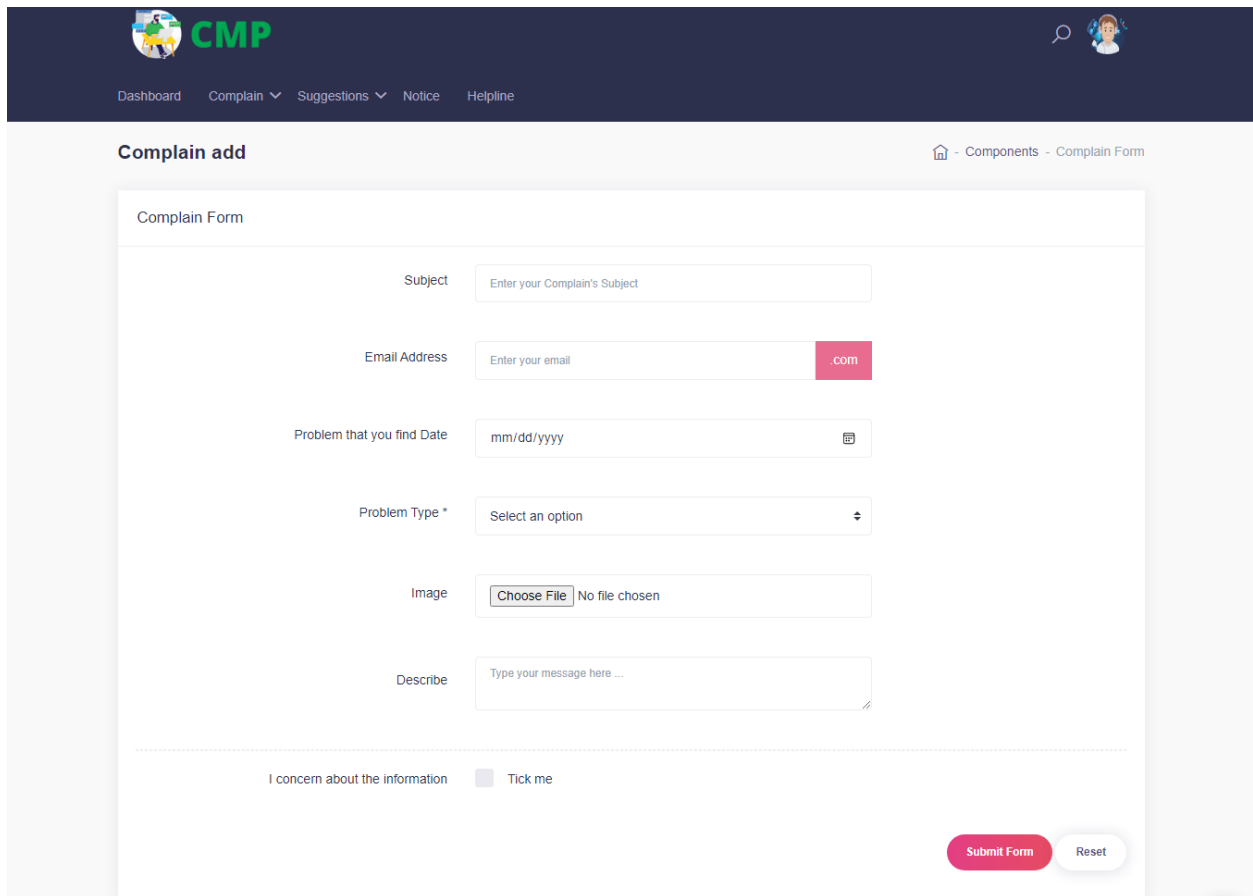


Fig 6.1.3.2 User Dashboard 2


6.1.4 User Complain





The screenshot shows a web application interface for a complaint form. At the top is a dark blue header with the 'CMP' logo on the left and a search icon and user profile icon on the right. Below the header is a navigation bar with links: 'Dashboard', 'Complain' (with a dropdown arrow), 'Suggestions' (with a dropdown arrow), 'Notice', and 'Helpline'. The main content area is titled 'Complain add' and includes a breadcrumb trail: 'Home - Components - Complain Form'. The form itself is titled 'Complain Form' and contains several input fields: 'Subject' with a placeholder 'Enter your Complain's Subject'; 'Email Address' with a placeholder 'Enter your email' and a red '.com' domain dropdown; 'Problem that you find Date' with a placeholder 'mm/dd/yyyy' and a calendar icon; 'Problem Type *' with a dropdown menu showing 'Select an option'; 'Image' with a 'Choose File' button and 'No file chosen' text; and 'Describe' with a text area placeholder 'Type your message here ...'. At the bottom of the form is a checkbox labeled 'I concern about the information' with the text 'Tick me' next to it. To the right of the form are two buttons: a red 'Submit Form' button and a white 'Reset' button.

Fig 6.1.4 User Complain


6.1.5 User Complain View

**CMP**

[Dashboard](#) [Complain](#) [Suggestions](#) [Notice](#) [Helpline](#)









Complains

 - Datatables

Complain List

Copy Excel Csv Pdf Print

Search:

ID	Subject	Date	Problem Type	Image	Status	Describe	Actions
1	Washroom Issue at MB Building	1 day ago	Washroom problem		Approved	Washroom doesn't have any handwash	 
2	Classroom Ac have sound AB 501	18 hours ago	AC problem		Pending	AC have sound	 


Showing 1 to 2 of 2 entries

[Previous](#) **1** [Next](#)



Design By Heaven

Fig 6.1.4 User Complain View


6.1.5 User Suggestion

**CMP**

[Dashboard](#) [Complain](#) [Suggestions](#) [Notice](#) [Helpline](#)



Suggestion add

 - Components - Suggestion Form

Suggestion Form

Topic

Email Address

Describe


☐ I concern about the information ☐ I am sure

Submit Form



Reset

Design By Heaven

Fig 6.1.5 User Suggestions


CMP

[Dashboard](#)
[Complain](#)
[Suggestions](#)
[Notice](#)
[Helpline](#)








Suggestions Home - Datatables

Suggestion List

Copy Excel Csv Pdf Print

Search:


ID ↑↓	Topic ↑↓	Email ↑↓	Describe ↑↓	Submission Date ↑↓	Status ↑↓	Actions ↑↓
2	Air fresher need in each room	heaven@gmail.com	Room have a bad smell sometimes	20 minutes ago	Approved	 
3	Classroom Speaker needed	heaven@gmail.com	Classroom Speaker needed	13 minutes ago	Pending	 

Showing 1 to 2 of 2 entries



Previous 1 Next

6.1.6 User Suggestion View

6.1.7 User Notice Board View


CMP

[Dashboard](#)
[Complain](#)
[Suggestions](#)
[Notice](#)
[Helpline](#)









Notice board Home - Notice board

Notice board List

Copy Excel Csv Pdf Print

Search:

ID ↑↓	Topic ↑↓	Department ↑↓	Type ↑↓	Upload Date ↑↓	Action ↑↓
1	Midterm Overlap Notice Mail Fall 2022	Software	Examination	18 minutes ago	
2	Notice For Final Defense Fall 2022	Software	Academic	17 minutes ago	
3	ProjectThesis Workshop on Technical Writing Notice for 28th batch	Software	Admission	16 minutes ago	
4	ProjectThesis Workshop on Technical Writing Notice for 28th batch	Software	Class Schedule	16 minutes ago	
5	Notice for Cyber Security Students (ID Starting from 181,182,183)	Software	Admission	16 minutes ago	

Showing 1 to 5 of 5 entries

Previous 1 Next

Fig 6.1.7 User Notice Board View

6.1.8 User Helpline

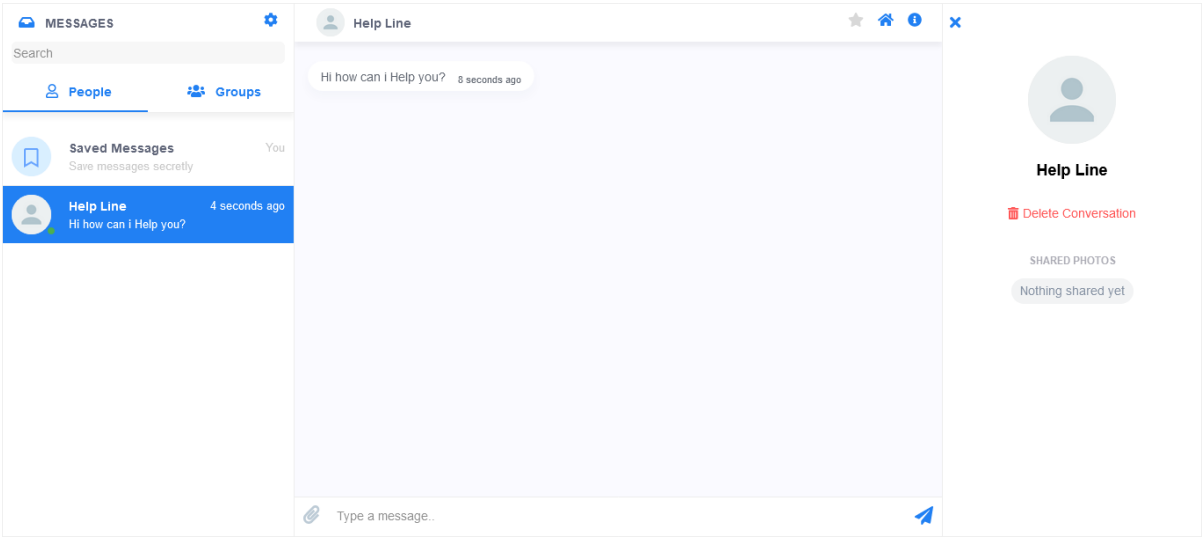


Fig 6.1.8 User Helpline

6.1.9 User Profile

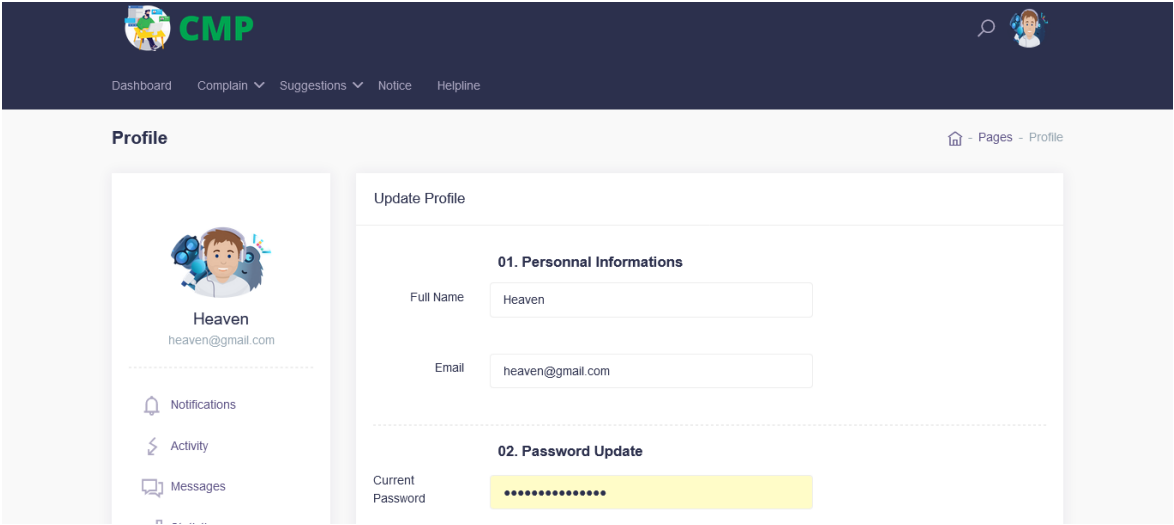


Fig 6.1.9 User Profile

6.1.10 User Notification

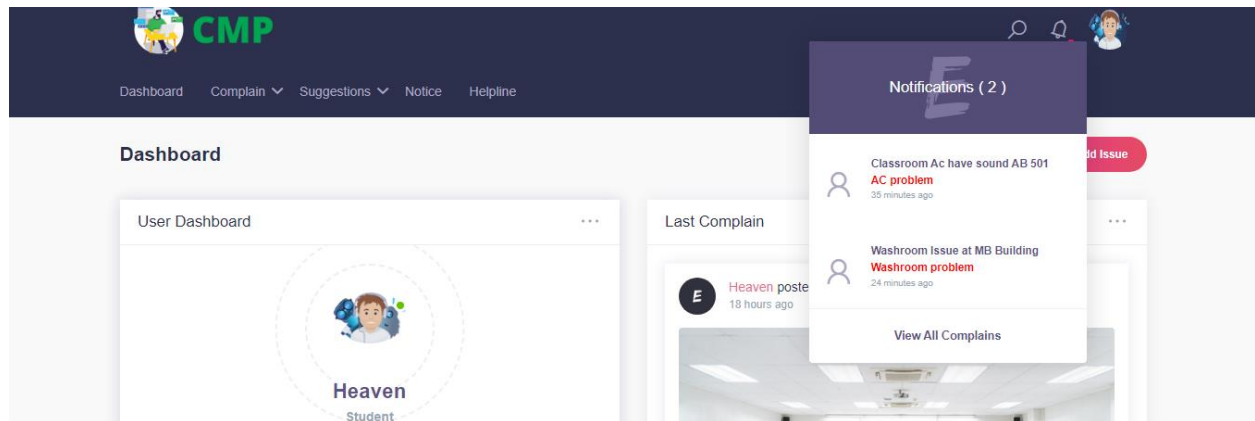


Fig 6.1.10 User Notification

For Admin

6.2.1 Admin Login

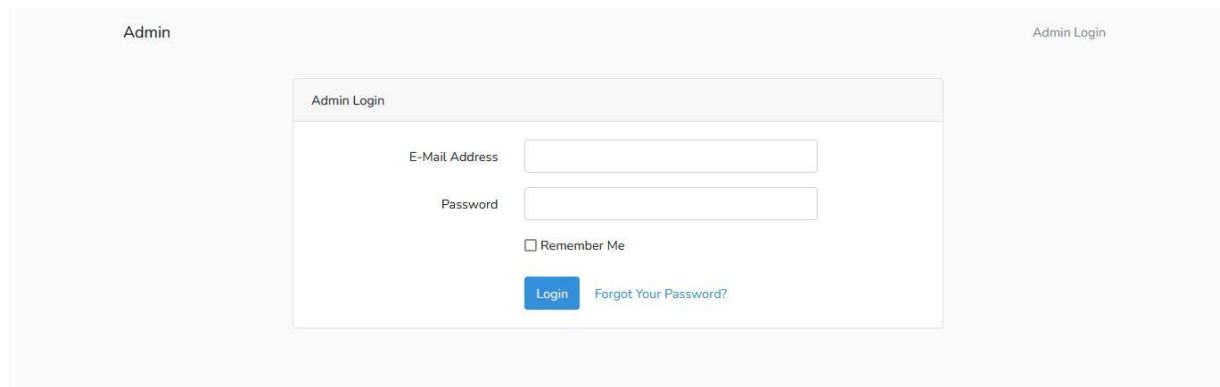


Fig 6.2.1 Admin Login

6.2.2 Admin Dashboard

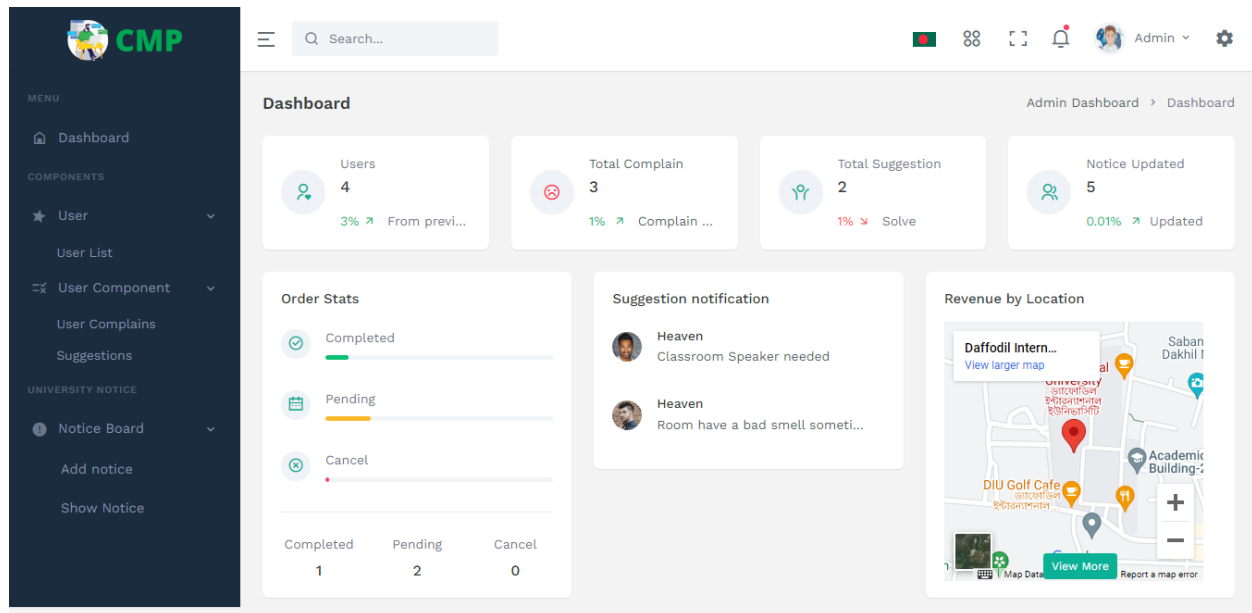


Fig 6.2.2 Admin Dashboard

6.2.3 Admin User List

The Admin User List page displays a table of user entries. The table includes columns for ID, Name, Email, User from, and Actions. The data is as follows:

ID	Name	Email	User from	Actions
1	User	testuser@gmail.com	44 minutes ago	
2	Help Line	helpline@gmail.com	13 minutes ago	
3	Heaven	heaven@gmail.com	11 minutes ago	
4	User	testuser2@gmail.com	27 minutes ago	

The page also includes a search bar, a "Showing 1 to 4 of 4 entries" indicator, and a "1" button for pagination. The footer shows the copyright information: "2022 © Beautiful Campus. Crafted with ❤ by Themesdesign".

Fig 6.2.3 Admin User List

6.2.4 Admin User Complain

User Complain

Tables > Data Tables

User Complain

Copy Excel PDF Column visibility

Search:

ID	Subject	Date	Problem Type	Image	Status
1	Washroom Issue at MB Building	25 minutes ago	Washroom problem		Approved
2	Classroom Ac have sound AB 501	36 minutes ago	AC problem		Pending
3	Projector problem at 501MB building	28 minutes ago	Projector problem		Pending

Showing 1 to 3 of 3 entries

2022 © Beautiful Campus. Crafted with ♥ by Themesdesign

Fig 6.2.4 Admin User Complain

6.2.5 Admin User Suggestions

User Suggestion

Tables > Data Tables

User Suggestion

Copy Excel PDF Column visibility

Search:

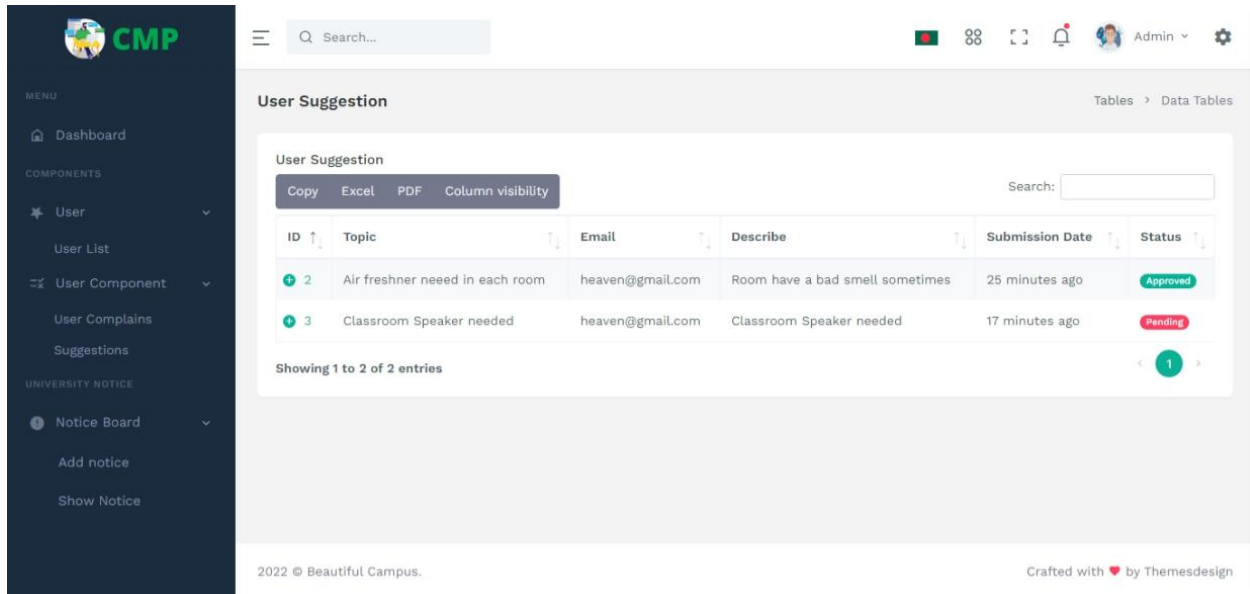
ID	Topic	Email	Describe	Submission Date	Status
2	Air freshner need in each room	heaven@gmail.com	Room have a bad smell sometimes	25 minutes ago	Approved
3	Classroom Speaker needed	heaven@gmail.com	Classroom Speaker needed	17 minutes ago	Pending

Showing 1 to 2 of 2 entries

2022 © Beautiful Campus. Crafted with ♥ by Themesdesign

Fig 6.2.5 Admin User Suggestions

6.2.6 Admin Notice Add



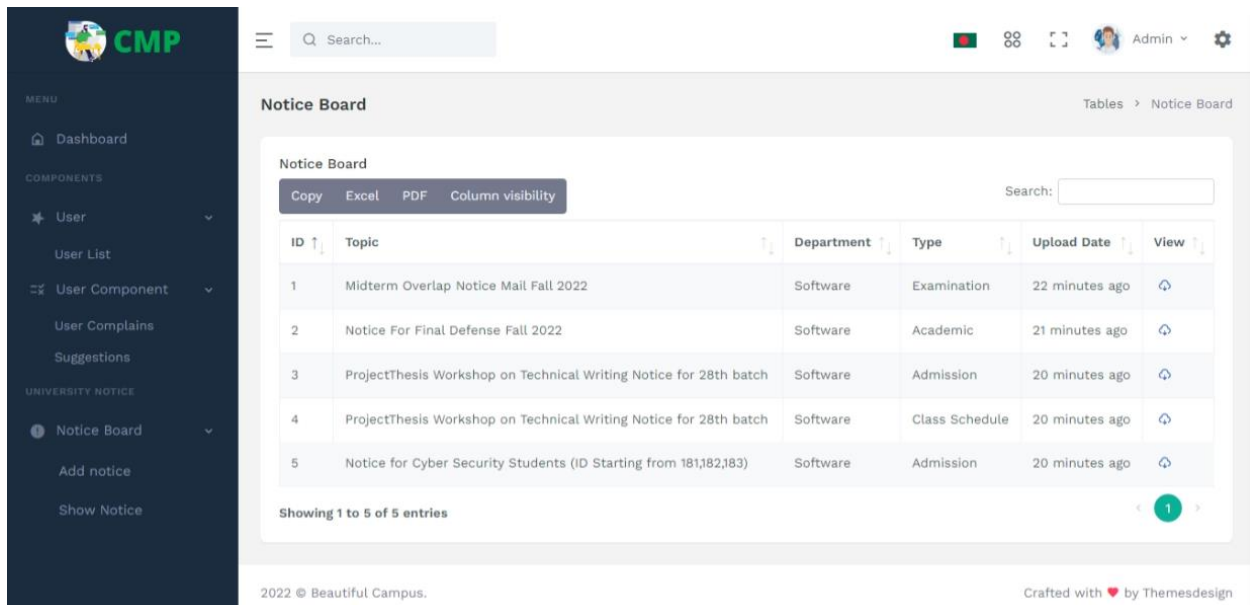
The screenshot shows the CMP Admin interface. The left sidebar contains a menu with 'User' selected. The main content area displays the 'User Suggestion' table. The table has columns: ID, Topic, Email, Describe, Submission Date, and Status. There are two entries: one approved and one pending.

ID	Topic	Email	Describe	Submission Date	Status
2	Air fresher need in each room	heaven@gmail.com	Room have a bad smell sometimes	25 minutes ago	Approved
3	Classroom Speaker needed	heaven@gmail.com	Classroom Speaker needed	17 minutes ago	Pending

Showing 1 to 2 of 2 entries

Fig 6.2.6 Admin Notice Add

6.2.7 Admin Notice Show



The screenshot shows the CMP Admin interface. The left sidebar contains a menu with 'Notice Board' selected. The main content area displays the 'Notice Board' table. The table has columns: ID, Topic, Department, Type, Upload Date, and View. There are five entries listed.

ID	Topic	Department	Type	Upload Date	View
1	Midterm Overlap Notice Mail Fall 2022	Software	Examination	22 minutes ago	View
2	Notice For Final Defense Fall 2022	Software	Academic	21 minutes ago	View
3	ProjectThesis Workshop on Technical Writing Notice for 28th batch	Software	Admission	20 minutes ago	View
4	ProjectThesis Workshop on Technical Writing Notice for 28th batch	Software	Class Schedule	20 minutes ago	View
5	Notice for Cyber Security Students (ID Starting from 181,182,183)	Software	Admission	20 minutes ago	View

Showing 1 to 5 of 5 entries

Fig 6.2.7 Admin Notice Show

6.2.8 Admin User Complain Approve

The screenshot shows the 'User Complain' section of the CMP Admin interface. The left sidebar contains a menu with 'Dashboard', 'User List', 'User Component', 'User Complain', and 'Suggestions'. The main content area displays a table of user complaints. The first complaint, ID 1, is 'Washroom Issue at MB Building' and is marked as 'Approved'. The second complaint, ID 2, is 'Classroom Ac have sound AB 501' and is marked as 'Pending'. The third complaint, ID 3, is 'Projector problem at 501MB building' and is marked as 'Pending'. The table has columns for ID, Subject, Date, Problem Type, Image, and Status. Below the table, there is a 'Describe' section for the first complaint, showing the description 'Washroom doesn't have any handwash' and an 'Actions' section with icons for edit, approve, and delete. The footer of the interface shows '2022 © Beautiful Campus.' and 'Crafted with ❤ by Themesdesign'.

ID	Subject	Date	Problem Type	Image	Status
1	Washroom Issue at MB Building	27 minutes ago	Washroom problem		Approved
2	Classroom Ac have sound AB 501	37 minutes ago	AC problem		Pending
3	Projector problem at 501MB building	29 minutes ago	Projector problem		Pending

Fig 6.2.8 Admin User Complain Approve

6.2.9 Admin Profile Update

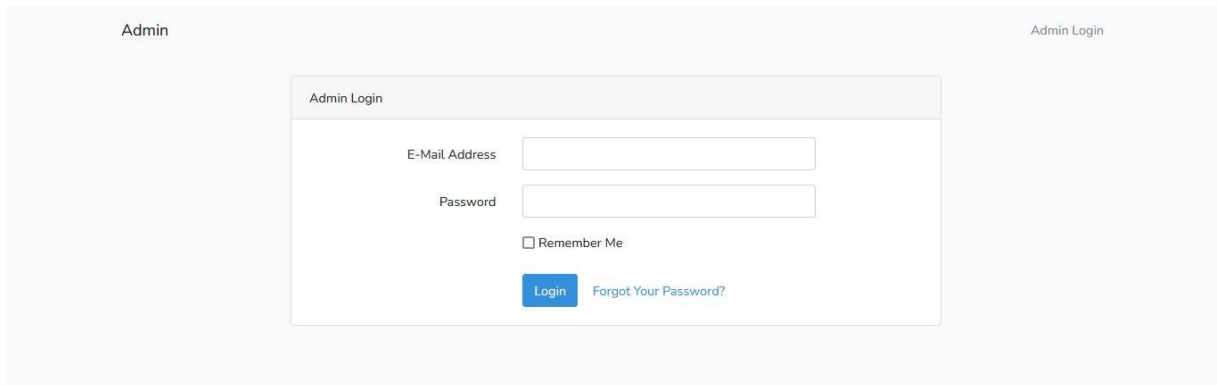
This screenshot is identical to the one above, showing the 'User Complain' section of the CMP Admin interface. It displays a table of user complaints with three entries: ID 1 (Approved), ID 2 (Pending), and ID 3 (Pending). The interface includes a sidebar menu, a search bar, and a footer with copyright information.

ID	Subject	Date	Problem Type	Image	Status
1	Washroom Issue at MB Building	27 minutes ago	Washroom problem		Approved
2	Classroom Ac have sound AB 501	37 minutes ago	AC problem		Pending
3	Projector problem at 501MB building	29 minutes ago	Projector problem		Pending

Fig 6.2.9 Admin profile password update

For Authority

6.3.1 Authority Login



The image shows a web form titled "Admin Login" within a light gray container. The form itself has a white background and a light gray header. It contains two input fields: "E-Mail Address" and "Password". Below the password field is a checkbox labeled "Remember Me". At the bottom of the form are two buttons: a blue "Login" button and a blue link "Forgot Your Password?".

Fig 6.3.1 Authority Login

6.3.2 Authority Dashboard

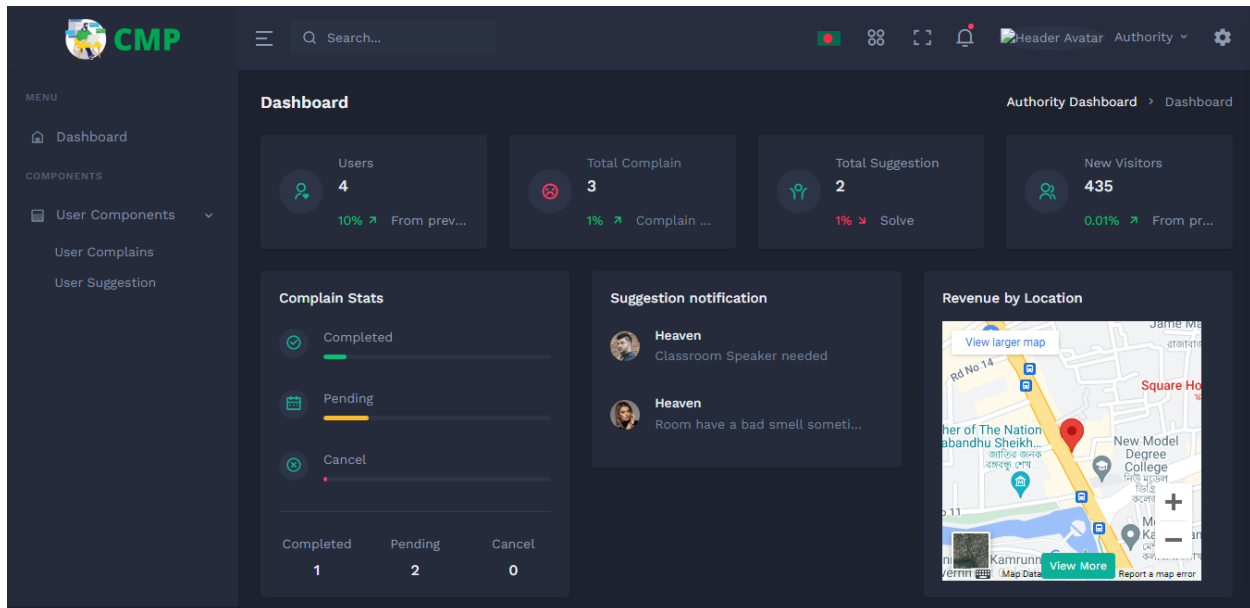


Fig 6.3.2 Authority Dashboard

6.3.3 Authority Dashboard 2

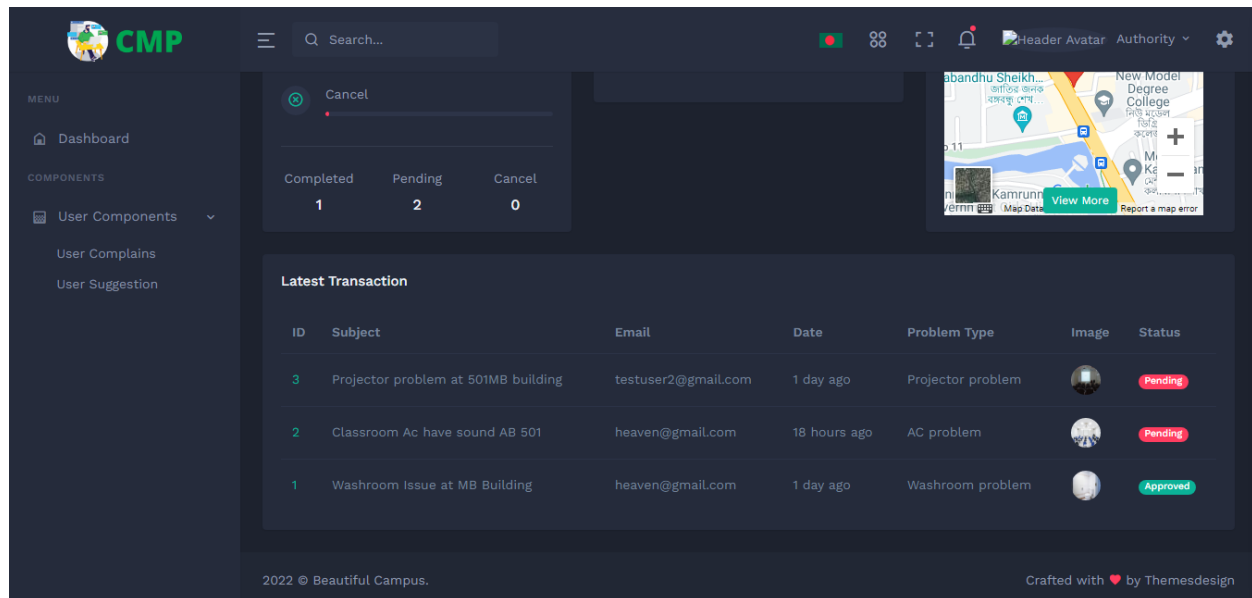


Fig 6.3.3 Authority Dashboard 2

6.3.4 Authority User Complain

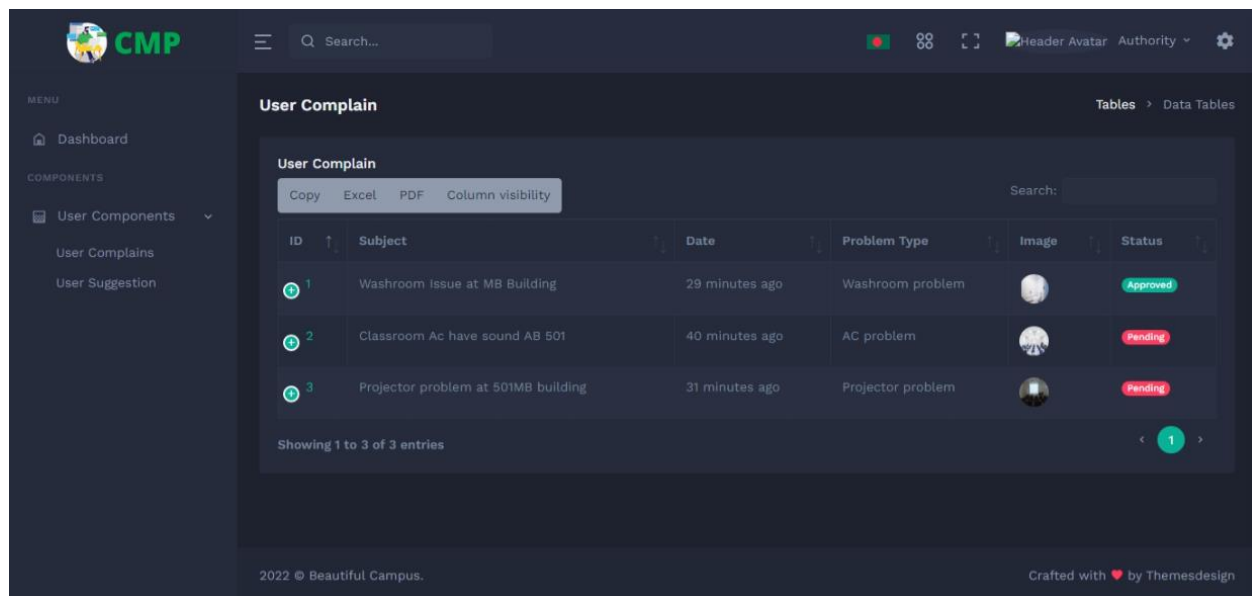


Fig 6.3.4 Authority User Complain

6.3.5 Authority User Suggestion

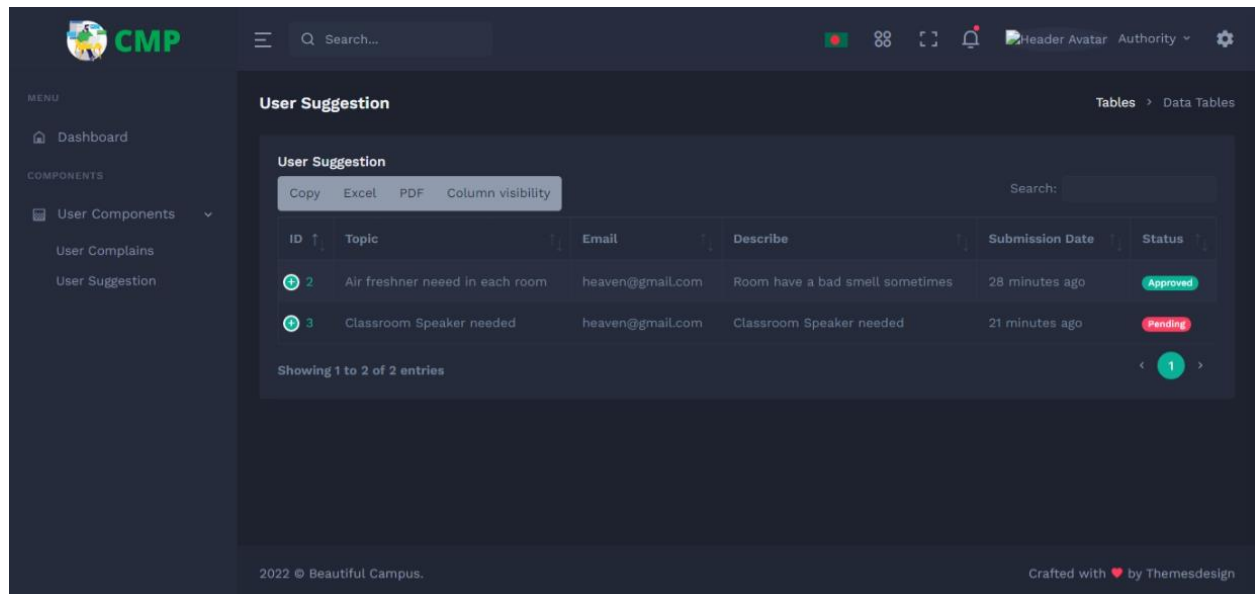


Fig 6.3.5 Authority User Suggestion

6.3.6 Authority User Complain Approve

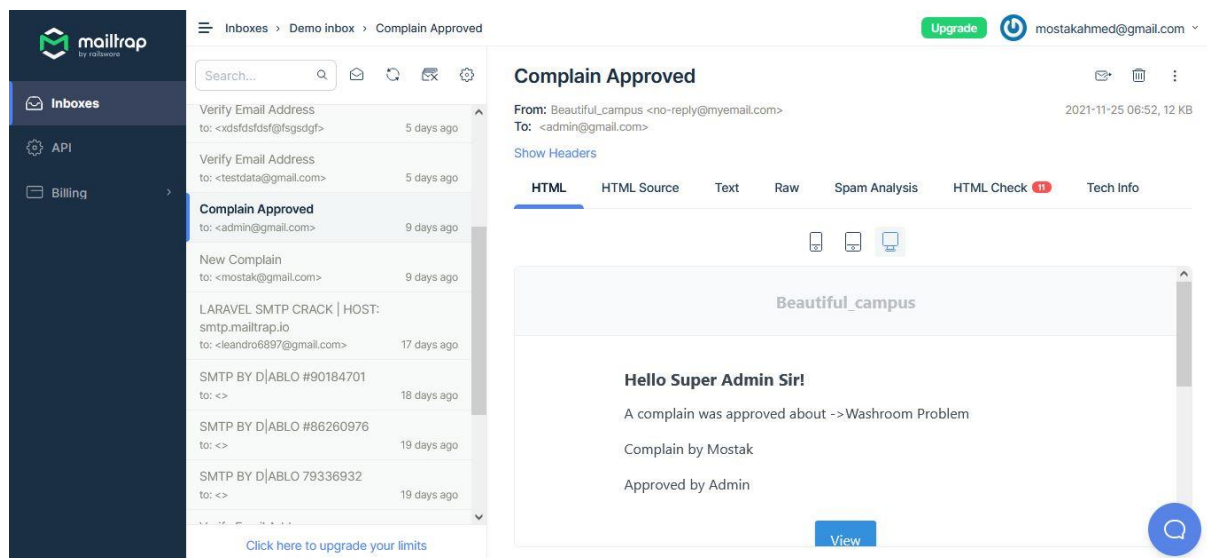


Fig 6.3.6 Authority User Complain Approve

6.3.6 Authority Profile update

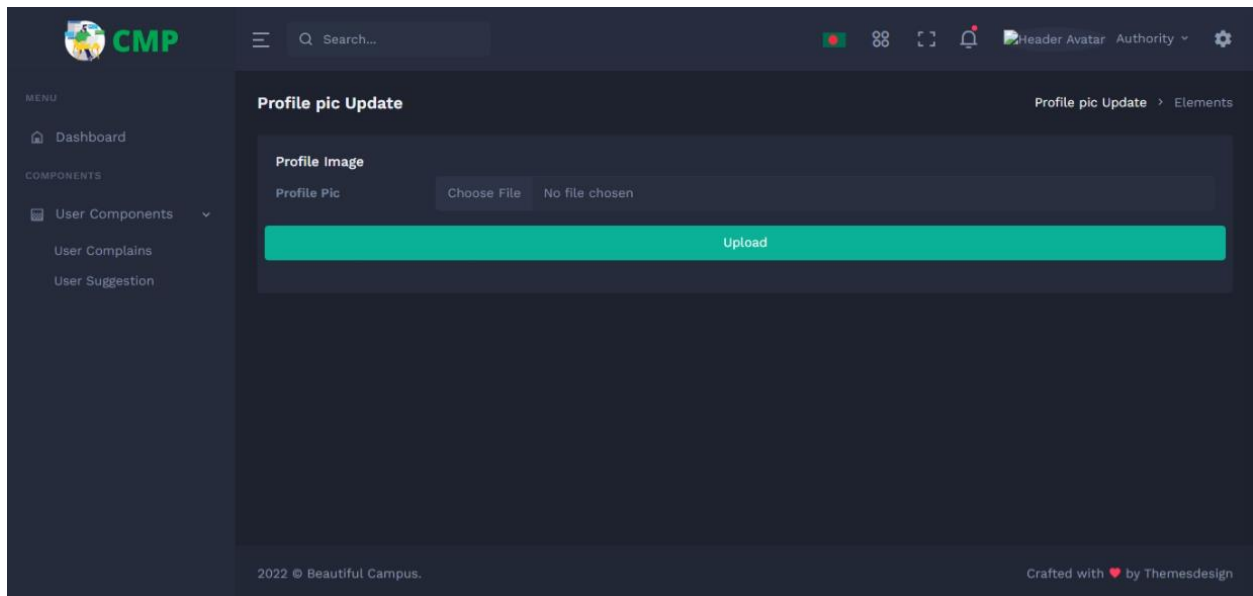


Fig 6.3.7 Authority Profile Update

CHAPTER 07

PROJECT SUMMARY

7.1 GitHub Link:

7.2 Limitations:

- The user has to have fast internet.
- The system does not accept accounts that have not been authorized, thus users must have active accounts to use it.

7.3 Obstacle & Achievement

Obstacle:

- Time and money constraints;
- learning new technologies and surroundings.

Achievement:

- Time and money constraints
- learning new technologies and surroundings

7.4 Future Work

In the future, we'll upload this project to a live server and work to address some significant problems so that students can use the website to access this service.

References

- [1] PNGTEM, Software Development Agile Model, 2019.
- [2] Lucid Chart, "Lucidchart," 2022.
- [3] ERDplus, "ERDplus," 2021.
- [4] XAMPP Apache + MariaDB + PHP, 2018.
- [5] NPM, NPM docs, 2019.
- [6] Php 8.1.1, Php.
- [7] Laravel.