



Daffodil
International
University

Beautiful Campus

Submitted by

Mostak Ahmed

171-35-1952

Department of Software Engineering

Daffodil International University

Supervised by

Ms. Nusrat Jahan

Assistant 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 “Beautiful Campus”, submitted by Mostak Ahmed (ID: 171-35-1952) 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.

APPROVAL

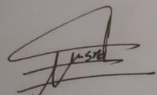
This thesis/project/internship titled on “Beautiful Campus”, submitted by Name: Md. Mostak Ahmed Nishat, ID: 171-35-1952 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



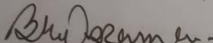
Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University



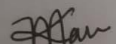
Internal Examiner 1

Nusrat Jahan
Assistant Professor
Department of Software Engineering
Daffodil International University



Internal Examiner 2

Khalid Been Badruzzaman Biplob
Senior Lecturer
Department of Software Engineering
Daffodil International University

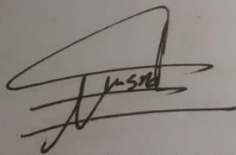


External Examiner

Professor Dr M Shamim Kaiser,
Professor
Institute of Information Technology
Jahangirnagar University

DECLARATION

I hereby declare that project titled “Beautiful Campus” has been completed by me under the supervision of Ms. Nusrat Jahan, Assistant 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.



Internal Examiner 1

Nusrat Jahan
Assistant Professor
Department of Software Engineering
Daffodil International University

ACKNOWLEDGEMENT

First and importantly, all praise and gratitude go to Almighty God for all of his blessing throughout my project work, which has given me patience and enabled me to complete my project on time. People, so I consider myself really fortunate to have obtained this as a result of the completion of this project.

All of them are the result of such monitoring and help, and I would want to express my gratitude to them.

I also express my heartfelt gratitude to my supervisor, Ms. Nusrat Jahan, Assistant Professor, Department of Software Engineering, Daffodil International University, Dhaka, and take our whole obligation.

She had a lot of knowledge in the Software Dept. industry, and her skillful judgment, supervision, and a high level of expertise helped me a lot with my project.

She is always ready to me when I need to supervise.

I'd like to express my appreciation and gratitude to Dr. Imran Mahmud, Associate Professor and Head of the Department of Software Engineering, for his kind assessment of my project, which enabled me to finish it effectively.

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 classmates who engaged in this discussion while completing course work.

Finally, and most importantly, I must respect my parents' wholehearted support and patience.

ABSTRACT

The Purpose of the project “BEAUTIFUL CAMPUS” is to digitally collect the survey of the complain by the university student, the project is based on web application that provide student a major way to organize, monitor, manage, and handle complaints, as well as to give the university an effective tool to identify and target problem areas, track complaints handling performance, and enhance solution processes. Beautiful Campus is online complain management system that empowers student with instant access to various information and collect the reasonable complain that a student face in 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
2.5 Security Requirement	8
2.6 Usability and Human-Interaction 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	57
SYSTEM TESTING	57
5.1 Introduction.....	57

5.2 Test Case	57
Test Case No -1.....	57
Testing Case No-2.....	58
Testing Case No-3.....	58
Testing Case No-4	59
Test Case No -5.....	60
Test Case No -6.....	60
CHAPTER 06	62
USER MANUAL.....	62
6.1 User manual.....	62
FOR User	62
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 Conclusion.....	77
7.5 Future Work.....	77
References.....	78

List of Figures

1.3.1 Agile Model	11
1.3.2 Gantt 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	8
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

Beautiful Campus offers a web-based solution to student difficulties that saves time, minimize hassle and get solution quickly. The Beautiful Campus project's goal is to make it easier to organize, monitor, manage, and handle complaints, as well as to give the university an effective tool to identify and target problem areas, track complaints handling performance, and enhance business processes.

Beautiful Campus is an online complaint management system that is used to reply quickly to student complaints.

Beautiful Campus is a complaint handling system that gives an online alternative for students to solve concerns while saving time and staying connected to the varsity coordinator.

The goal of the complaints management system is to make it simpler to monitor, track, and address complaints in order to offer the varsity with an efficient tool for identifying problem areas, monitoring complaint handling performance, and making Campus more improved.

This technique allows students to save time while also eliminating corruption in the classroom. The suggested solution eliminates the requirement for the students to visit the coordinator's office to resolve a problem.

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 era of Digital Bangladesh, we have to solve every problem Digitally. In order to maintain a big varsity campus suitable for the students sometimes we have to face many problems. Through this Digital Campaign “Beautiful Campus”, we can easily represent our problem to the authorities of our varsity so that we can get the solution of that problem very easily.

1.2.2 Benefits & Beneficiaries

Benefits

- User can easily add suggestion through the system to Admin
- User can add complain any kind of problem to admin by system.
- User can get notify if their complain approve or not.
- User can get help from Help Line on LIVE Chat.
- User can get University notice on Dashboard notice section.

Beneficiaries

- Student (User)
- Admin
- Super Admin

1.2.3 Goals

- Provide Digital Solution
- Easy paperless System
- Send report anonymously.
- Make campus Beautiful.
- Make campus more comfort for Student.
- Get every notification from university.

1.3 Proposed System Model

A software model is a simplify representation of a software process. Each Model represents a process from a specific perspective.

1.3.1 Agile Model

Our proposed model is Agile model. The Agile Model word in software means the ability to respond to changes when needed technology and people in Agile Model. Engineering terms carried out by the functional teams. [1]

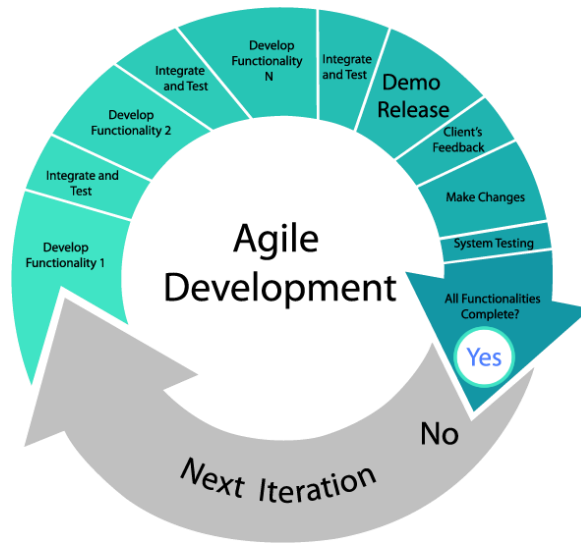


Fig 1.3.1 Agile Model

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
FC.U.RQ 1	Login	Users can log in using username and password	Functional	High
FC.U.RQ 2	Registration	Users can register himself/herself on the system by giving require information.	Functional	High
FC.U.RQ 3	Assign Complain	Users can submit any type of complain with description	Functional	High
FC.U.RQ 4	Assign Suggestions	Users can submit any type of Suggestions with description	Functional	Mid
FC.U.RQ 5	View Complain	User can view their assigned review.	Functional	Mid
FC.U.RQ 6	Update Complain	User can update their complain.	Functional	Mid
FC.U.RQ 7	View Suggestion	User can view their assigned Suggestion.	Functional	Mid
FC.U.RQ 8	Get Help	Users can help from Helpline by Live Chat.	Functional	Mid
FC.U.RQ 9	Get Notification	User can get every notice from university notice board.	Functional	Mid

Fig 2.1.1 Functional Requirement for user

For Admin:

RQ ID	RQ Name	Description	RQ Type	Priority
FC.A.RQ 1	Login	Admin can log in using email and password.	Functional	High
FC.A.RQ 2	See All Complain	Admin can see all User's Complain Submission.	Functional	High

FC.A.RQ 3	Solve the problem	Admin can solve the problem physically and assign it the system as a solve.	Functional	High
FC.A.RQ 4	Change Status	Admin can approve user complain/Suggestion request.	Functional	High
FC.A.RQ 5	Generate Report	Admin send a email if he/she approve any complain.	Functional	High
FC.A.RQ 6	Manage Account	Admin can update, delete the user account.	Functional	High
FC.A.RQ 7	View User list	Admin can check user list	Functional	High
FC.A.RQ 7	Give Solution	Admin can give instant help to user by live Chat.	Functional	High
FC.A.RQ 8	Give Notice	Admin can upload notice in system.	Functional	High
FC.A.RQ 9	View Notice	Admin can view Notice which is 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 using email and password	Functional	High
FC.A.RQ 2	View Complains	Authority can see all User's Complain Submission	Functional	High
FC.A.RQ 3	View Suggestions	Authority can see all User's Suggestion Submission	Functional	High
FC.A.RQ 4	User complain approve	Authority can approve user complain request	Functional	High
FC.A.RQ 5	View all user list	Authority can view all user list.	Functional	High
FC.A.RQ 6	Check Report	Authority can audit or check report that generate by admin.	Functional	High

2.2 Performance Requirement

2.2.1 Speed and Latency Requirement

- Data would be insert in MySQL database within few second.
- Database Query would be response on time and bring the result within few second.
- System UI should load within 3 second. However, it depends on machine and internet speed.

2.2.2 Precisions or Accuracy Requirements

- After successful login user should show the accurate result.
- All the complaint and application data should store accurate and valid information to the database.
- Specific complain and application would be view for specific user.
- Only register user can update their complain information.
- Only register users can make digital license and QR code by using their information.

2.3 Dependability Requirement

2.3.1 Reliability Requirements

- The user registration should register as a new user and update database with given information.
- Only accurate information can give access on the system.
- If user forget their password, he/she can change new password.
- Every new user must be verifying their email id in first login.
- Every user can update their complain information.
- This system should run on a web browser.
- This system should conduct operations immediately upon user request.
- This system design is responsive so it can comfortable for any device (Mobile, Tab, PC).

2.4 Maintainability and Supportability Requirement

2.4.1 Maintenance Requirement

- Modify the system when there are changes in the application environment.
- Fix bug when the system is corrupted.
- Fix accidental data mistakes by user.

2.4.2 Supportability Requirements

- Provide user manual.
- For user better experience, provide FAQ.

2.5 Security Requirement

2.5.1 Access Requirement

- Only registered and email verified user can login to the system.
- Only Admin can approve user complaint.
- Only specific email ID and password will accept for Authority/Admin login.
- Admin and Authority can check user complaints.
- Admin can send mail notification to authority.

2.6 Usability and Human-Interaction Requirement

2.6.1 Ease of Use Requirement

- System user interface should be more user friendly.
- New user should have knowledge about the system.
- Profile maintenance and user manual is not complex.

2.6.2 Understandability and Politeness Requirement

- Any kind of user should understand the system.
- Non-technical user can also operate the system.

CHAPTER 03

SYSTEM ANALYSIS

Introduction:

In general, systems development consists of two major components: data integration and computer architecture. The process of planning a new system or one to enhance or supplement an existing system known as software architecture. But, before we can start planning, we must first extensively study the existing system and determine how technology may best be used to improve productivity. The way to acquire and analyzing facts, diagnosing problems, and applying the information to propose system improvements is then referred to as network analysis. This is the duty of the systems analyst. It is the process of designing a good business system or upgrading an old one and outlining its components to meet specific needs. Before you begin preparing, you must properly review the existing system and identify what computer can be used very effectively.

3.1 Use Case

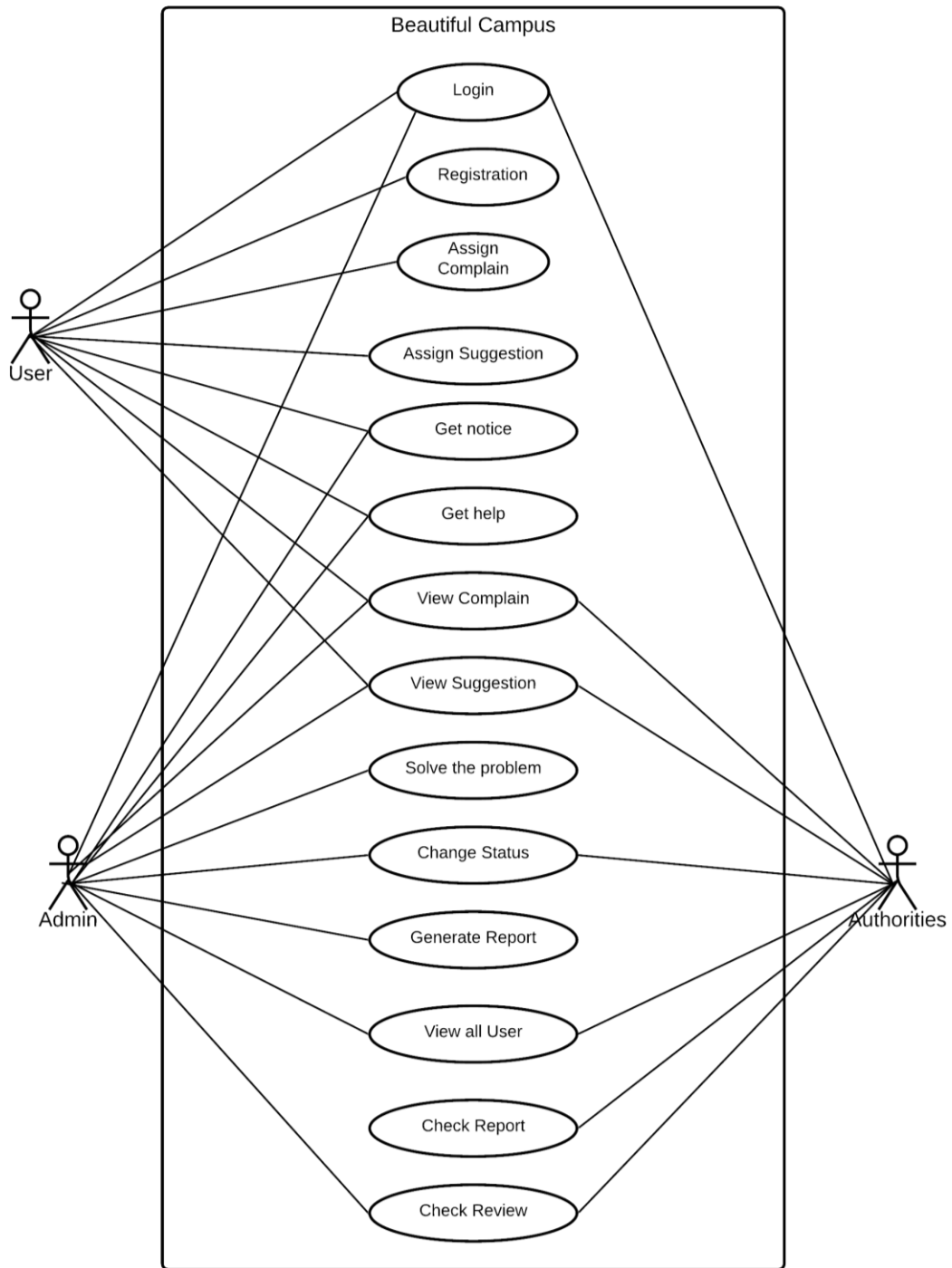


Fig 3.1.1: Use case [2]

3.1.1 Use Case for User

Use Case name	Login	
Scenario	User	
Brief Description	Users should login with their own valid email address.	
Actor	User (Student)	
Pre-condition	Must be use valid email and login type	
Post-Condition	Users must confirm email verification before login	
Flow of Condition	User	System
	<ul style="list-style-type: none"> • Users must confirm email verification before login 	<ul style="list-style-type: none"> • Confirm valid email address. • Confirm valid password • Email verification. • Give access to the system
Exception Condition	Give permission for access to the account.	

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 complain 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 complain 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

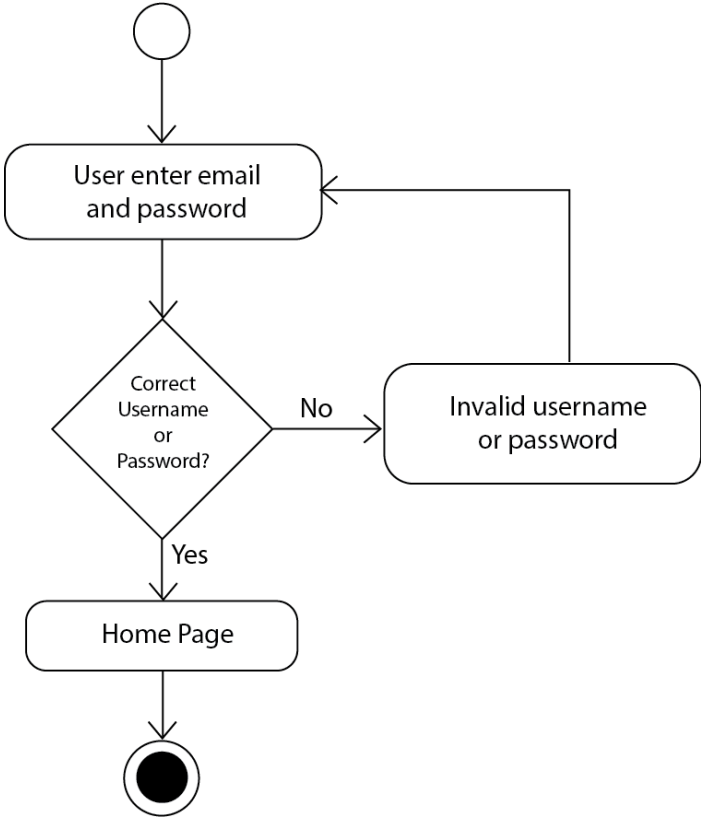


Fig 3.2.1.1 User Login

User should enter email and password after verifying user will enter home page.

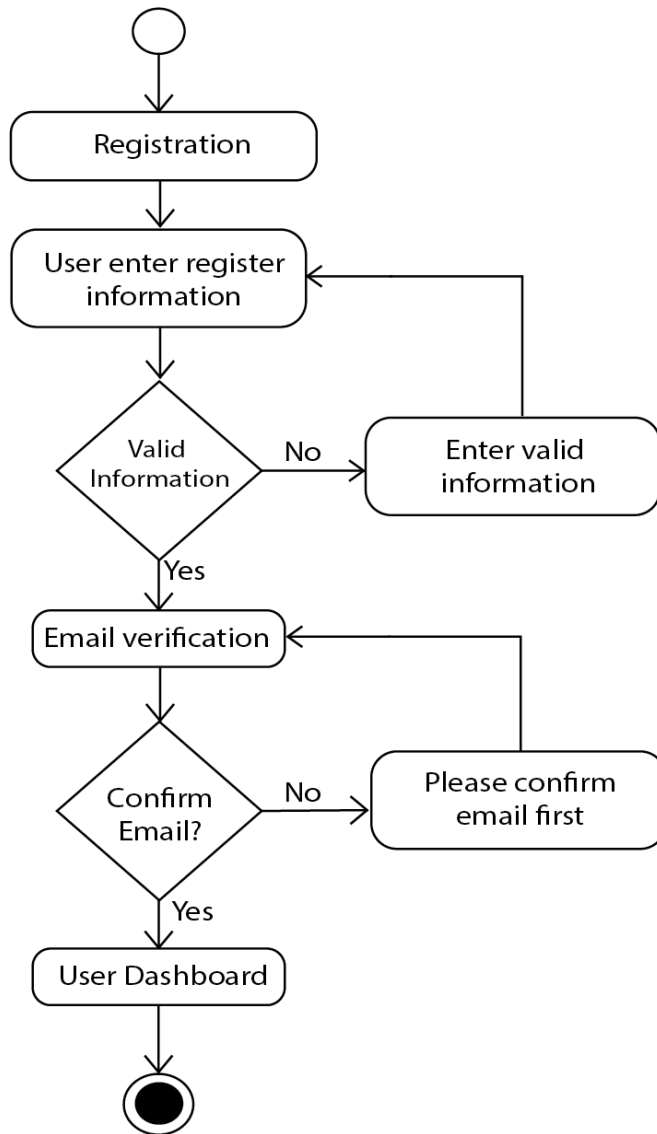


Fig 3.2.1.2 User Registration

Users should register themselves by their own email address. After verifying email verification then user will enter the user dashboard.

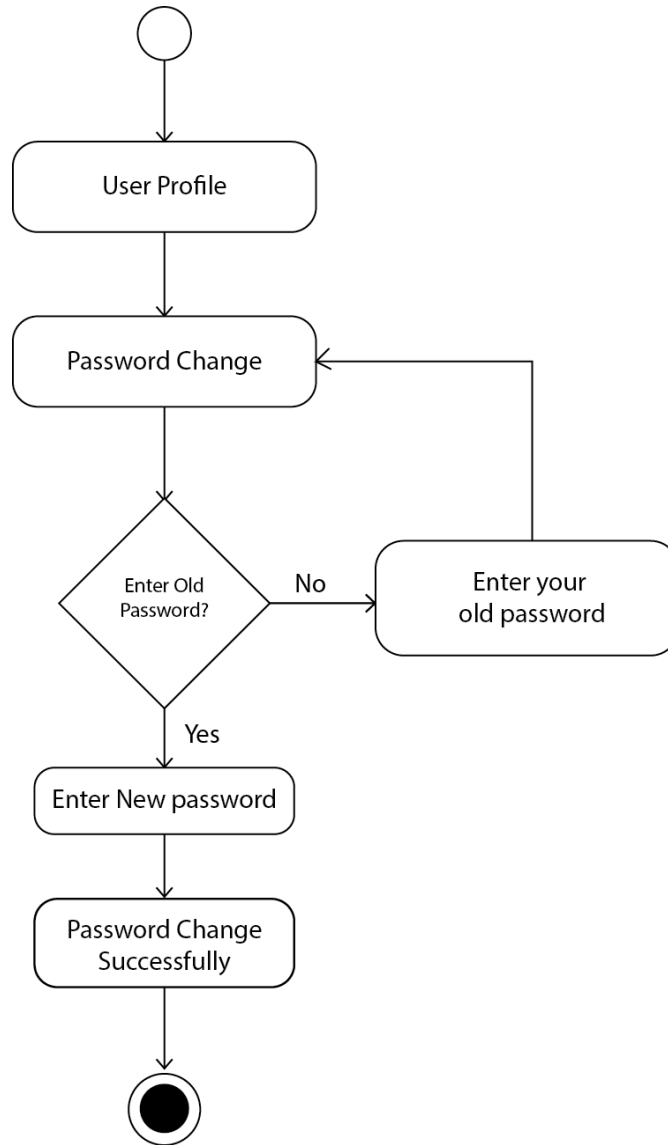


Fig 3.2.1.3 User Password Change

User can change their password by using old password.

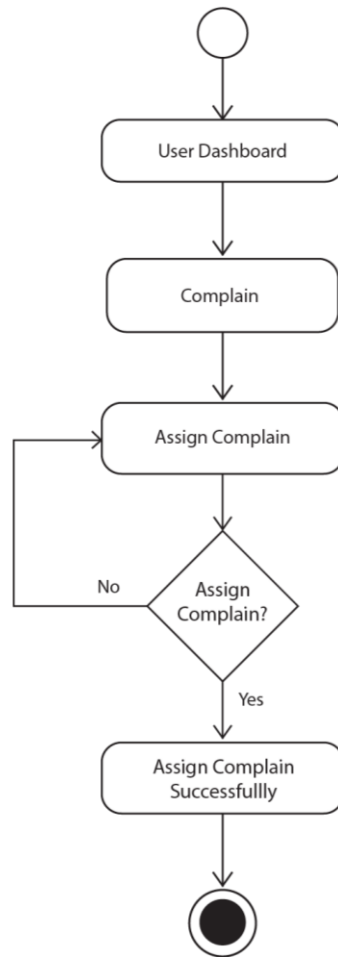


Fig 3.2.1.4 User Assign Complain

After entering dashboard User will assign a complain.

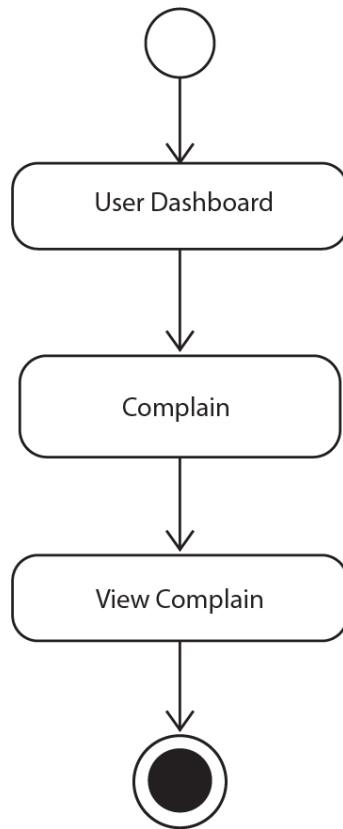


Fig 3.2.1.5 User View Complain

After assigning complain user can view his own complain

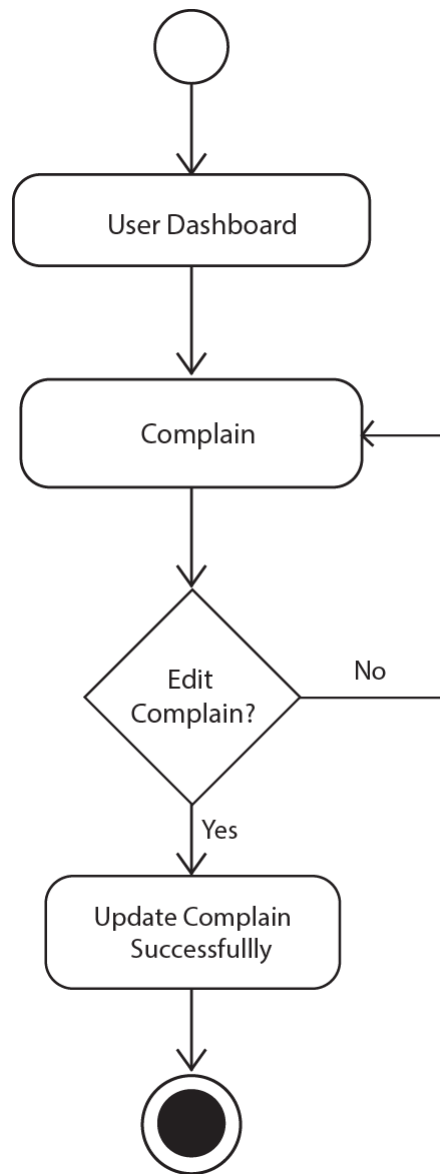


Fig 3.2.1.6 User Edit Complain

User Can edit complain after assigning complain in the system.

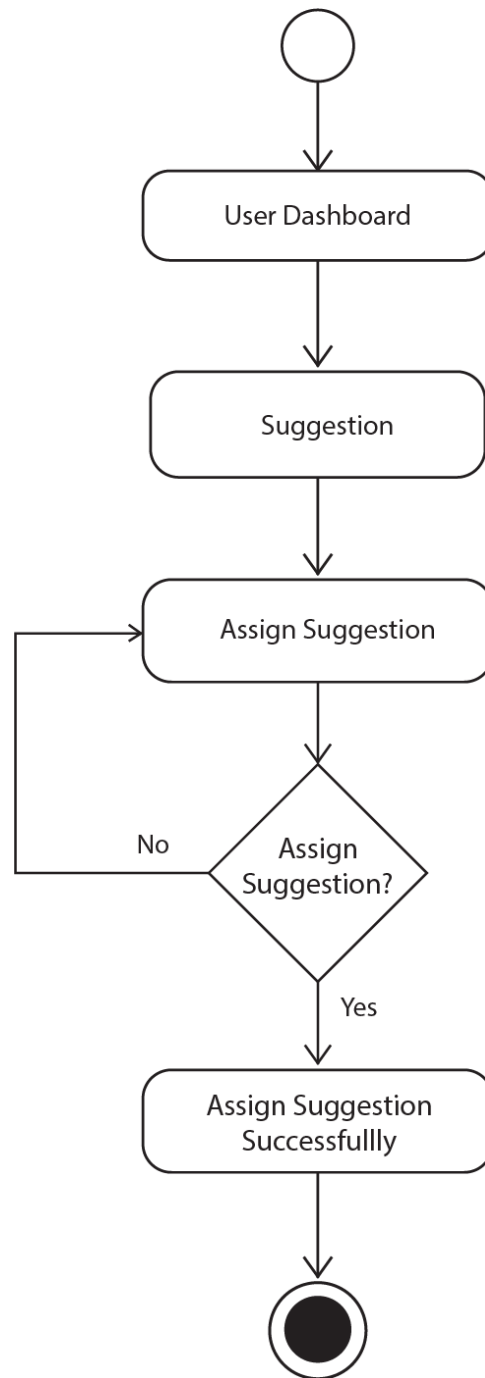


Fig 3.2.1.7 User Assign Suggestion

After entering dashboard User will assign a suggestion.

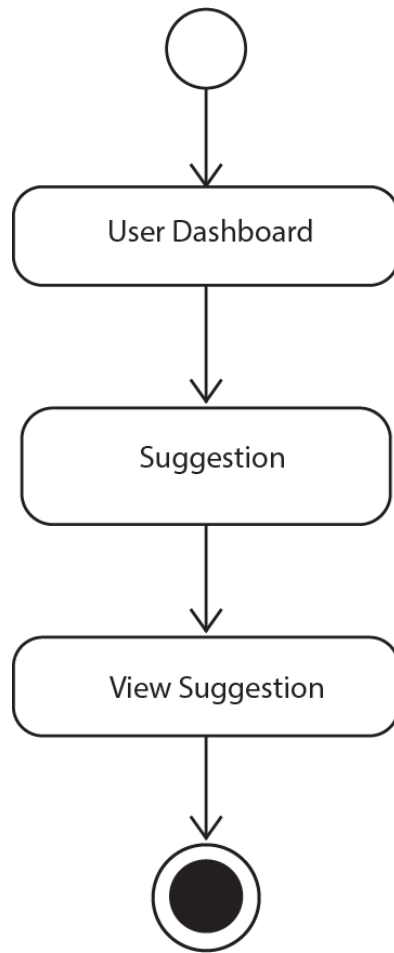


Fig 3.2.1.8 User View Suggestion

User Can view suggestion after assigning suggestion in the system.

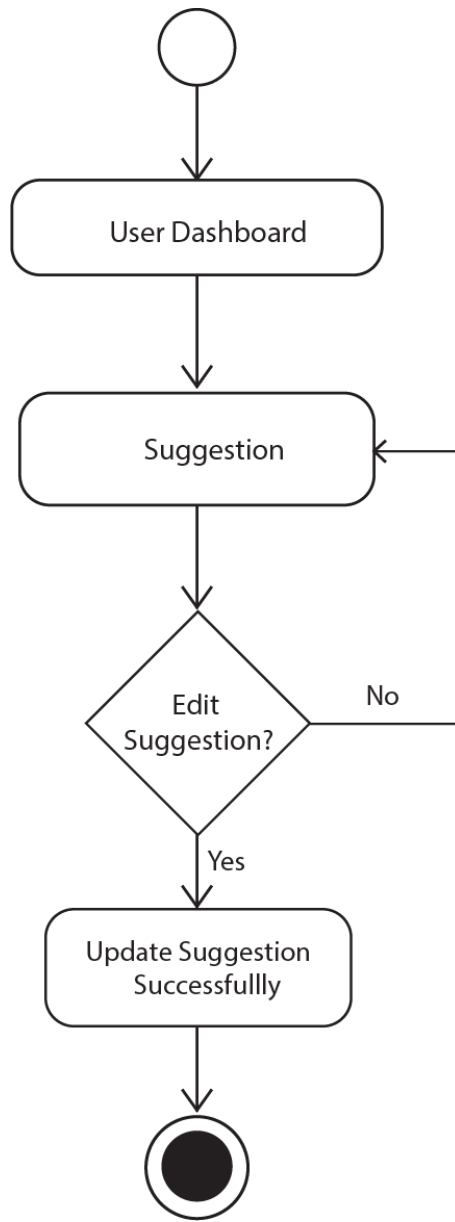


Fig 3.2.1.9 User Edit Suggestion

User Can edit suggestion after assigning a suggestion in the system

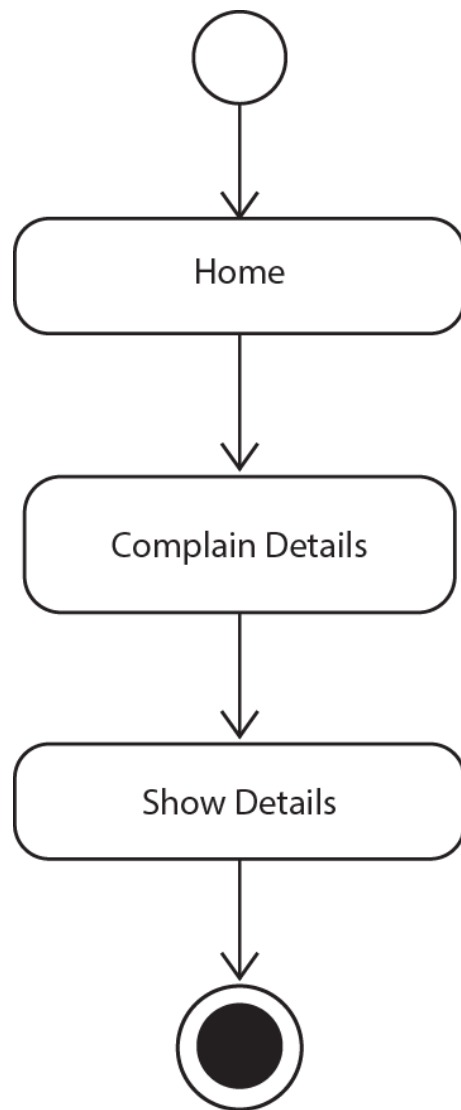


Fig 3.2.1.10 User Complain Details

User can view complain status and view complain details.

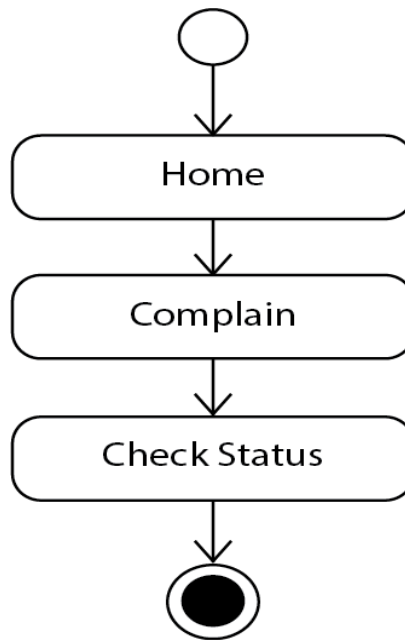


Fig 3.2.1.11 User Complain Status

User Can check complain status like the status is pending or approve.

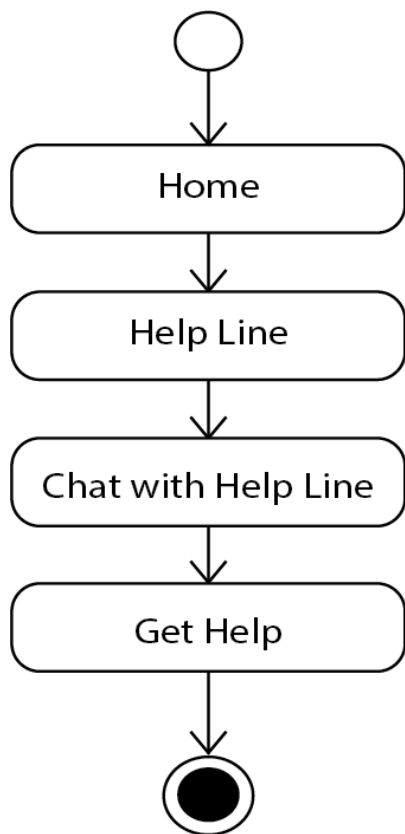


Fig 3.2.1.12 User Get Help

User can chat with help line by using this system.

3.2.2 Activity Diagram for Admin

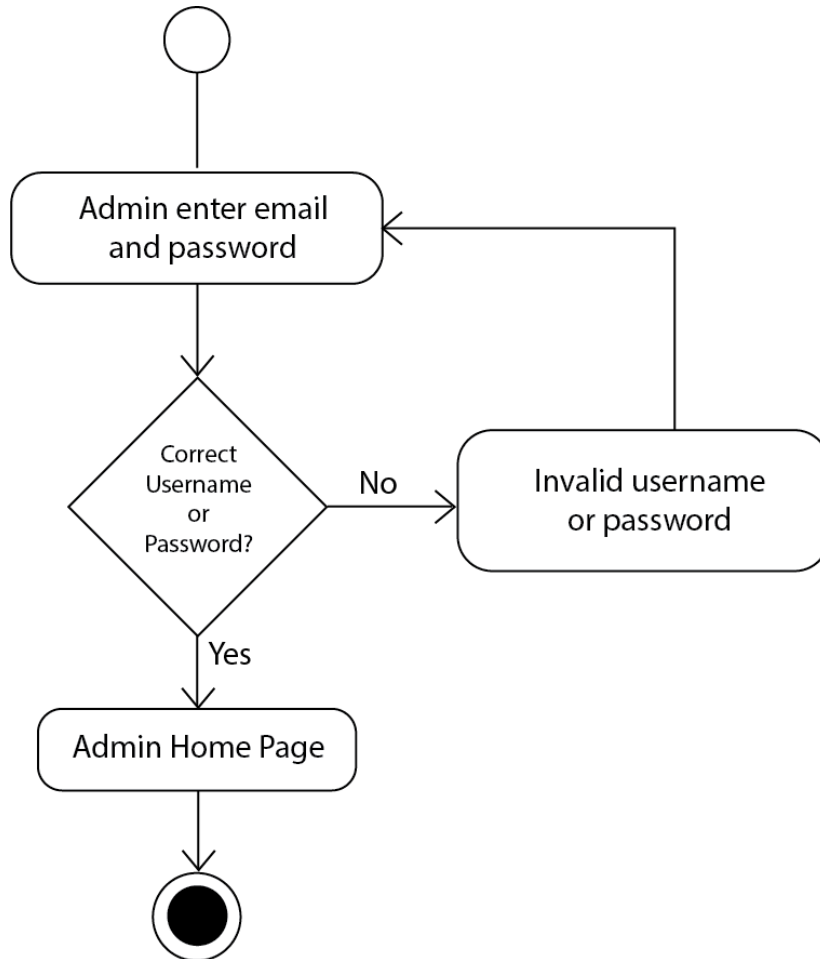


Fig 3.2.2.1 Admin Login

Admin enter email and password and then he can access to the system.

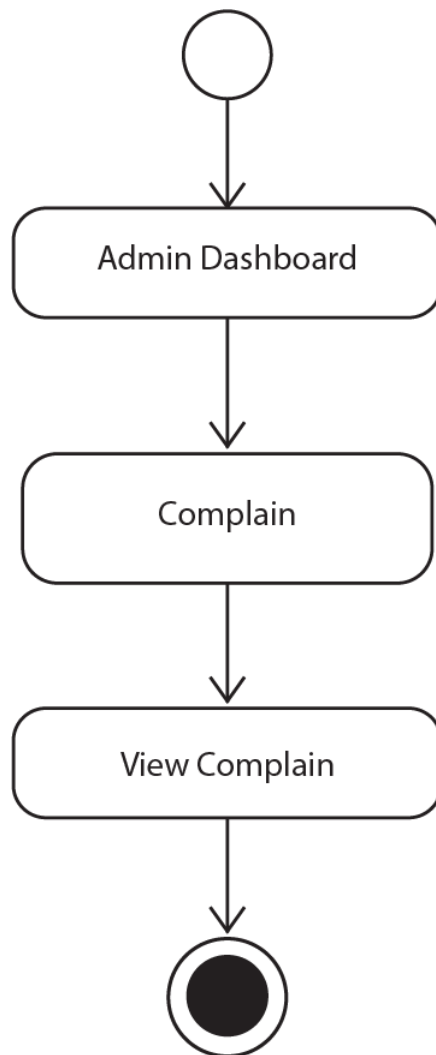


Fig 3.2.2.2 Admin Complain

Admin can check user complain

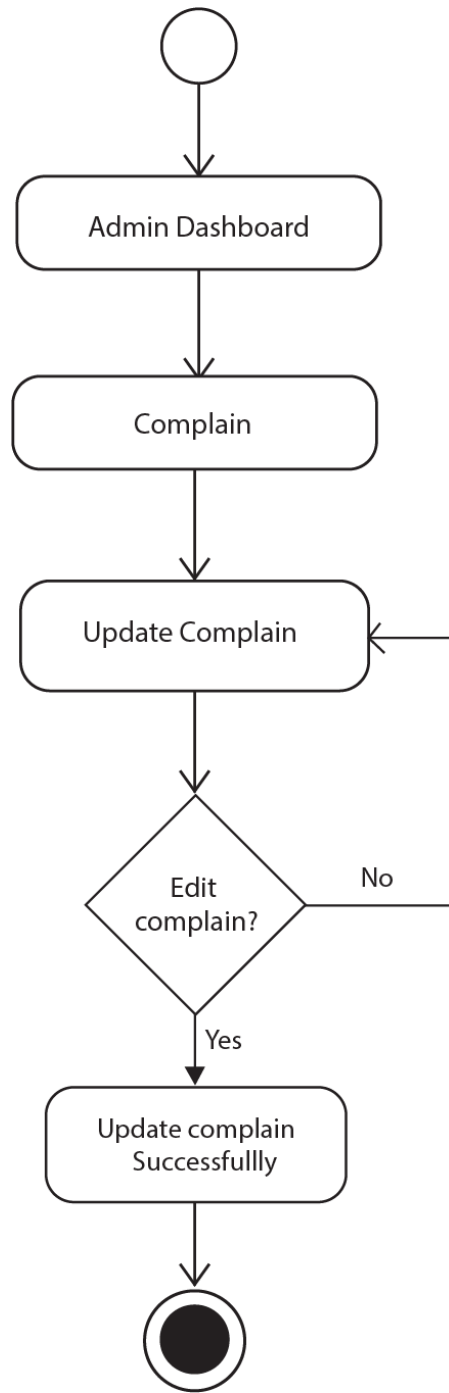


Fig 3.2.2.3 Admin Update Complain

Admin can update user complain if user input mistakenly.

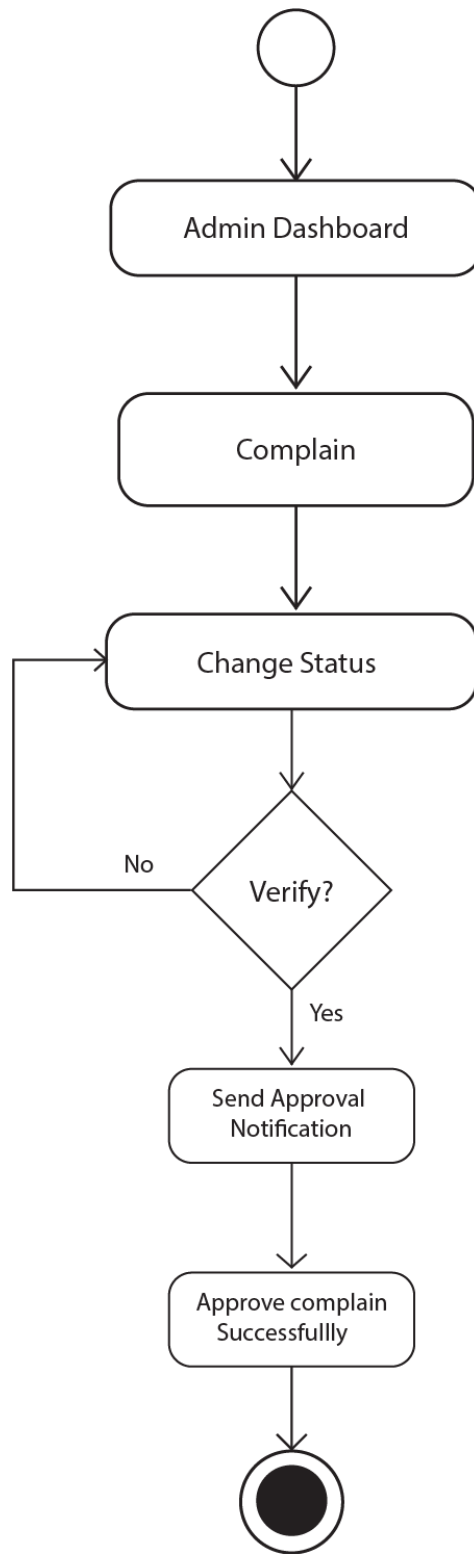


Fig 3.2.2.3 Admin Complain Approve

Admin can change the status the user complain and send a approval notification to super admin or authority.

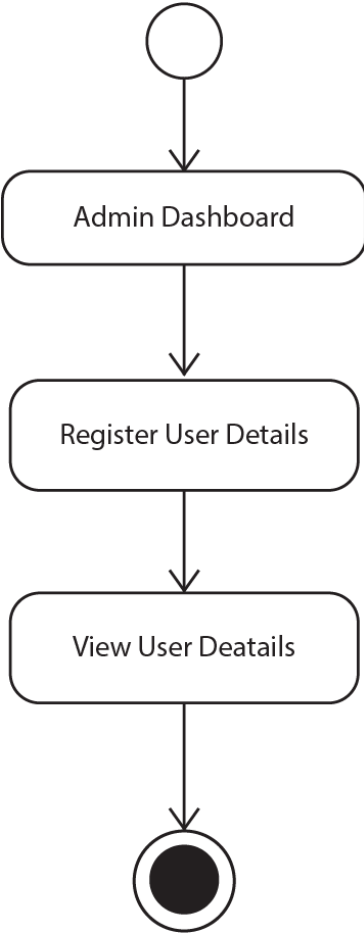


Fig 3.2.2.4 Admin User Details

Admin can check user details in the system.

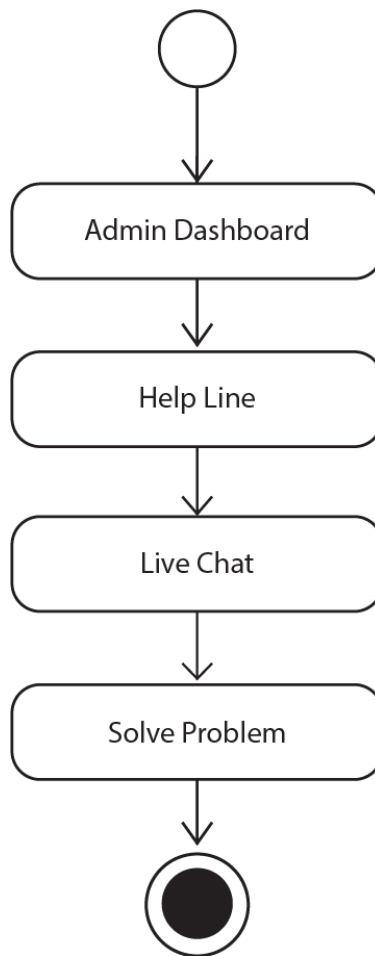


Fig 3.2.2.5 Admin User Help

Admin can help user by chat with them as a helpline.

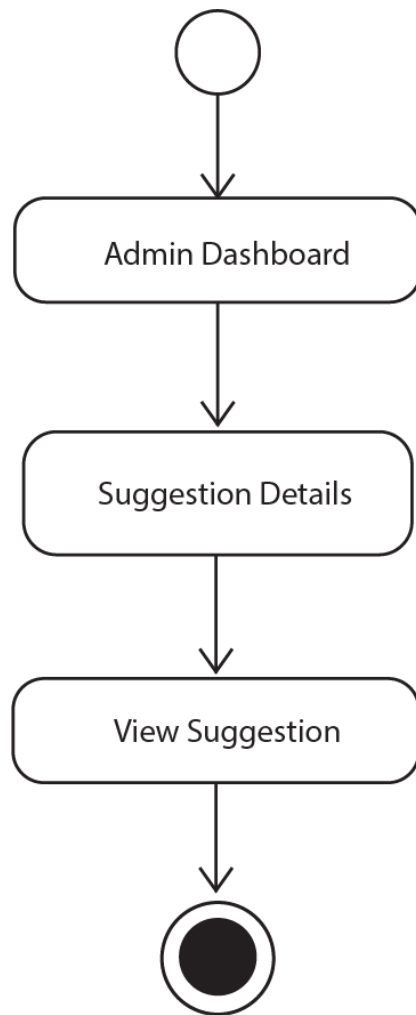


Fig 3.2.2.5 Admin User Suggestion Details

Admin can check user suggestion details in the system.

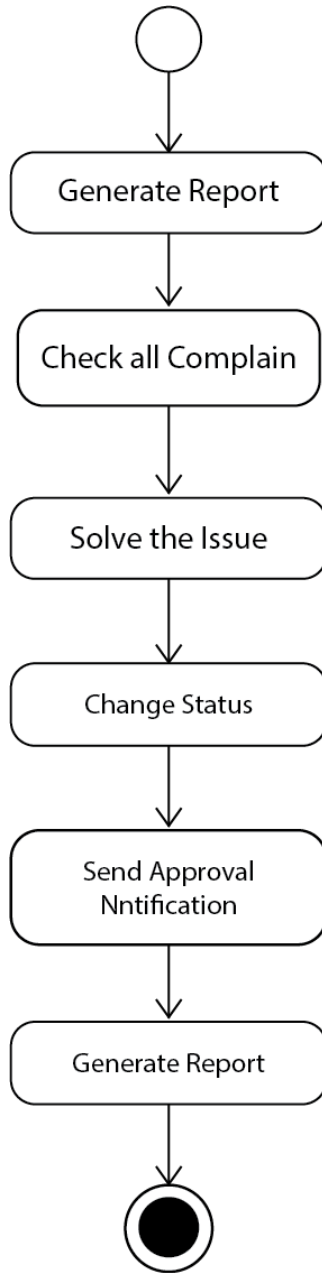


Fig 3.2.2.6 Admin Generate Report

Admin can generate report of user complain and suggestion and send it to super admin.

3.2.3 Activity Diagram for Authority

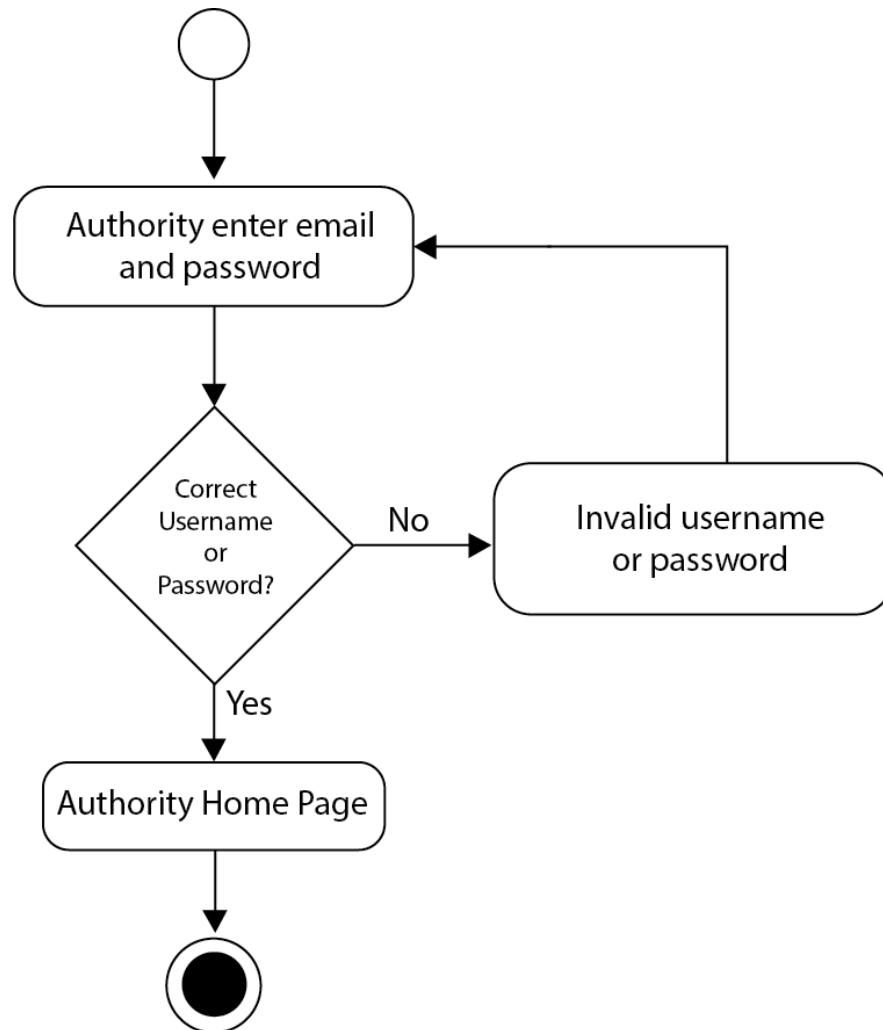


Fig 3.2.3.1 Authority Login

Authority can login to system by provide right email id and password.

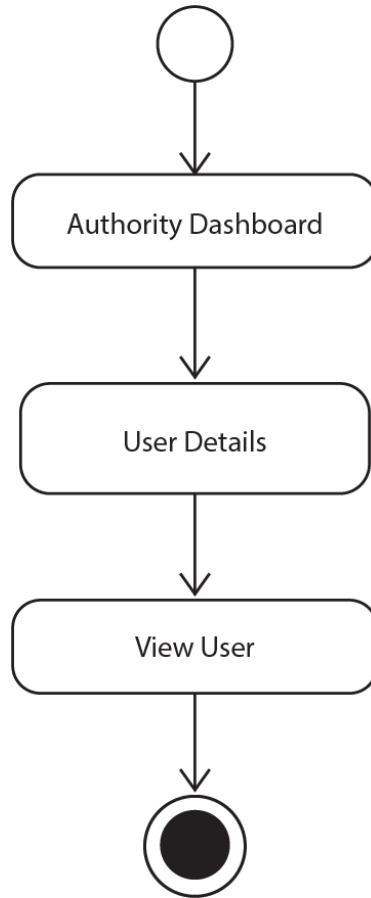


Fig 3.2.3.2 Authority View User

Authority can view user details like admin.

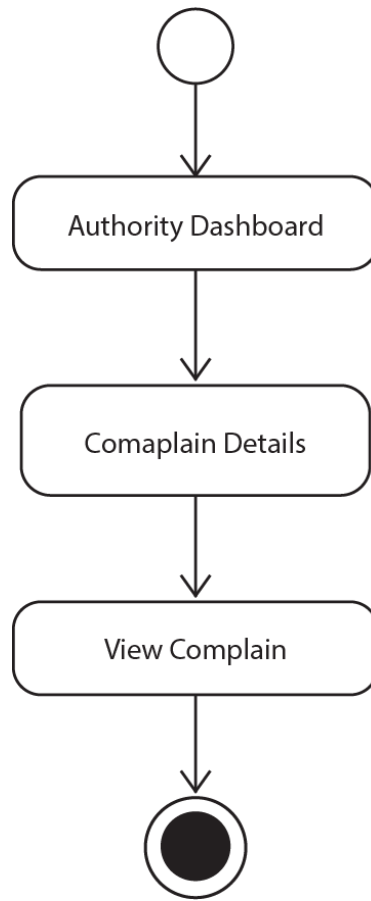


Fig 3.2.3.2 Authority User Complain

Authority can check user complain details in this system.

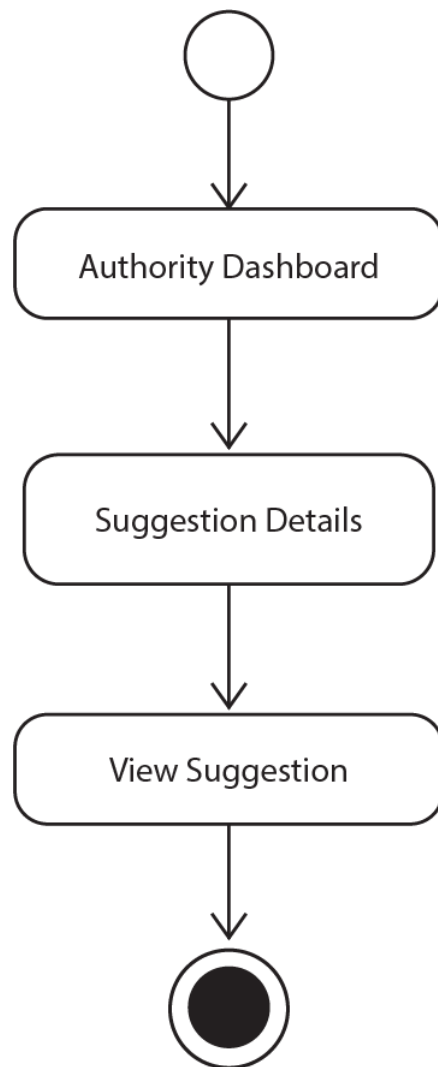


Fig 3.2.3.3 Authority User Suggestion

Authority can check user suggestion details in this system.

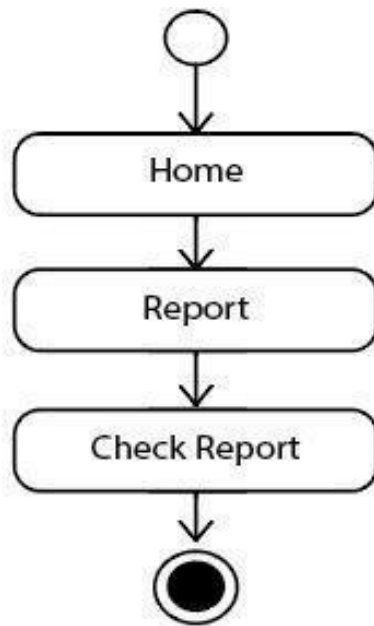


Fig 3.2.3.4 Authority Report check

Authority can check admin complain approval details in this system.

3.3 Sequence Diagram

3.3.1 Sequence Diagram for User

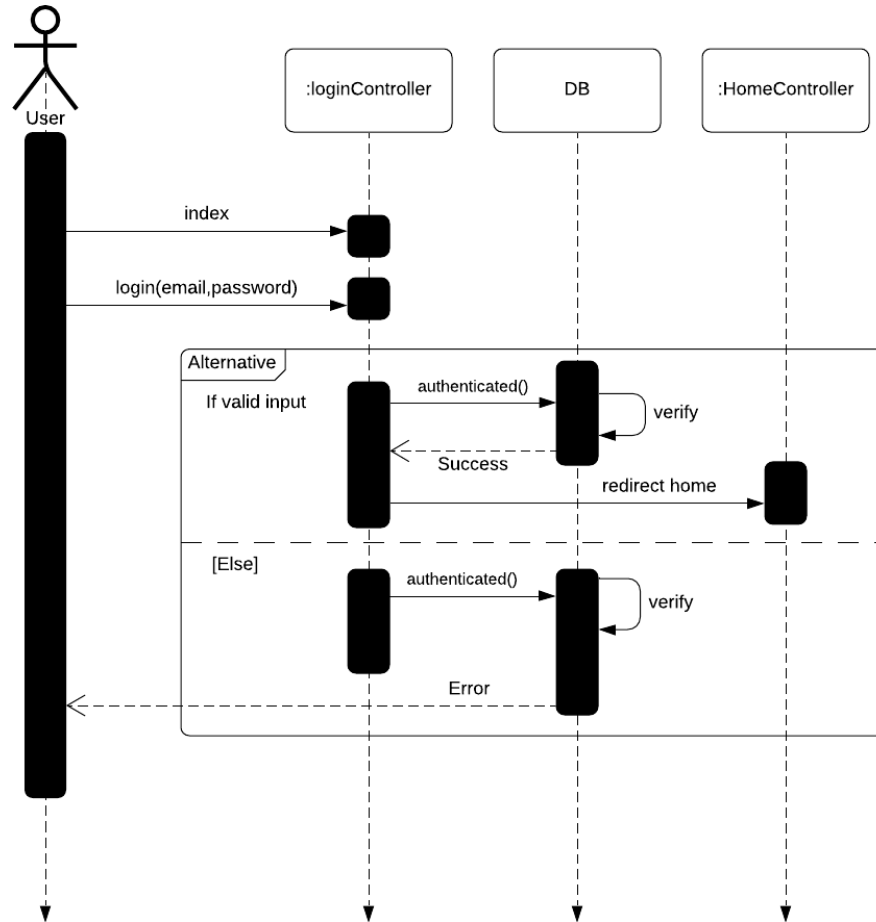


Fig 3.3.1.1 User login

User can login in system by input email and password then system check in DB after checking it from database then system redirect user in dashboard.

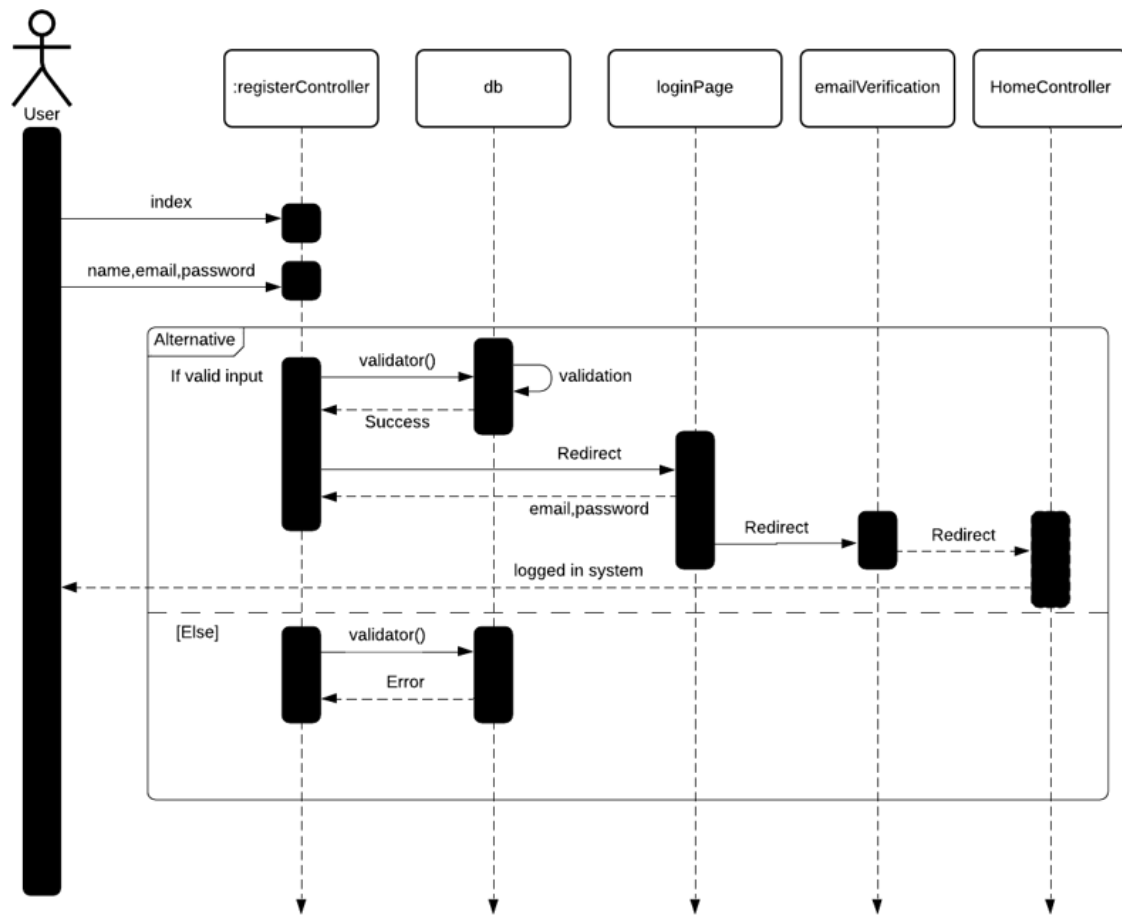


Fig 3.3.1.2 User Registration

After insert valid email address and password a verification email will sent in user email account after verify user will redirect it to homepage.

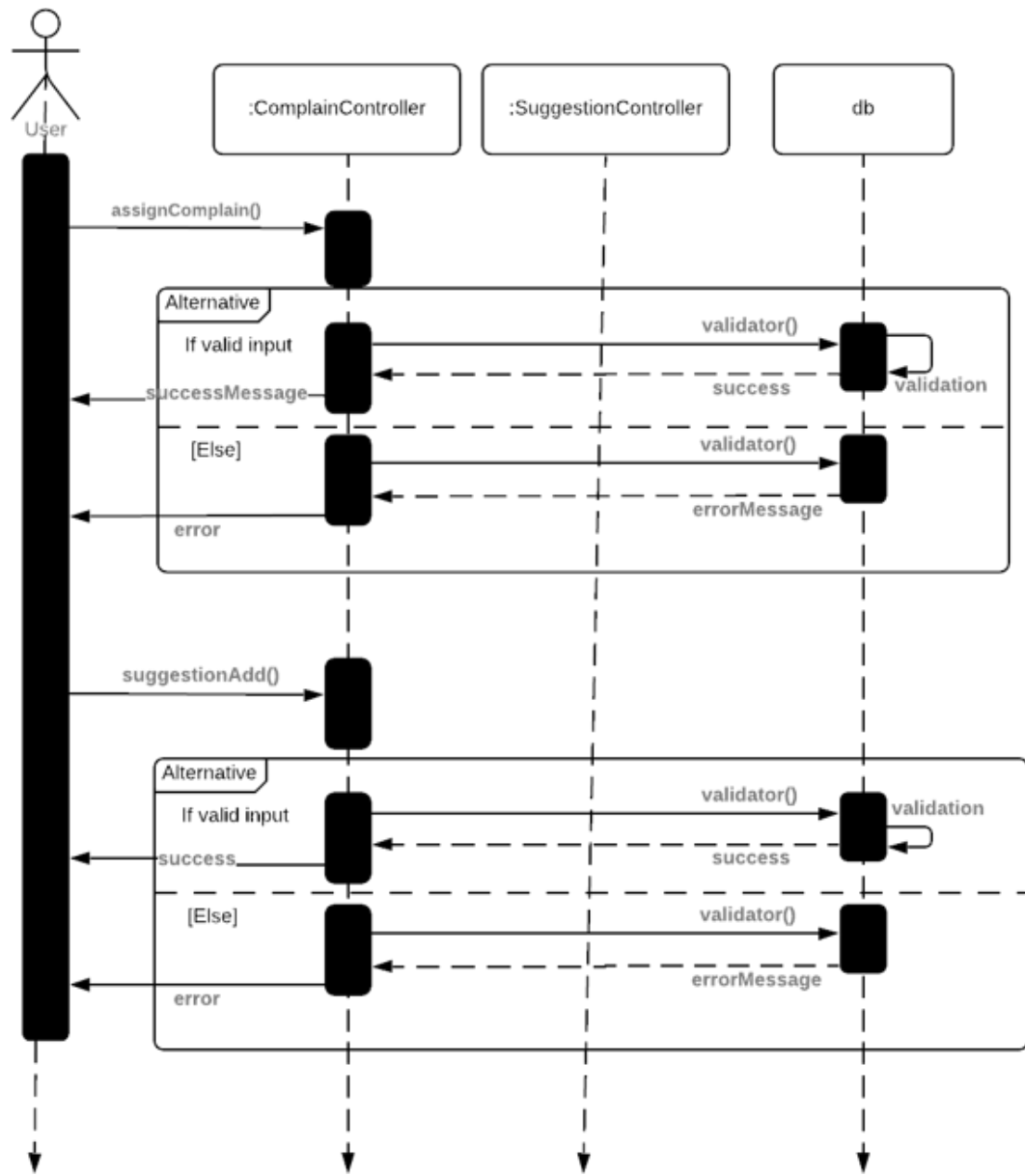


Fig 3.3.1.2 User Complain and Suggestion

User can add complain and suggestion insert valid information. A Custom validation can check user input data.

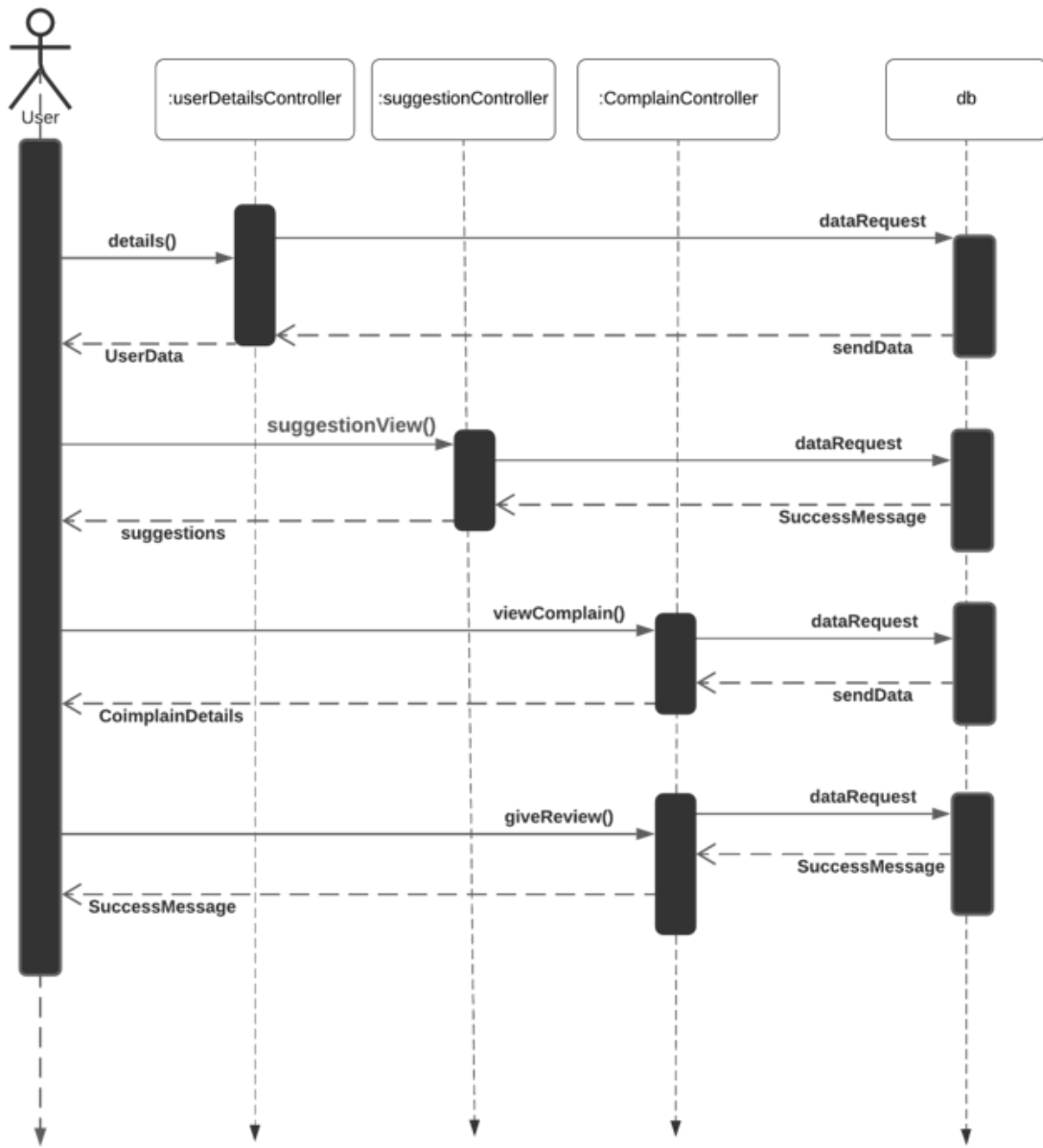


Fig 3.3.1.3 User Main Sequence

User can view complain and suggestion data by sending request.

3.3.2 Sequence Diagram for Admin

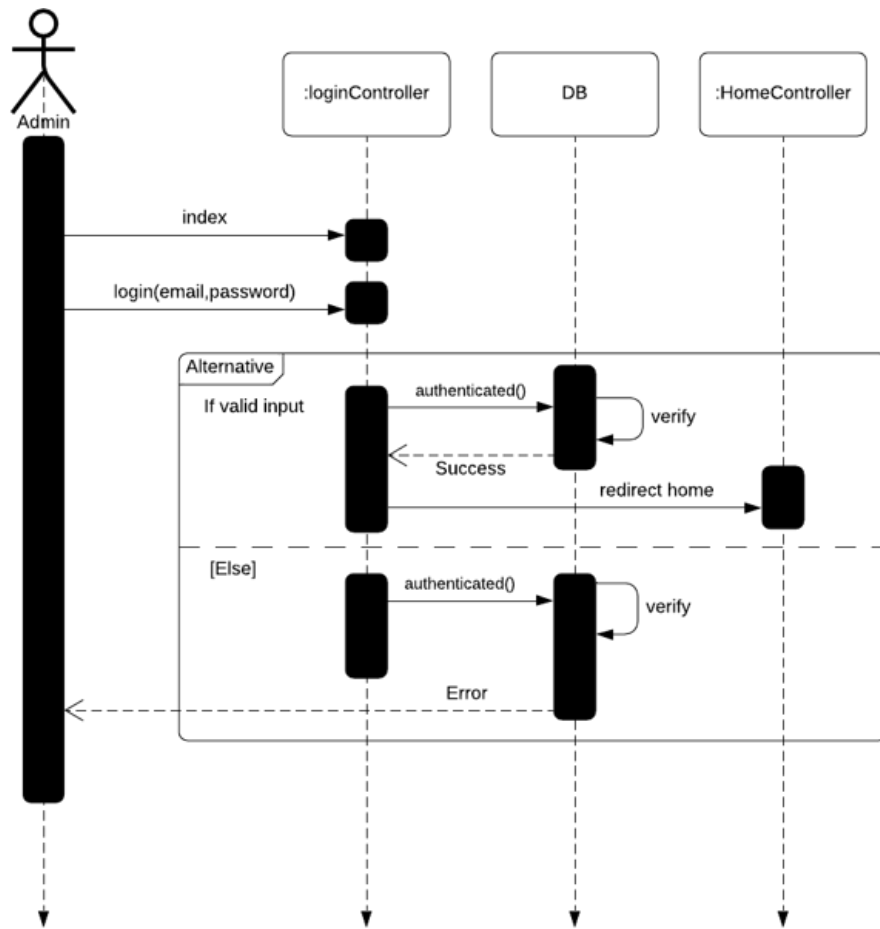


Fig 3.3.2.1 Admin login

Admin can login into system after providing valid information a validation middleware will check that the info admin provide is valid or not. If the information is valid then admin can access the system.

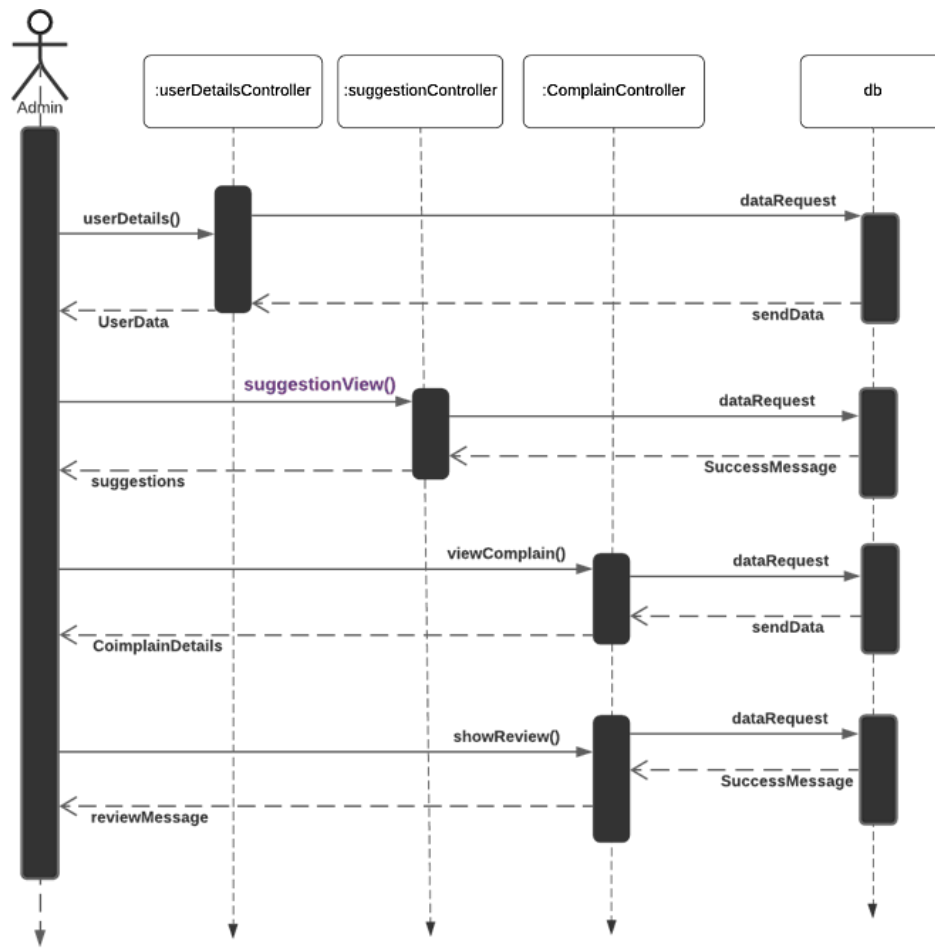


Fig 3.3.2.2 Admin Main Options

Admin can view user details, complains and suggestions by sending a request in system.

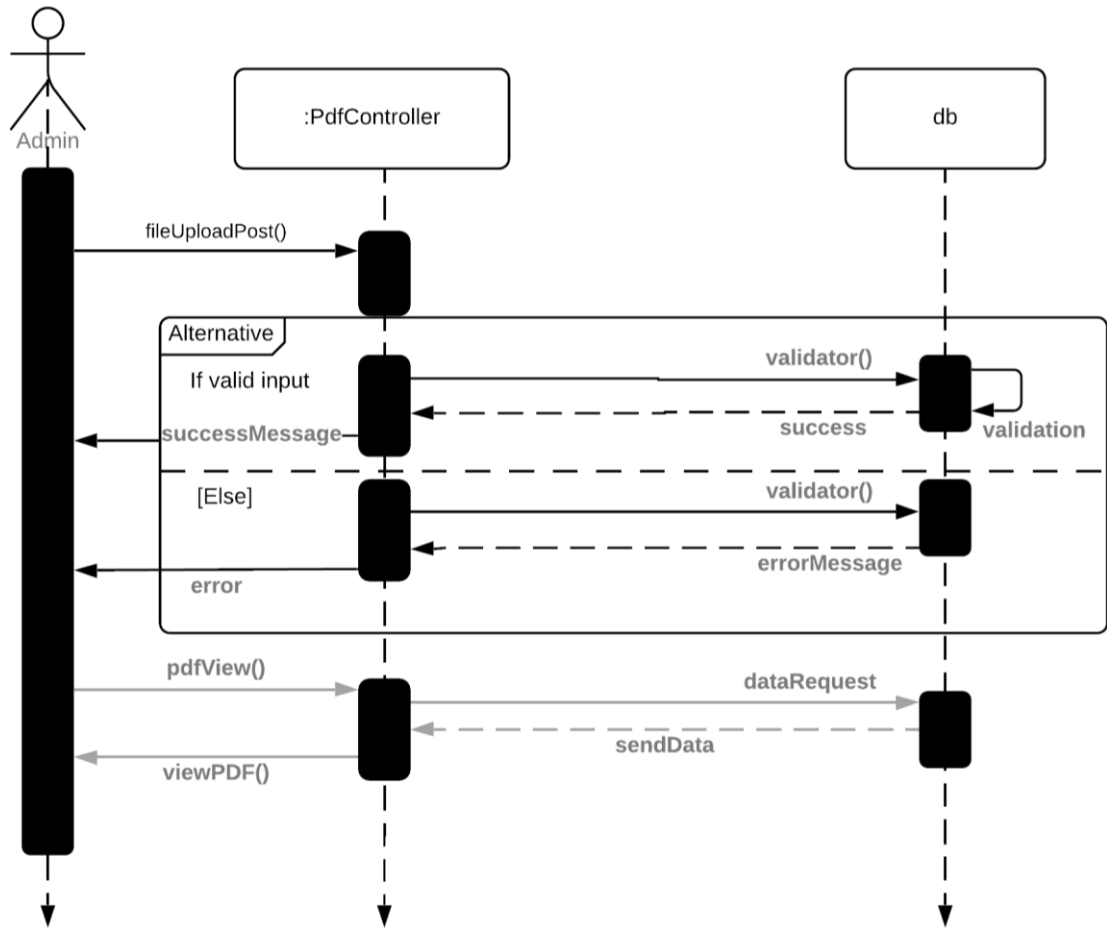


Fig 3.3.2.3 Admin Notice Update

Admin can upload notice by providing some information.

3.3.3 Sequence Diagram for Authority

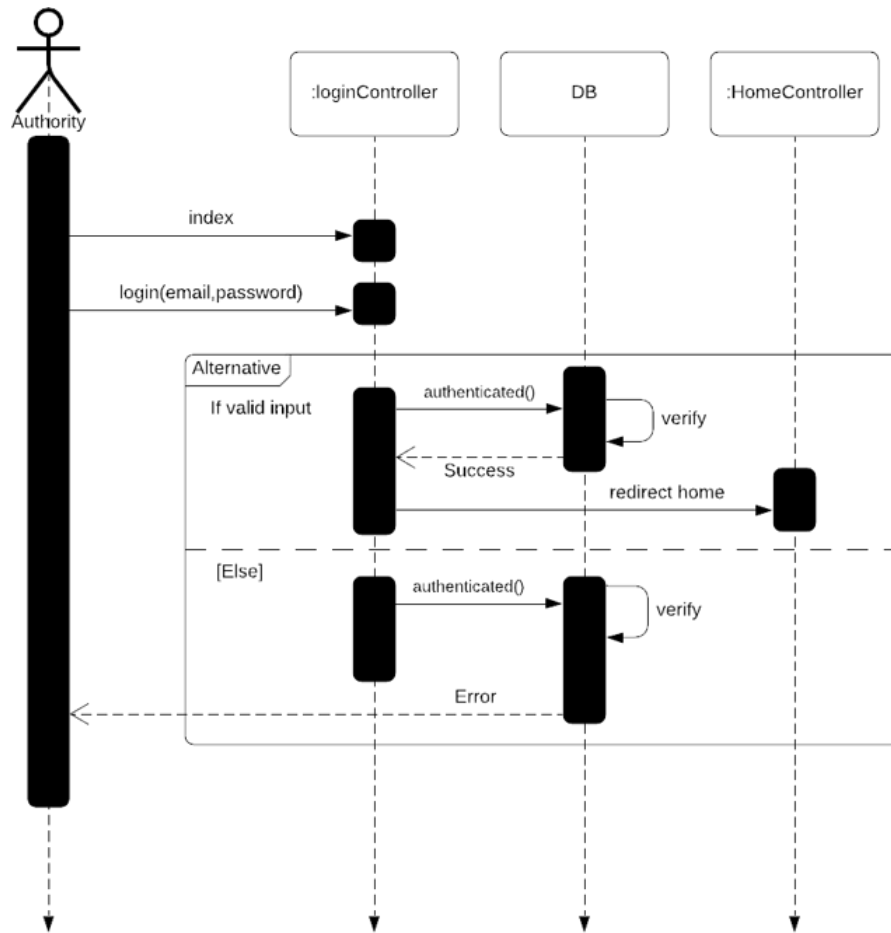


Fig 3.3.3.1 Authority Login

Authority can login into system after providing valid information a validation middleware will check that the info authority provide is valid or not. If the information is valid then authority can access the system.

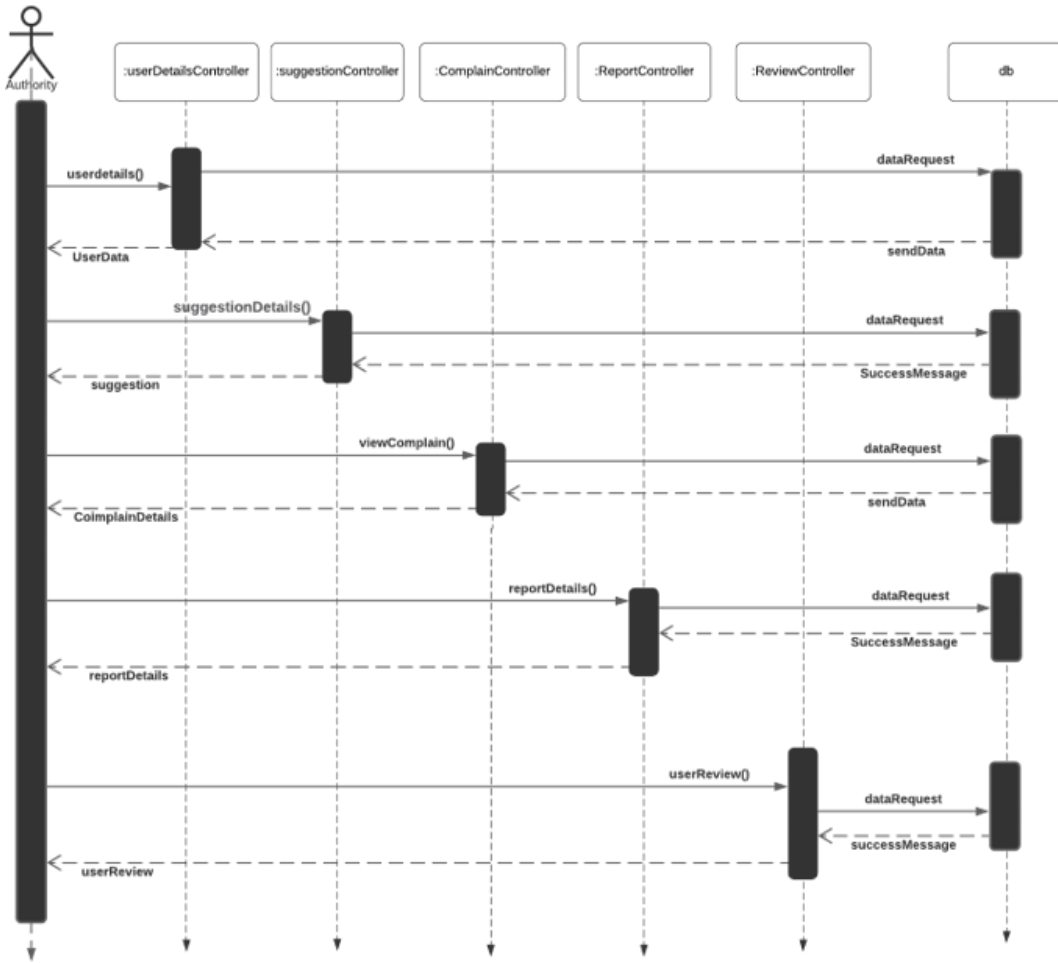


Fig 3.3.3.2 Authority Main Options

Authority can access user details, user complain, suggestions. Authority also check report which is generate by admin.

3.4 ERD Diagram

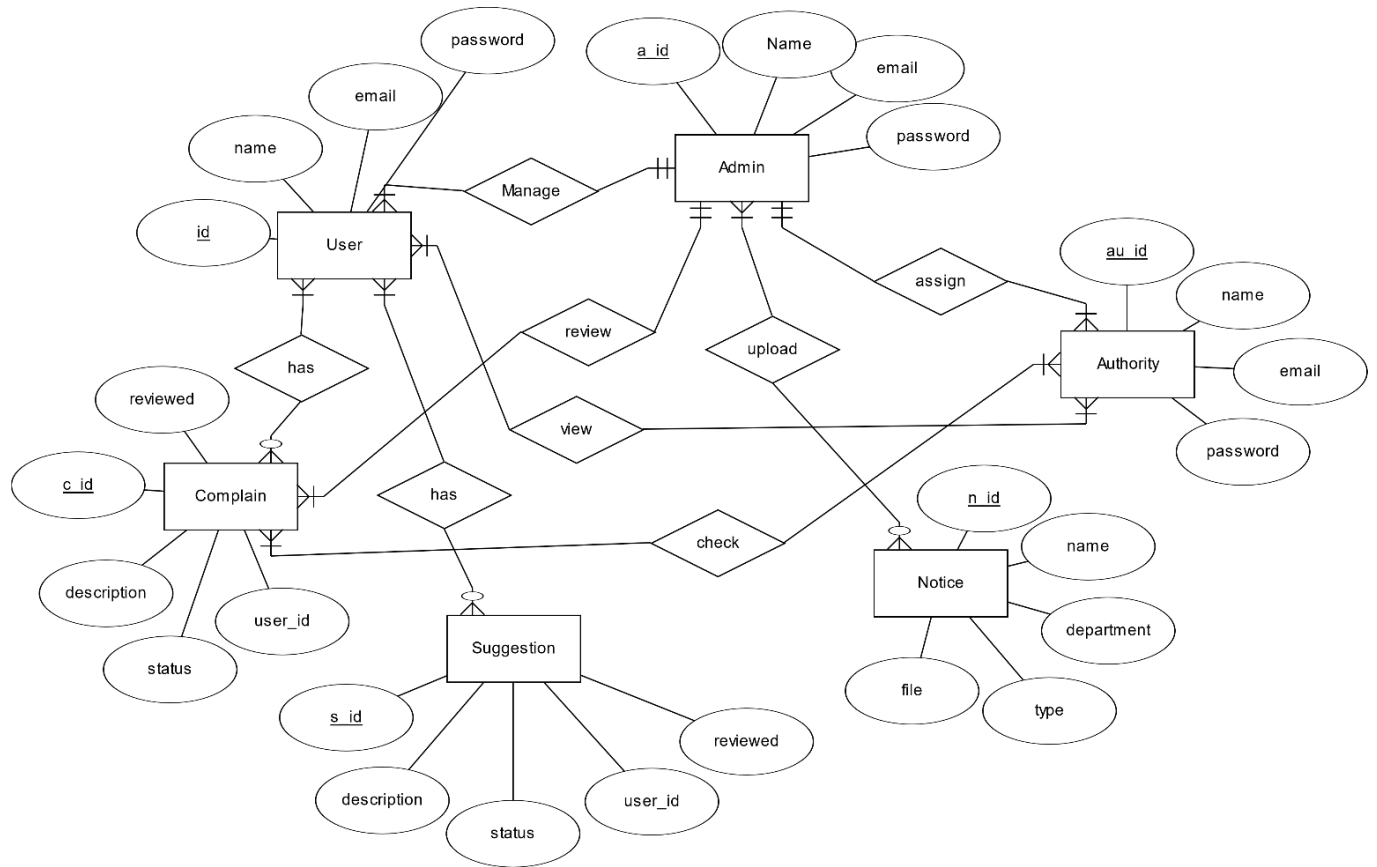


Fig 3.4.1 ERD Diagram [3]

ERD diagram Description: User can complain zero to many in this system. A user can have zero complain or many complain on other side a complain was came from one-to-many user. Same as in suggestion table A user can have zero suggestion or many suggestions on other side a suggestion was came from one-to-many user. Admin can manage minimum one user to many users. On the other hand, a user manages by one admin to many admin. A complain can solve by admin or many complain or an Admin can solve one complain to many complain. An authority can view user details or many user details. An admin can upload zero or many notice in this system.

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)

Xampp is an free open platform web server solution consisting mainly of The Apache, mainly consisting of Apache HTTP Server, Maria DB database, and PHP and Perl programming languages script interpreters. [4]

4.2.2 NPM(6.X)

NPM is a package manager for JavaScript and largest software registry. Discover package of reuse code and assemble them in new ways. [5]

4.2.3 PHP 7.3.27

PHP is hypertext Preprocessor is a general-purpose language for Web development. [6]

PHP Version 7.3.27



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: PhpStrom
- Command Prompt
- Server: LocalHost:8000

4.3.3 Version Control

- Git

CHAPTER 05

SYSTEM TESTING

5.1 Introduction

System testing is focused at error identification. The main principle of this activities are to ensure the process is correctly communicate and meet the requirement. The system testing help to test function effectively work or not.

5.2 Test Case

A Test case is a set of conditions or variables under so that a tester will determine the system under test satisfied requirement or work properly or not.

The developing process test case can help to find the problem in the requirement or design of an application.

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: Mostak Ahmed
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: mostak@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:22.10.2020
Test Title: User Register	Test executed by: Mostak Ahmed
Description: Test User Register	Test executed date: 22.10.2020
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: mostak@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:22.10.2020
Test Title: User Complain Section	Test executed by: Mostak Ahmed

Description: Test User create complain page	Test executed date: 24.10.2020
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:22.10.2020
Test Title: User suggestion Section	Test executed by: Mostak Ahmed
Description: Test User create suggestion page	Test executed date: 24.10.2020
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.
-----------------	---------------------------------

Test Case No -5

Test Case ID.1	Module name: Admin Login
Test Priority: High	Test Date:07.12.2021
Test Title: Admin login	Test executed by: Mostak Ahmed
Description: Test Admin login page	Test executed date: 07.12.2021
Pre-condition:	Admin must have valid email and password.
Test steps:	<ol style="list-style-type: none"> 1. Go to login page 2. Provide valid email & password 3. Click Login button
Test Data:	Admin: Email: Admin@gmail.com Password: 12345
Expected Results:	Admin should able to login
Actual Result:	Admin logged in successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully Logged in.

Test Case No -6

Test Case ID.1	Module name: Authority Login
Test Priority: High	Test Date:07.12.2021
Test Title: Authority login	Test executed by: Mostak Ahmed
Description: Test Authority login page	Test executed date: 07.12.2021
Pre-condition:	Authority must have valid email and password.

Test steps:	<ol style="list-style-type: none"> 1. Go to login page 2. Provide valid email & password 3. Click Login button
Test Data:	Authority: Email: Authority @gmail.com Password: 12345
Expected Results:	Authority should able to login
Actual Result:	Authority logged in successfully
Status (Pass/Fail):	Pass
Post-condition:	Successfully Logged in.

CHAPTER 06

USER MANUAL

6.1 User manual

FOR User

6.1.1 User login page [7]

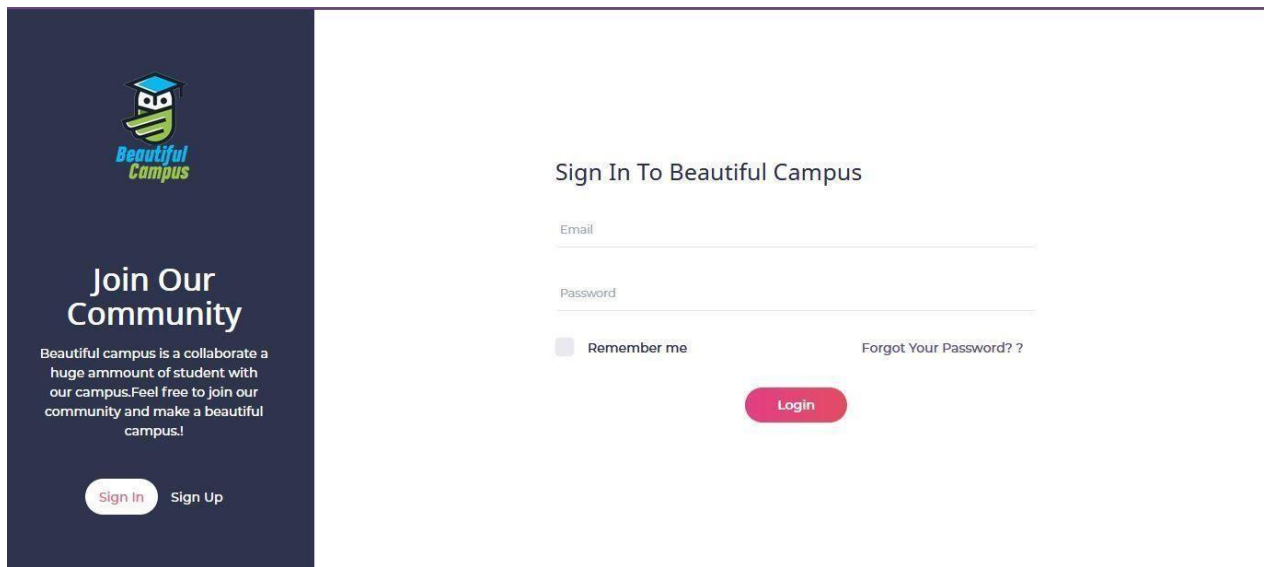


Fig 6.1.1 User Login page

6.1.2 User Registration page

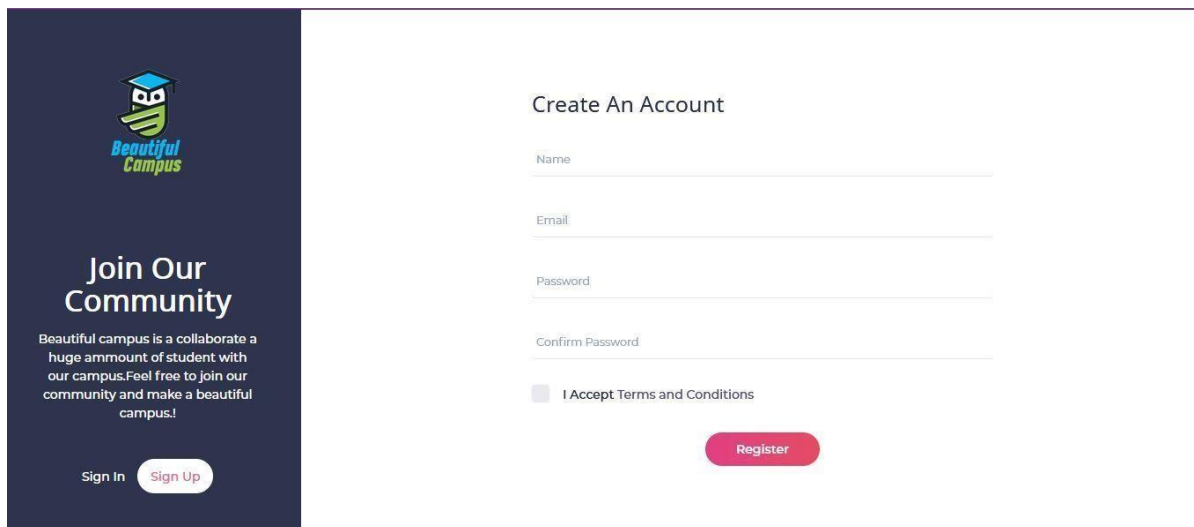


Fig 6.1.2 User Registration page

6.1.3 User Dashboard

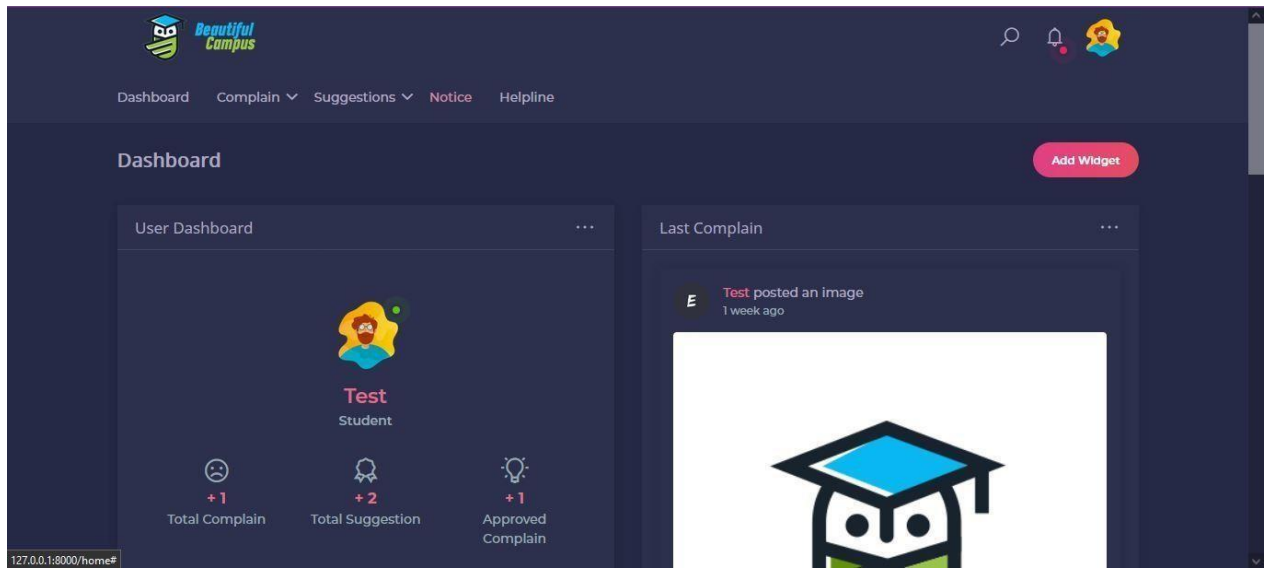


Fig 6.1.3.1 User Dashboard 1

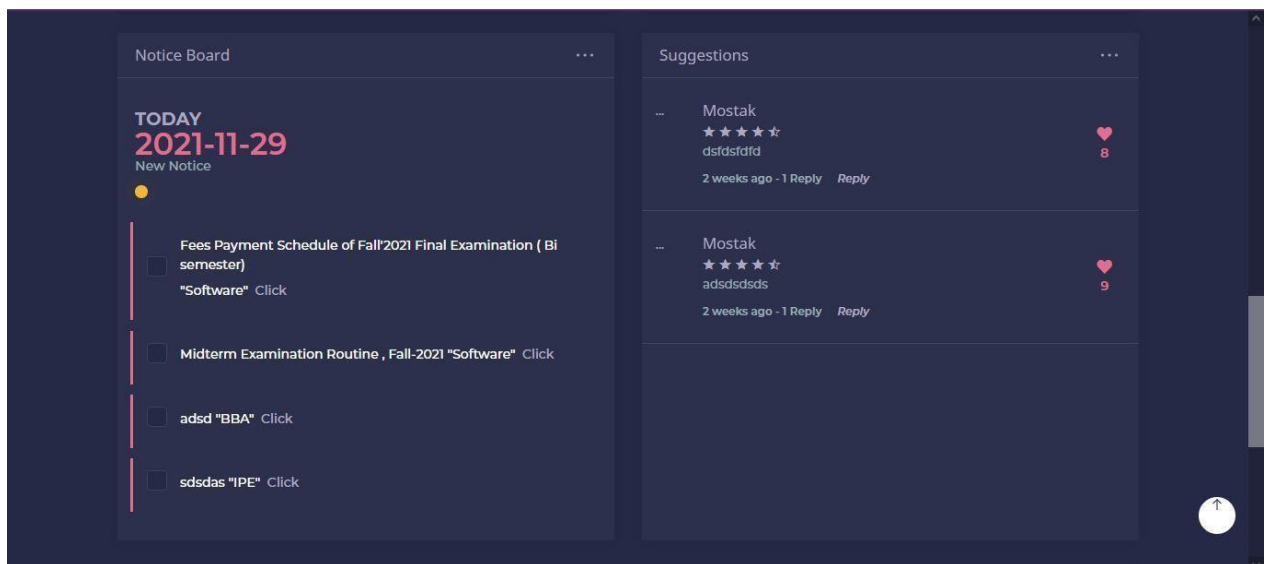


Fig 6.1.3.2 User Dashboard 2

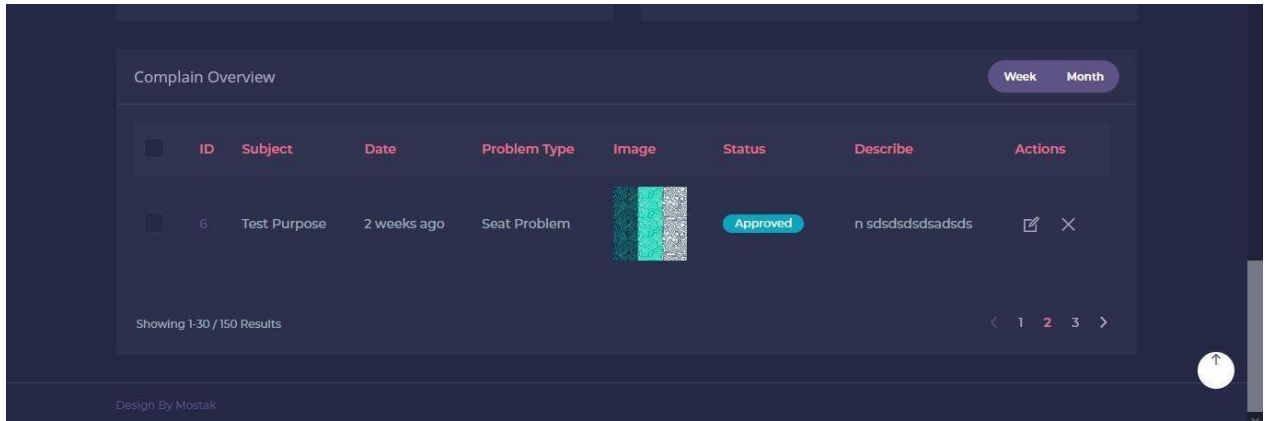


Fig 6.1.3.3 User Dashboard 3

6.1.4 User Complain

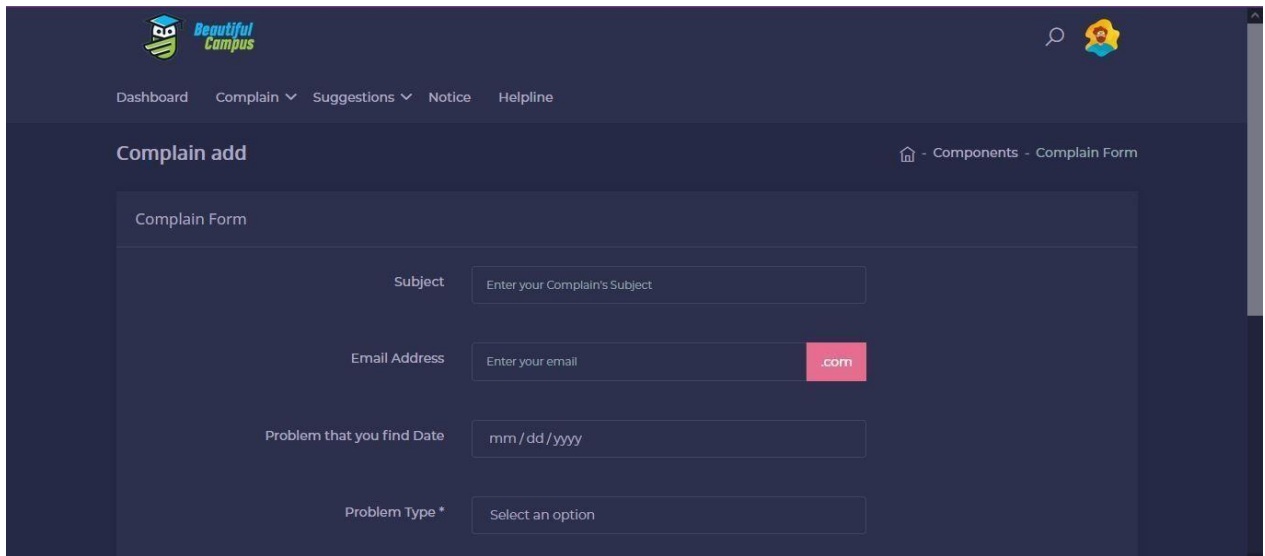


Fig 6.1.4 User Complain

6.1.5 User Complain View

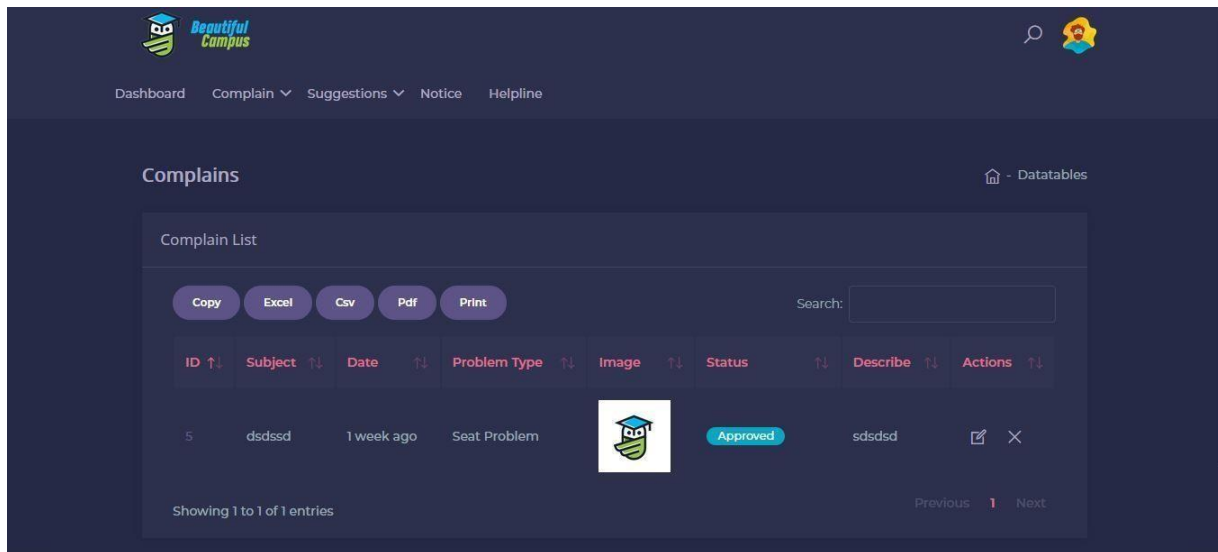


Fig 6.1.4 User Complain View

6.1.5 User Suggestion

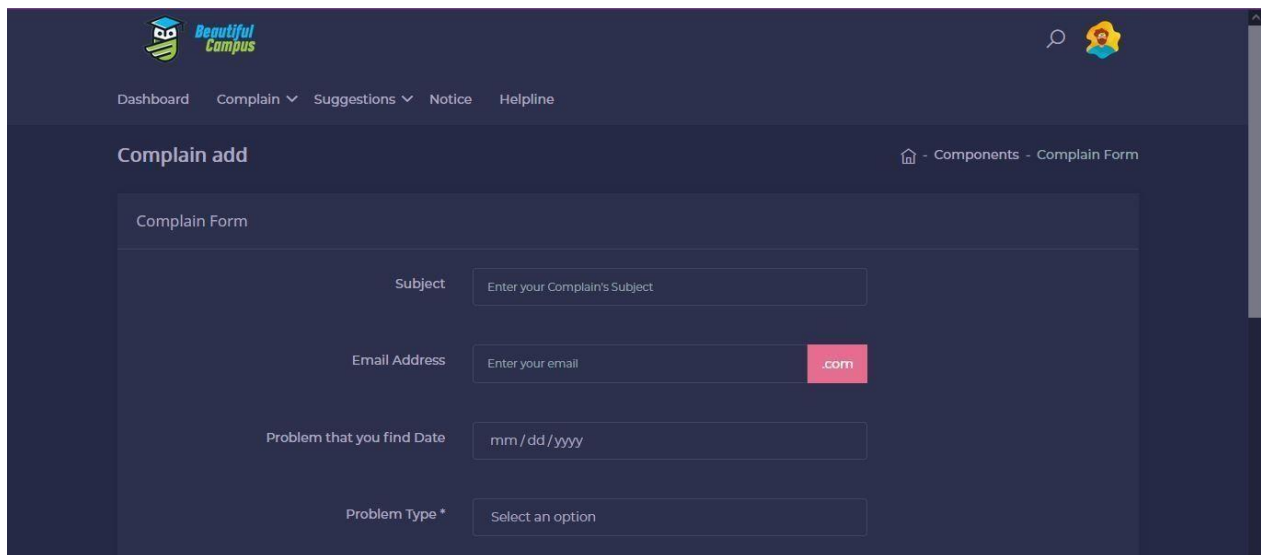


Fig 6.1.5 User Suggestions

6.1.6 User Suggestion View

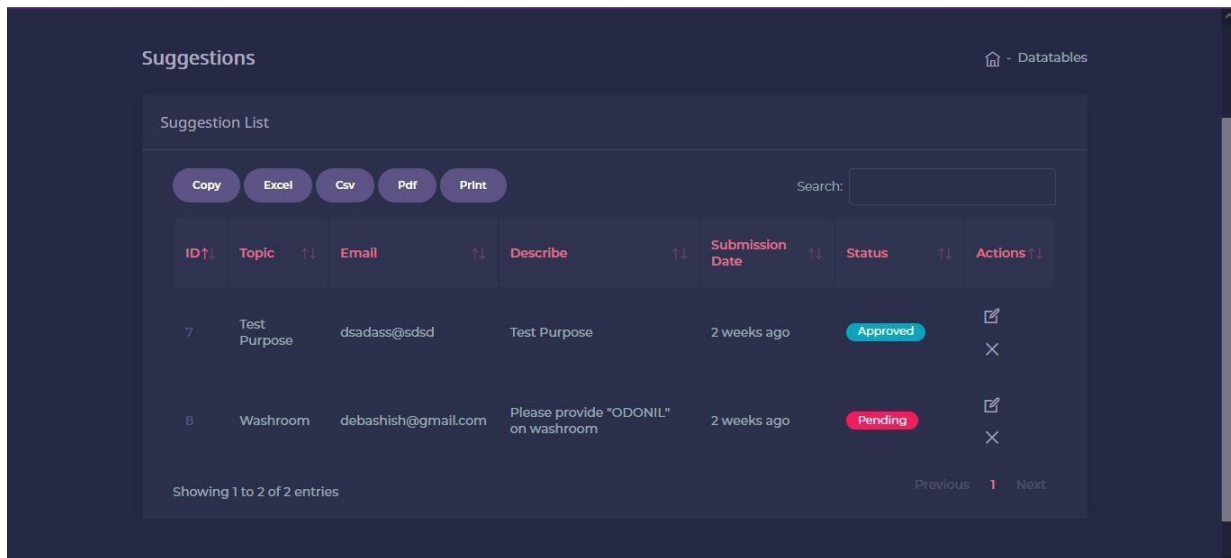


Fig 6.1.6 User Suggestions View

6.1.7 User Notice Board View

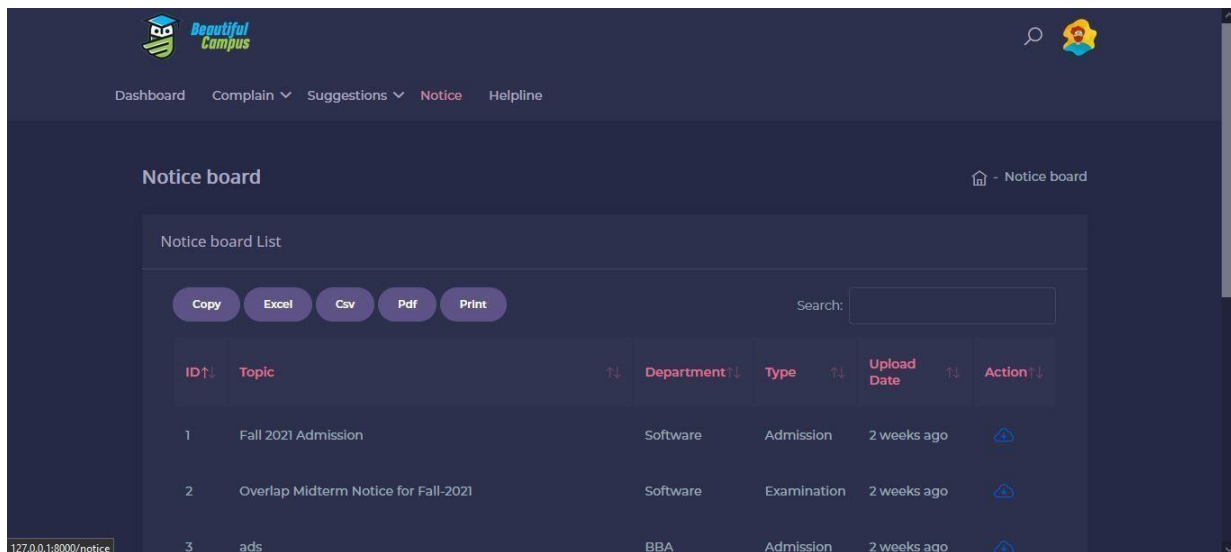


Fig 6.1.7 User Notice Board View

6.1.8 User Helpline

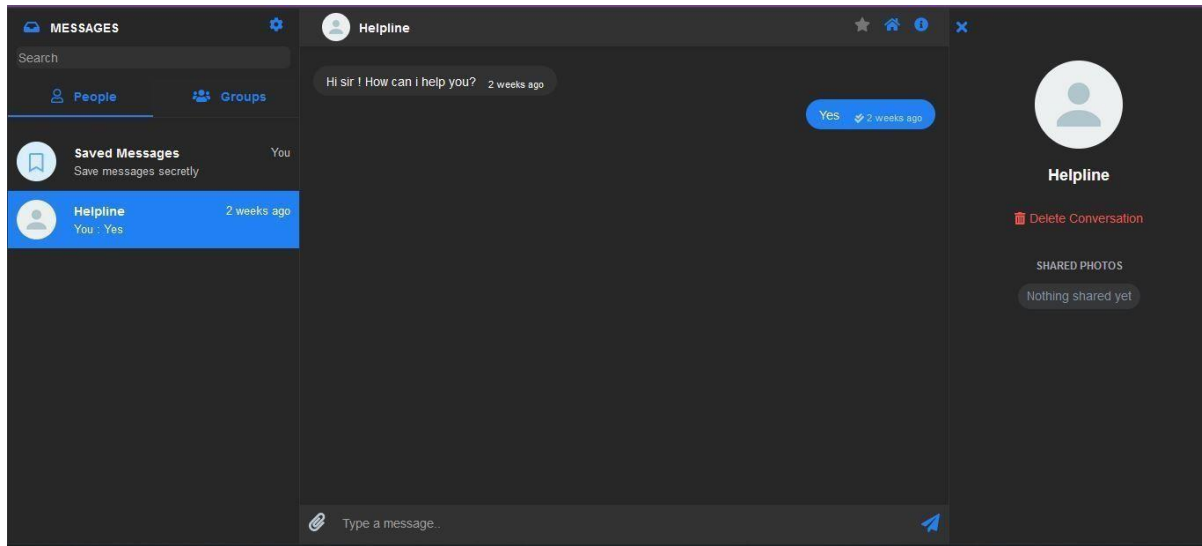


Fig 6.1.8 User Helpline

6.1.9 User Profile

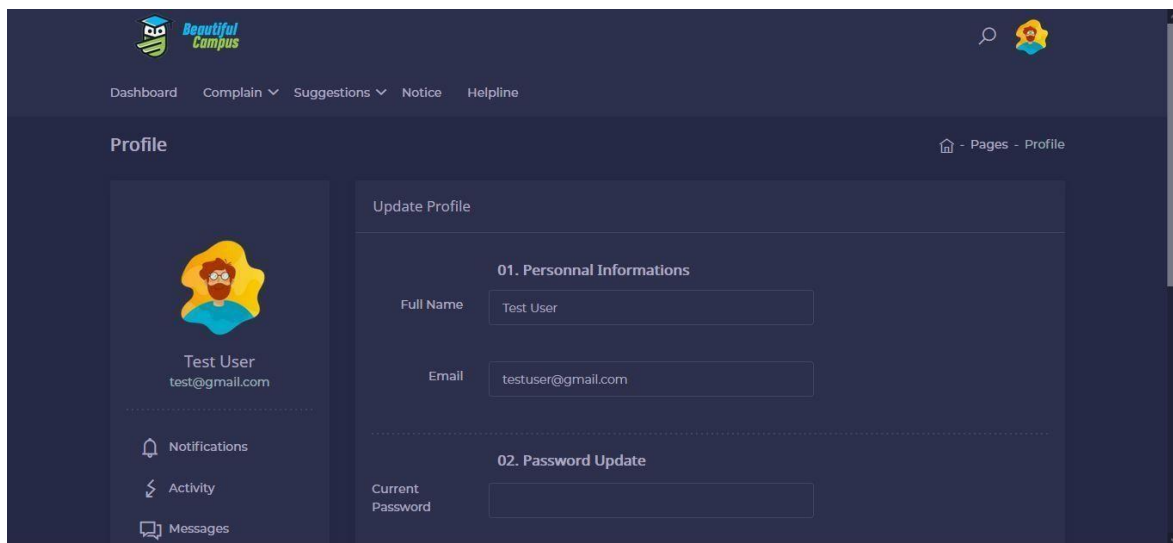


Fig 6.1.9 User Profile

6.1.10 User Notification

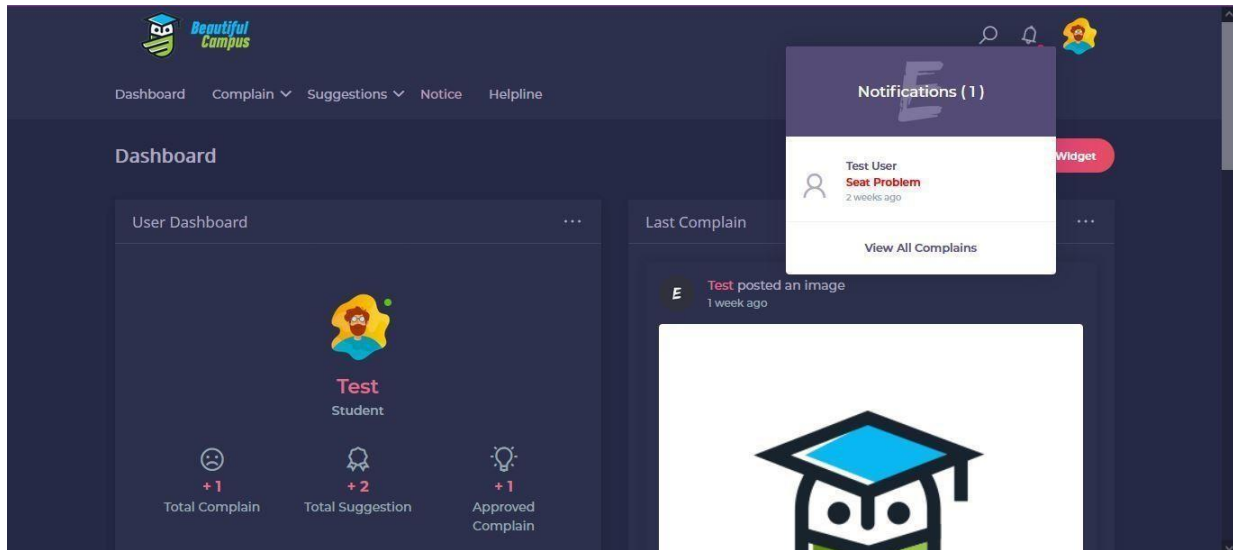


Fig 6.1.10 User Notification

For Admin

6.2.1 Admin Login

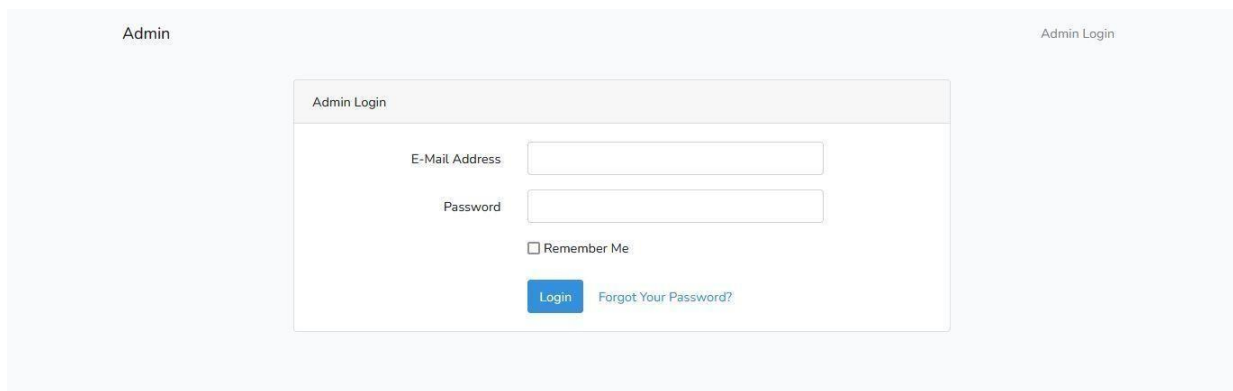


Fig 6.2.1 Admin Login

6.2.2 Admin Dashboard

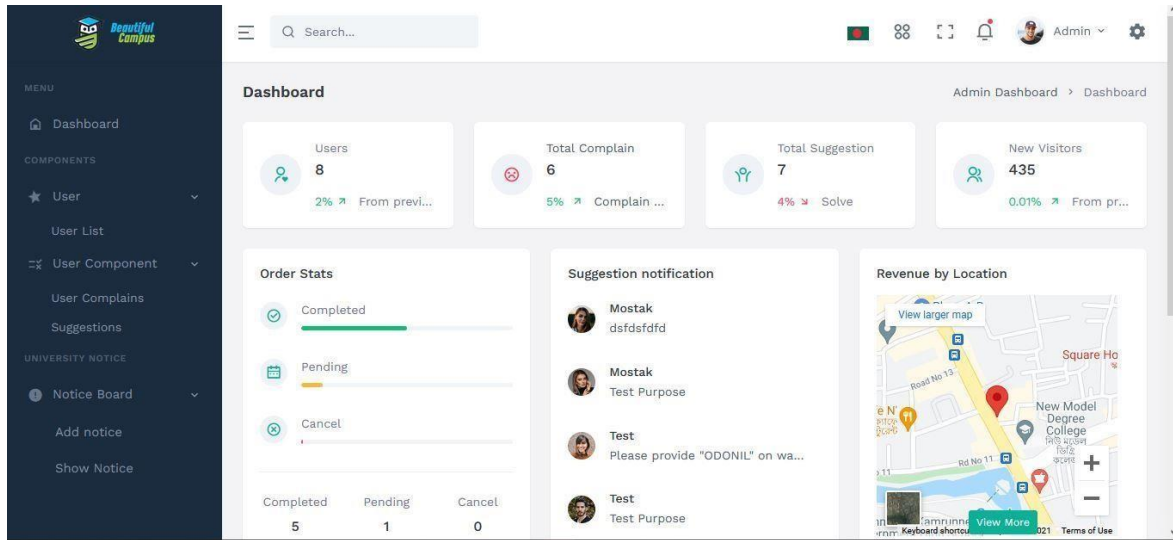


Fig 6.2.2 Admin Dashboard

6.2.3 Admin User List

The Admin User List page displays a table of all registered users, allowing for management and search. The table includes the following data:

ID	Name	Email	User from	Actions
2	Test	testuser@gmail.com	4 days ago	
3	Mostak	mostak@gmail.com	2 weeks ago	
4	Helpline	helpline@gmail.com	1 week ago	
5	Test Data	testdata@gmail.com	4 days ago	

Additional features include a search bar, table actions (Copy, Excel, PDF, Column visibility), and pagination (Showing 1 to 4 of 4 entries).

Fig 6.2.3 Admin User List

6.2.4 Admin User Complain

User Complain

Copy Excel PDF Column visibility Search:

ID	Subject	Date	Problem Type	Image	Status	Describe	Actions
3	Test Purpose	2 weeks ago	Washroom problem		Approved	Washroom problem	
4	Test	3 weeks ago	Washroom problem		Approved	asdsdasdsa	
5	Test Purpose	2 weeks ago	Seat Problem		Approved	sdsdsd	

Showing 1 to 3 of 3 entries

Fig 6.2.4 Admin User Complain

6.2.5 Admin User Suggestions

User Suggestion

Copy Excel PDF Column visibility Search:

ID	Topic	Email	Describe	Submission Date	Status	Actions
5	Test purpose	testuser@diu.edu.bd	Provide more light in 404AB	2 weeks ago	Pending	
7	Test Purpose	test@diu.edu.bd	Test Purpose	2 weeks ago	Approved	
8	Washroom	mostak@gmail.com	Please provide "ODONIL" on washroom	2 weeks ago	Pending	

Showing 1 to 3 of 3 entries

Fig 6.2.5 Admin User Suggestions

6.2.6 Admin Notice Add

The screenshot shows the 'Add Notice' form in the Beautiful Campus admin interface. The form includes the following fields:

- Name:** A text input field.
- Department:** A dropdown menu with 'Software' selected.
- Select:** A dropdown menu with 'Examination' selected.
- PDF:** A file upload area with a 'Browse...' button and the text 'No file selected.'
- Upload:** A green button to submit the notice.

The interface also features a dark sidebar menu on the left with options like 'Dashboard', 'User', 'User Component', 'User Complains', 'Suggestions', 'Notice Board', 'Add notice', and 'Show Notice'. The top right corner shows a search bar, user profile, and settings icons.

Fig 6.2.6 Admin Notice Add

6.2.7 Admin Notice Show

The screenshot displays the 'Notice Board' table in the Beautiful Campus admin interface. The table has the following columns: ID, Topic, Department, Type, Upload Date, and View. The table contains 6 rows of data. Above the table, there are buttons for 'Copy', 'Excel', 'PDF', and 'Column visibility', along with a search bar.

ID	Topic	Department	Type	Upload Date	View
1	Fall 2021 Admission	Software	Admission	2 weeks ago	
2	Overlap Midterm Notice for Fall-2021	Software	Examination	2 weeks ago	
3	ads	BBA	Admission	2 weeks ago	
4	Dev	EEE	Admission	2 weeks ago	
5	sdsdas	IPE	Academic	2 weeks ago	
6	adsd	BBA	Payment	2 weeks ago	

Fig 6.2.7 Admin Notice Show

6.2.8 Admin User Complain Approve

The screenshot displays the 'Notice Board' interface. On the left is a dark sidebar with a menu. The main content area features a search bar and a table of notices. The table has columns for ID, Topic, Department, Type, Upload Date, and View. Below the table are buttons for 'Copy', 'Excel', 'PDF', and 'Column visibility', along with a search input field.

ID	Topic	Department	Type	Upload Date	View
1	Fall 2021 Admission	Software	Admission	2 weeks ago	View
2	Overlap Midterm Notice for Fall-2021	Software	Examination	2 weeks ago	View
3	ads	BBA	Admission	2 weeks ago	View
4	Dev	EEE	Admission	2 weeks ago	View
5	sdsdas	IPE	Academic	2 weeks ago	View
6	adsd	BBA	Payment	2 weeks ago	View

Fig 6.2.8 Admin User Complain Approve

6.2.9 Admin Profile Update

The screenshot shows the 'Password Change' form. It includes three input fields for 'Old Password', 'New Password', and 'New Password', each with a masked password field. A green 'Change' button is positioned below the fields. The interface also features a sidebar, a search bar, and a notification bell icon.

Fig 6.2.9 Admin profile password update

For Authority

6.3.1 Authority Login

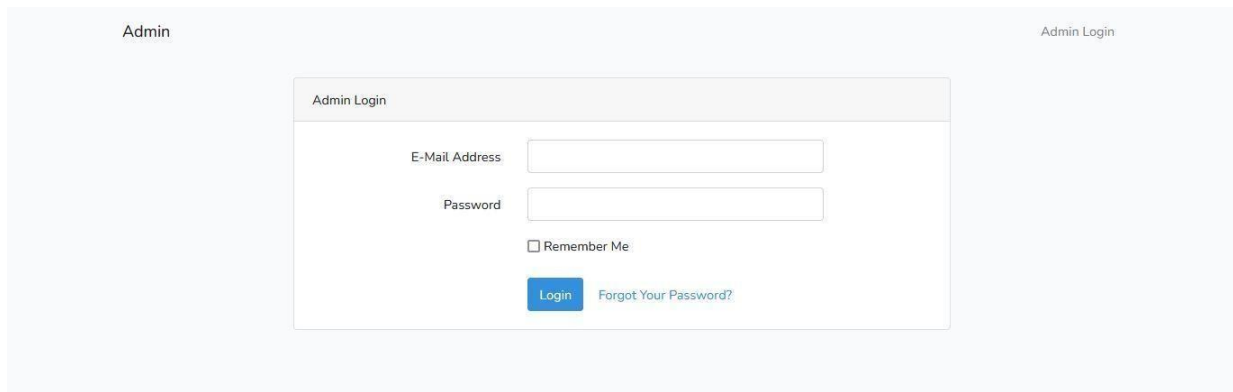


Fig 6.3.1 Authority Login

6.3.2 Authority Dashboard

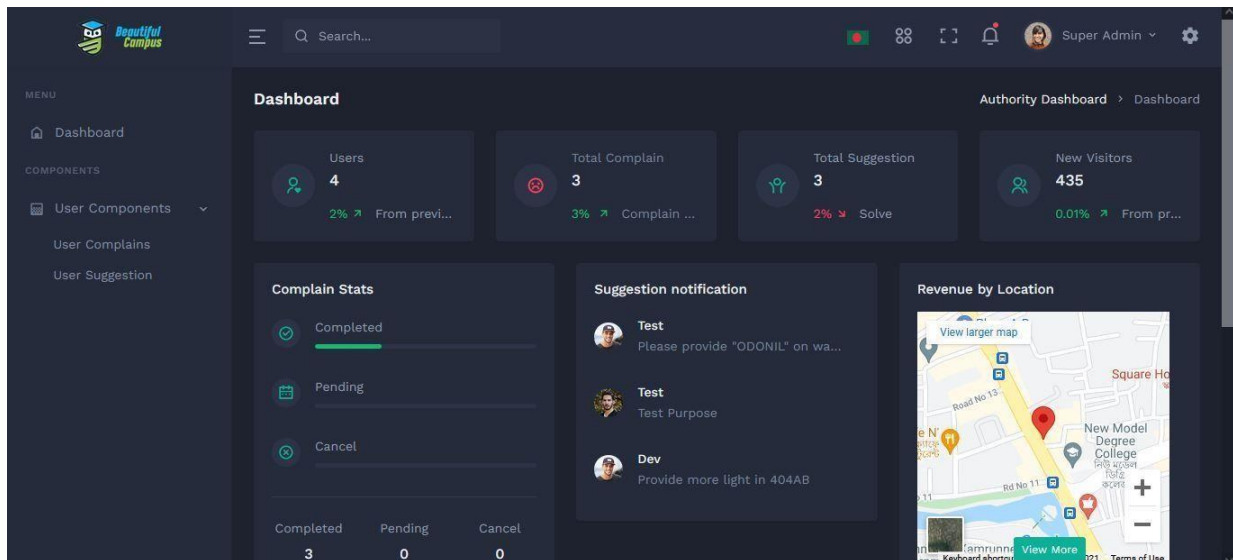


Fig 6.3.2 Authority Dashboard

6.3.3 Authority Dashboard 2

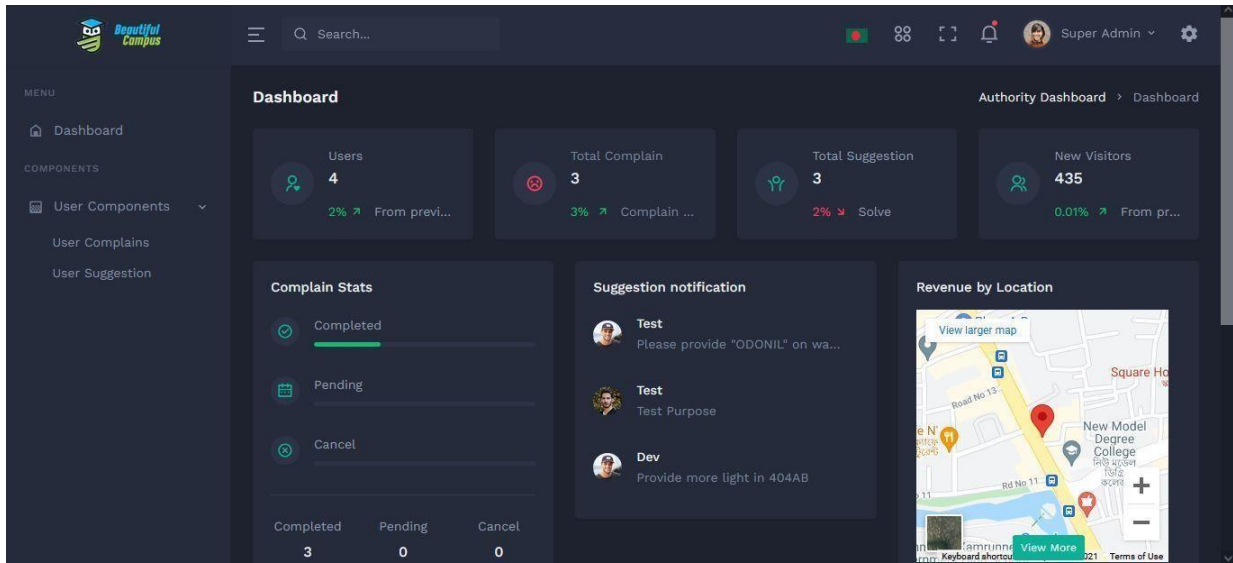


Fig 6.3.3 Authority Dashboard 2

6.3.4 Authority User Complain

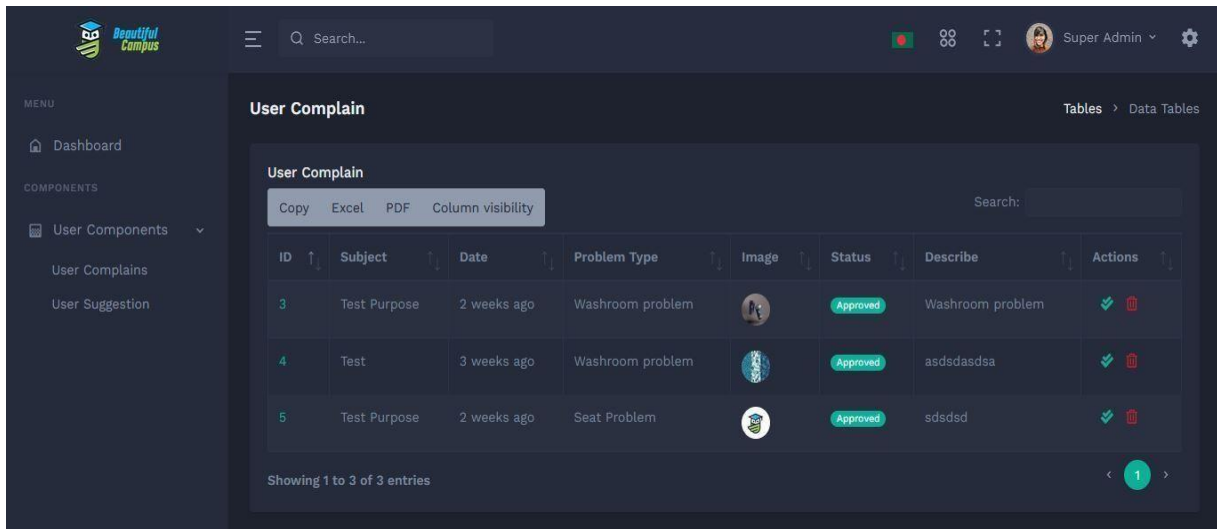


Fig 6.3.4 Authority User Complain

6.3.5 Authority User Suggestion

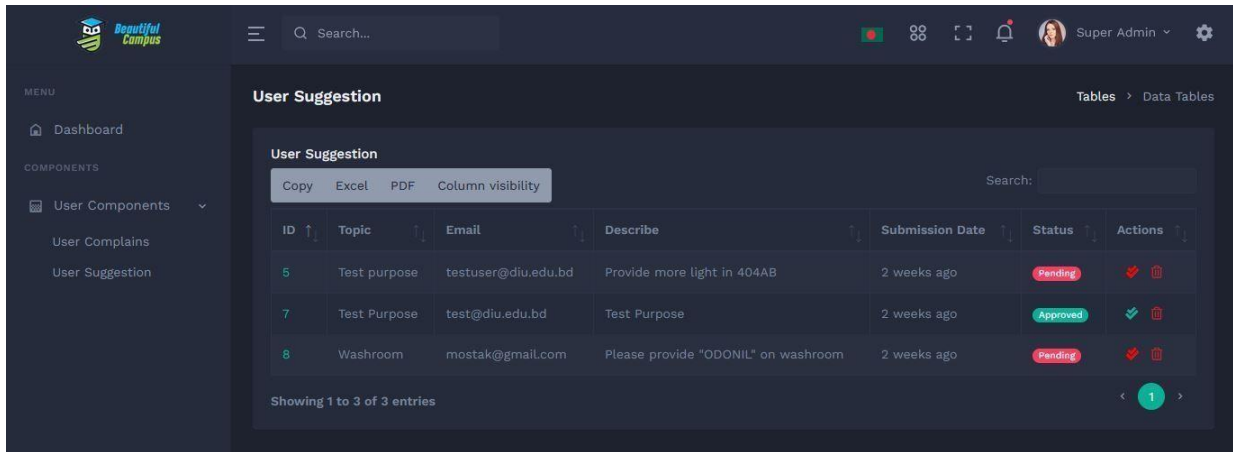


Fig 6.3.5 Authority User Suggestion

6.3.6 Authority User Complain Approve

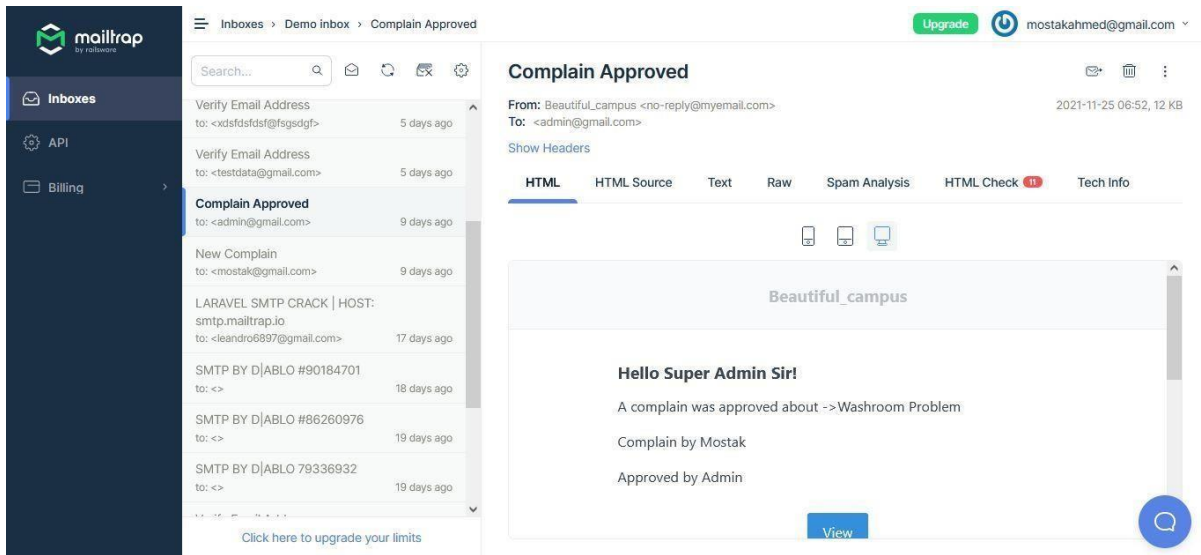


Fig 6.3.6 Authority User Complain Approve

6.3.6 Authority Profile update

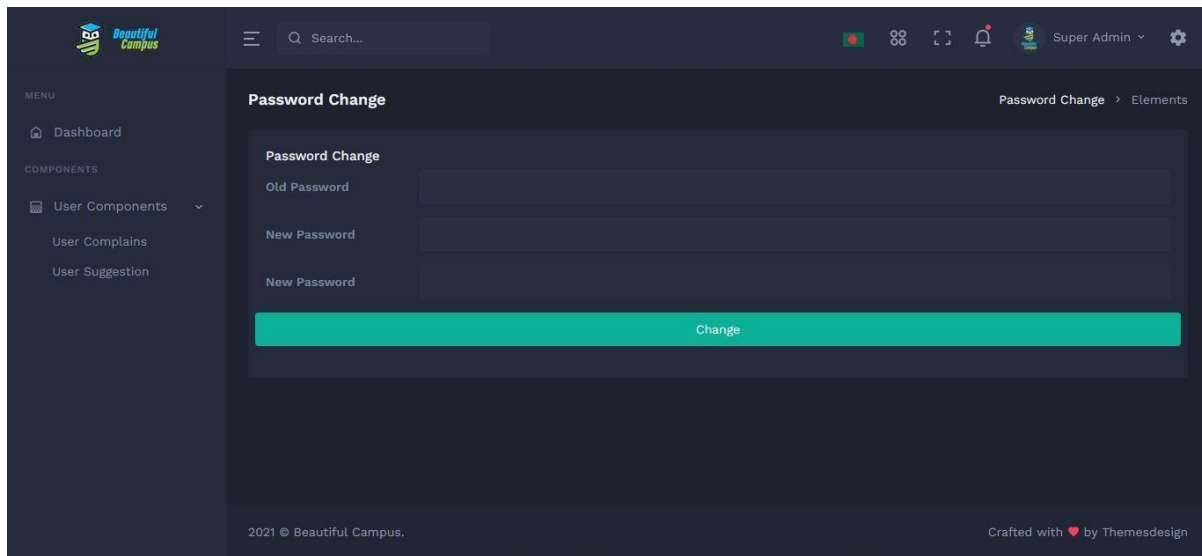


Fig 6.3.7 Authority Profile Update

CHAPTER 07

PROJECT SUMMARY

7.1 GitHub Link:

7.2 Limitations:

- User must need a high-speed internet.
- User must have a valid account to use the system
- The system doesn't accept unauthorize account.

7.3 Obstacle & Achievement

Obstacle:

- Learning new Technology and new Environment.
- Limited time and budget.

Achievement:

- Learn new technology.
- Solve a real time problem.

7.4 Conclusion

Although I created this system (Employee Bus Transport System), it has a lot of flaws. I did my best to overcome my restrictions. This technology will provide better service to the employees of a company, will allow them to reach the office on time. Basically, the system will provide transportation services to such employees and also increase the value of the company.

7.5 Future Work

In future we will add this project to live-server and try to solve some more major problems so that students can get this service through website.

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.

Turnitin Originality Report

Processed on: 23-Jan-2022 16:13 +06
 ID: 1746274280
 Word Count: 6627
 Submitted: 1

171-35-1952 By Mostak Ahmed

Similarity Index

22%

Similarity by Source

Internet Sources: 20%
 Publications: 2%
 Student Papers: 8%

7% match (Internet from 05-Jun-2020)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3927/P148856%20%2828-%29.pdf?isAllowed=y&sequence=1>

4% match (Internet from 05-Jan-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5683/171-35-1827%20%2819-%29.pdf?isAllowed=y&sequence=1>

2% match (Internet from 05-Jan-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5728/171-35-2043%20%2820-%29.pdf?isAllowed=y&sequence=1>

2% match (Internet from 05-Jan-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5719/171-35-1979%20%2820-%29.pdf?isAllowed=y&sequence=1>

1% match (Internet from 05-Jan-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5696/171-35-1904%20%2817-%29.pdf?isAllowed=y&sequence=1>

1% match (Internet from 05-Jan-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5725/171-35-2016%20%2821-%29.pdf?isAllowed=y&sequence=1>

1% match (student papers from 02-Dec-2018)

[Submitted to University of Greenwich on 2018-12-02](#)

< 1% match (Internet from 15-Mar-2020)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3553/P13659%20%2829%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 05-Jan-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5718/171-35-1958%20%2824-%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 06-Jan-2020)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3547/P13646%20%2824%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 28-Jul-2019)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2940/P12395%20%285%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 24-Dec-2021)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6567/151-35-1126%20%2818-%29.pdf?isAllowed=y&sequence=1>

< 1% match (student papers from 16-Jan-2022)

[Submitted to Daffodil International University on 2022-01-16](#)