OFFICIAL NOTIFICATIONS ARCHIVE

Final Year Project

Ateeq-Ur-Rehman Gondal

Session (2013-2017)

Supervised By

Dr. Akmal Saeed Khatak

Department of Computer Sciences,

Quaid-i-Azam University,

Islamabad, Pakistan.

A report submitted to Department of Computer Sciences, Quaid-i-Azam University, Islamabad as partial fulfillment of the requirements for the award of the degree of BS Computer Science.

ACKNOWLEDGEMENT

First, I would like to express my sincere and humble gratitude to Almighty Allah whose blessings and guidance has been a real source of all the achievements in my life.

I would like to thank my project supervisor Dr. Akmal Saeed Khattak. The door to my supervisor office was always open whenever I ran into a trouble spot or had a question about my project. He consistently helped me.

I would also like to acknowledge my honorable teachers Dr. Afzal Bhatti, Dr. Onaiza, Dr. Ghazanfar Farooq, Dr. Mudassar Azam Sindhu, Dr. Mubashar Mushtaq, Dr. Khalid Saleem, Dr. Rabeeh, Ms. Memona, Ms. Iffrah Farukh, Mr. Naqi and Dr. Usman for their valuable support during my studies.

I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. I am thankful to my brothers and sisters for their prayers and moral support during my educational career.

Finally, I would acknowledge my friends, class fellows, who were there for me always. Their time and company helped a lot being here. I would also like to thank Aneeq-Ur-Rehman and Muhammad Aaliyan. They were true motivation for me. Whenever I go to home, they are the reason of my smile. They are true friends. I greatly acknowledge financial support from my brother Abdul Wahid Gondal during my studies.

Thanks,
Ateeq-Ur-Rehman Gondal
BS CS, Quaid-i-Azam University Islamabad.

DEDICATION

I dedicate this project to my parents, teachers, family and friends.

Table of Contents

CHAPTER 1 SOFTWARE PROJECT MANAGEMENT

1.1 Introduction 1

1.1.1 Project Overview 1

1.1.2 Project Deliverables 2

1.2 Project Organization 4

1.2.1 Software Process Model 4

1.2.2 Tools and Techniques 5

1.3 Project Management Plan 6

1.3.1 Gant Chart 7

CHAPTER 2 SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 Introduction 8

2.1.1 Archives 8

2.1.2 Project Overview 9

2.2 Specific Requirements 10

2.2.1 Proposed System 10

2.2.2 Motivation 11

2.2.3 Related Work 11

2.2.4 Purpose 13

2.2.5 Scope 14

2.2.6 Interfaces 14

2.2.7 Software Product Features 15

2.2.8 Use Cases Description 16

2.2.9 Use Cases Diagram 30

2.2.10 Software System Attributes 31

2.2.11 Database Requirements 32

CHAPTER 3 SOFTWARE DESIGN DESCRIPTION

3.1 Introduction 34

3.1.1 Design Overview 34

3.1.2 Requirements Traceability Matrix 34

3.2 Software Architectural Design 35

3.2.1 Chosen System Architecture 35

3.3 Detailed Description of Components 37

3.3.1 Login Form 37

3.3.2 Registration Form 37

3.3.3 Notification Form 37

3.3.4 Notification Controller 37

3.3.5 Models 37

3.3.6 Registration Controller 37

3.3.7 Authentication Controller 38

3.3.8 Category Controller 38

3.3.9 Tag Controller 38

3.4 Detailed Design 39

3.4.1 Domain Model 39

Conceptual Classes 39

Domain Model Diagram 40

3.4.2 Sequence Diagrams 41

3.4.3 Class Diagram 49

3.5 User Interfaces Design 50

3.5.1 User Interface Screens 50

CHAPTER 4 SOFTWARE IMPLEMENATIONS

4.1 Introduction 56

4.2 Programming Language 56

4.3 Laravel Framework 56

4.4 Software and Tools used 56

4.5 Screens 57

4.5.1 Screens for Admin 57

4.5.2 Screens for Faculty 62

CHAPTER 5 SOFTWARE TEST DOCUMENTATION

5.1 Introduction 65

5.2 System Overview 65

5.2.1 Test Approach 65

5.3 Test Plan 66

5.3.1 Features to be tested 66

5.4 Test Cases 66

5.4.1 Log in 66

5.4.2 Sign up 68

5.4.3 Search Notification 71

5.4.4 Add Notification 73

5.4.5 Add Category 75

5.4.6 Browse a Category 77

5.5 Test Results 78

CHAPTER 6 CONCLUSIONS

6.1 Introduction 79

6.2 Conclusion 79

6.2.1 Future Enhancement 79

6.3 Extra Work 80

References

List of Figures

Figure 1. 1 Project Plan 6

Figure 1. 2 Gant Chart 7

Figure 2. 1 Use Case diagram 30

Figure 2. 2 Entity Relationship Diagram 33

Figure 3. 1 Traceability matrix 34

Figure 3. 2 System Architecture Diagram 36

Figure 3. 3 Component diagram 38

Figure 3. 4 Domain Model 40

Figure 3. 5 Sign up sequence diagram 41

Figure 3. 6 Log in sequence diagram 42

Figure 3. 7 Add Notification sequence diagram 43

Figure 3. 8 Add Category sequence diagram 44

Figure 3. 9 Browse Category sequence diagram 45

Figure 3. 10 Search Notification sequence diagram 46

Figure 3. 11 View Notification sequence diagram 47

Figure 3. 12 Delete Notification sequence diagram 48

Figure 3. 13 Class Diagram 49

Figure 3. 14 Login UI 50

Figure 3. 15 Create account UI 51

Figure 3. 16 Search Notification UI 52

Figure 3. 17 Search results UI 53

Figure 3. 18 Category View UI 54

Figure 3. 19 Category browse UI 55

Figure 4. 1 Sign Up Screen 57

Figure 4. 2 Sign in screen 58

Figure 4. 3 Home Screen 58

Figure 4. 4 Home Screen 2 59

Figure 4. 5 Add Notification Screen 59

Figure 4. 6 Browse Notifications Screen 60

List of Figures

Figure 4. 7 All Notifications Screen 60

Figure 4. 8 Add Category Screen 61

Figure 4. 9 All Categories 61

Figure 4. 10 Send Email Screen 62

Figure 4. 11 Register Screen 62

Figure 4. 12 Sign in Screen 63

Figure 4. 13 Search Screen for users 64

Figure 4. 14 Browse Categories Screen 65

ACRONYMS AND ABBREVIATIONS

XP	Extreme Programming
ASD	Adaptive Software Development
DSDM	Dynamic System Development Method
FDD	Feature Driven Development
AM	Agile Modeling
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
SQL	Structured Query Language
MVC	Model View Controller
UC	Use Case
TC	Test Case

CHAPTER 1

SOFTWARE PROJECT MANAGEMENT

1 SOFTWARE PROJECT MANAGEMENT

1.1 Introduction

Official Notifications Archive is a project designed to manage official notifications in Department of Computer Sciences, Quaid-i-Azam University Islamabad. It is a web based archive. Notifications will be added in the archive from Computer Science Office. Faculty members will be able to search, browse and download notifications from this archive.

1.1.1 Project Overview

These days, archives are a great source to get information quickly. Archives of different types exists e.g. colleges and university archives, research archives, religious archives. Official Notifications Archive is designed to archive official notifications.

Opportunity

The Information Age, also called the Computer Age, the Digital Age and the New Media Age, is coupled tightly with the advent of personal computers. These computers are used in every part of life to get information. Old style paper based files are being converted to archives. There is no such archive exist in Computer Science Department, Quaid-i-Azam University Islamabad. Notifications are managed manually in traditional paper based files. Managing notification in this way is quite a difficult and time-consuming task. If someone needed a notification which was issued 2 or 3 years ago, it will be very difficult to get this notification. It will take a lot of time to find notification. Official Notifications Archive will help to search notifications online. The archive will take Department of Computer Science toward advancement.

Goal

The goal of the project is to design such archive which will enable computer science department to search notifications quickly. This is a web based application providing basic search facilities on these notifications. Application will provide a simple user interface to add or search notifications.

Objectives

The objective of the project is to make archive of official notifications so that these notifications can be searched easily. Notifications will be fed as images in the archive and will be tagged under different categories. Different categories are Scholarships, Students, Faculty members etc. Notifications will be searchable by date, tag and title. Notification can be browsed by categories and year.

1.1.2 Project Deliverables

Project deliverables are the products resulted during a project execution. Deliverables are necessary for completion of project. These deliverables test the milestones achieved during a project. Each deliverable is a step toward finishing of project.

1. System overview

In this deliverable product overview, will be explained.

2. What is proposed system?

In this document, the proposed system will be explained. What is proposed system and system details will be explained. Motivation, related work and proposed solution will also be explained.

3. Project Management Plan

In this deliverable, project planning will be discussed. Tasks will be identified and defined.

4. Software Requirement Specifications

This deliverable will explain software requirement specifications. In this deliverable purpose of the product, scope of the product, product perspective like software interfaces, hardware interfaces, user interfaces and communication interfaces will be described. User characteristics and main software features will also be described.

5. Functional Requirements

This deliverable will describe functional requirement of the software. In this deliverable use cases, will be identified and explained in fully dressed format.

6. Database requirements

This deliverable will be analysis of database requirements of the system. This will explain, what database will be used? Which entities will be stored in database? Which tables will be created in database?

7. SOFTWARE DESIGN DOCUMENTATION

This deliverable will be analysis of system design including system architectural design, design class diagram, sequence diagrams, sequence diagrams, component diagram and user interfaces.

8. Software test documentation

This document deliverable will be the software testing documentation. It will include software testing approaches, test cases and test logs.

9. Admin module implementation

This deliverable will be the implementation of back end module of system.

10. client/faculty module implementation

This deliverable will be the implementation of client side of system from where users can search notifications.

1.2 Project Organization

Project organization involves several steps like choosing software process model, tools and techniques that will be used for the project development.

1.2.1 Software Process Model

Software process models are used to develop software in a systematic way. There are lot of process models available. Different software models are used for different scenarios. Examples of software process models are Water Fall model, Prototyping, Incremental Development, Spiral Development etc. The approach used for development of this system is Agile Software Development.

What is Agile Software Development?

Agility is the ability to create and respond to change to succeed in an uncertain and turbulent environment. Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto.

Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context. [1]

Reason to choose Agile Software Development

Agile Software Development has a lot of advantages. It is iterative approach mean features are delivered incrementally and product continuous to be developed. Agile software development is used to develop software quickly. This software development technique involves active user throughout the development. This technique embrace changes leading to development of right product.

Chosen Agile Process Model

There are different agile process models are available.

- Extreme Programming (XP)
- Adaptive Software Development (ASD)

- Dynamic Systems Development Method (DSDM)
- Scrum
- Crystal
- Feature Driven Development (FDD)
- Agile Modeling (AM)

Each of these models has its own advantages based on specific scenarios. Agile process model used for development of this system is Extreme Programming(XP). XP relies on object-oriented approach. Key activities involved are Planning, Design, Coding and Testing.

1.2.2 Tools and Techniques

1. Tools used for development of Document

Tool used for the documents production is Microsoft Word 2016.

2. Tools used for creating diagrams and figures

Tool that will be used for creating UML Diagrams is ArgoUML, Other recommended tools are Pencil Tool, Microsoft Visio.

3. Tools used for development of Project Plan

Tool that will be used for development of Project Plan is ProjectLibre.

4. Languages used for software development

Languages used for the software development will be

- **HTML5 (**Markup Language**)**
- **CSS (**Cascading Style Sheets**)**
- **JavaScript and JQuery**
- **PHP**
- **MySQL**

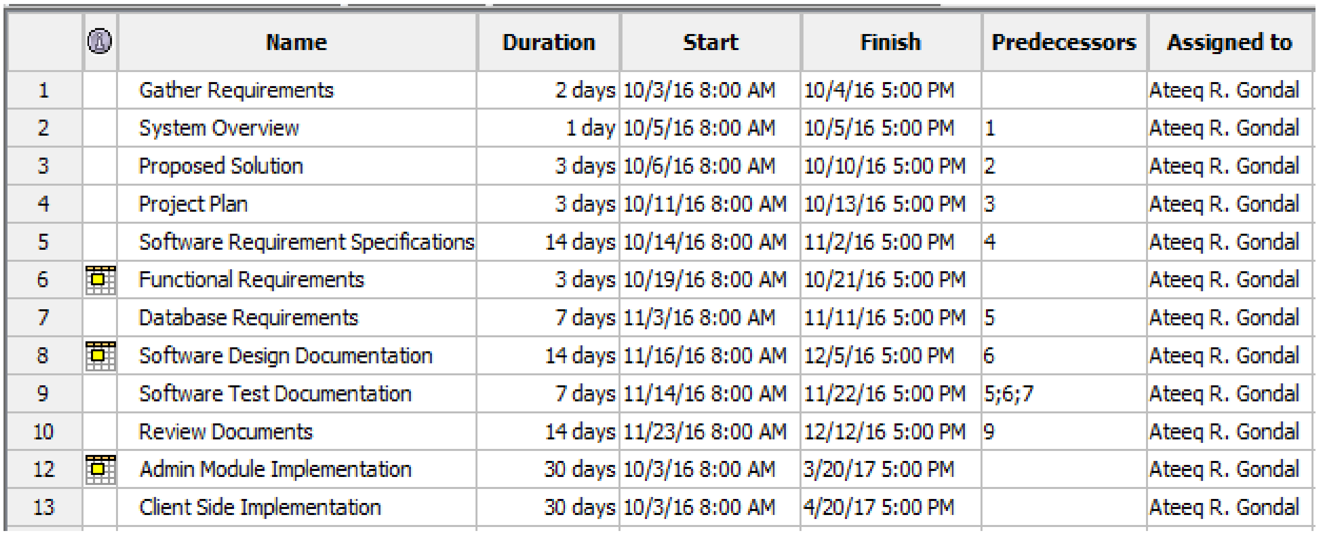
PHP MVC Framework

MVC Framework that will be used for development will be Laravel Framework.

1.3 Project Management Plan

Software project management is an umbrella activity within software engineering. It begins before any technical activity is initiated and continues throughout the modeling, construction, and deployment of computer software. [2]

Project plan for this project is constructed using project libre. Below are the tasks being their predecessor and time allocation for each task.



	ⓘ	Name	Duration	Start	Finish	Predecessors	Assigned to
1		Gather Requirements	2 days	10/3/16 8:00 AM	10/4/16 5:00 PM		Ateeq R. Gondal
2		System Overview	1 day	10/5/16 8:00 AM	10/5/16 5:00 PM	1	Ateeq R. Gondal
3		Proposed Solution	3 days	10/6/16 8:00 AM	10/10/16 5:00 PM	2	Ateeq R. Gondal
4		Project Plan	3 days	10/11/16 8:00 AM	10/13/16 5:00 PM	3	Ateeq R. Gondal
5		Software Requirement Specifications	14 days	10/14/16 8:00 AM	11/2/16 5:00 PM	4	Ateeq R. Gondal
6	▦	Functional Requirements	3 days	10/19/16 8:00 AM	10/21/16 5:00 PM		Ateeq R. Gondal
7		Database Requirements	7 days	11/3/16 8:00 AM	11/11/16 5:00 PM	5	Ateeq R. Gondal
8	▦	Software Design Documentation	14 days	11/16/16 8:00 AM	12/5/16 5:00 PM	6	Ateeq R. Gondal
9		Software Test Documentation	7 days	11/14/16 8:00 AM	11/22/16 5:00 PM	5;6;7	Ateeq R. Gondal
10		Review Documents	14 days	11/23/16 8:00 AM	12/12/16 5:00 PM	9	Ateeq R. Gondal
12	▦	Admin Module Implementation	30 days	10/3/16 8:00 AM	3/20/17 5:00 PM		Ateeq R. Gondal
13		Client Side Implementation	30 days	10/3/16 8:00 AM	4/20/17 5:00 PM		Ateeq R. Gondal

Figure 1. 1 Project Plan

1.3.1 Gant Chart

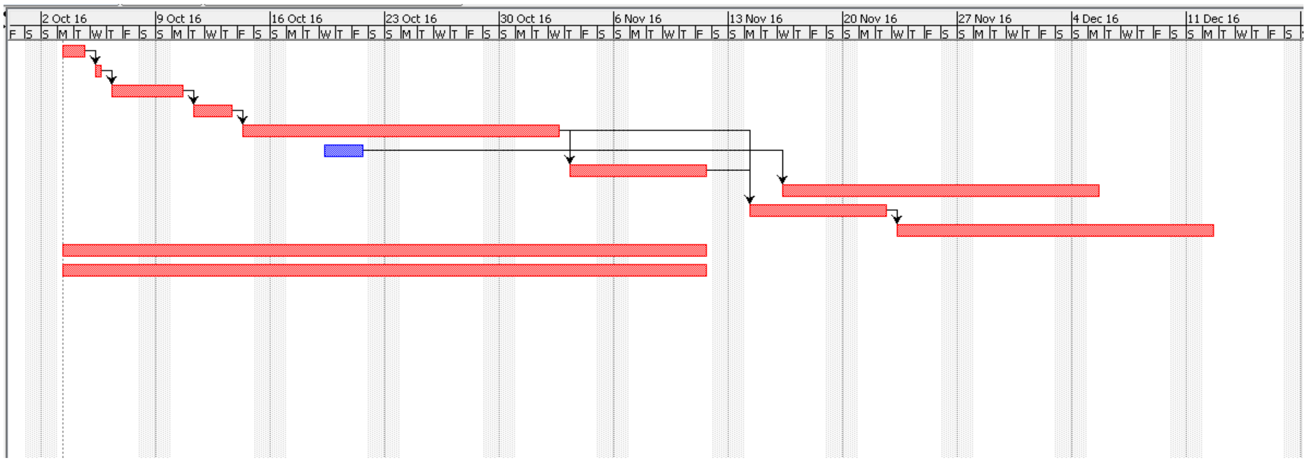


Figure 1. 2 Gant Chart

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATIONS

2 SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 Introduction

In this chapter, software requirements will be discussed. Project overview, proposed system, motivation behind this system, related work etc. will be explained.

2.1.1 Archives

Archive means a collection of records of or about an institution, family, etc. [3]. There are many varieties of archives like colleges and universities archives, corporate archives, government archives, historical archives and religious archives.

- **College and University Archives**

These types of archives contain institutional data, student's data, faculty data, libraries, research etc.

- **Corporate Archives**

Corporate archives are archival departments within a company, organization or corporation that manage and preserves the records of that business. These repositories exist to serve the needs of company staff members and to advance business goals. Corporate archives allow varying degrees of public access to their materials depending on the company's policies and archival staff availability.

- **Historical Societies**

Historical societies are organizations that seek to preserve and promote interest in the history of a region, a historical period, nongovernment organizations, or a subject. The collections of historical societies typically focus on a state or a community, and may be in charge of maintaining some governmental records as well.

Examples: The Wisconsin Historical Society, the National Railway Historical Society, the San Fernando Valley Historical Society.

- **Museums**

Museums and archives share the goal of preserving items of historical significance, but museums tend to have a greater emphasis on exhibiting those items, and maintaining diverse collections of [i]artifacts or artwork rather than books and papers. Any of the types of repositories mentioned in this list may incorporate a museum, or museums may be stand-alone institutions. Likewise, stand-alone museums may contain libraries and or archives.

Examples: The Metropolitan Museum of Art, Smithsonian National Air and Space Museum.

- **Religious Archives**

Religious archives are archives relating to the traditions or institutions of a major faith, denominations within a faith, or individual places of worship. [4]

Reference Link: **http://www2.archivists.org/sites/all/files/UsingArchives_Final.pdf**

2.1.2 Project Overview

Official Notifications Archive is a web based archive to manage official notifications. Official notifications in the department of Compute Science will be archived. Archive will provide basic search facilities. Notifications will be fed as image. The notification will be tagged under different categories like Faculty, Students, Scholarships etc. System will search the notifications based on these tags, categories and text in the notification. Archive will provide a simple user interface. The system is designed for the department of computer science, Quaid-i-Azam University Islamabad. It will facilitate department office and faculty members.

2.2 Specific Requirements

2.2.1 Proposed System

In the department of computer science, notifications are tagged and stored in paper based record files of different categories. Different categories are Scholarships, Students, Faculty, Faculty appointments and General Notifications etc. Searching of notifications from these record is quite a difficult task. Getting notifications from the traditional paper based file records is time consuming thing. These operations are purely paper based. The proposed solution for managing notifications in an advanced way is to make a web based archive. Notifications will be archived. Archive will provide basic search facilities. Notifications will be tagged under different categories. For example, there is a notification related to HEC merit based scholarships. This notification will be added to system in Scholarship category. Tags that will be added along with this notification will be "HEC Scholarship", "Merit based scholarship". To make this archive, initially notification will be collected, converted to images, annotation of images and then uploading images to the archive. There are two modes to access information. One is push mode and the other is pull mode. Notifications will be accessed through the pull mode of access. Pull mode includes querying the data and browsing the data. Once notifications are added to the archive, they will be searchable by categories, tags, date and title.

1 Collection of Notifications

In department of Computer Science, official notifications are stored in different places in the form of paper based file records. To make archive of these notifications, first notifications will be collected from the office. They will be scanned to convert them to image. These images will be stored in the archive.

2 Feeding as an image

After collecting notifications, notifications will be converted to images. These images will be fed as images under different categories and will be tagged. Tags can be manually defined by the administrator. These tags will be used to search notifications from the archive.

3 Browsing, Searching of Notifications

When the notifications will be fed, they can be browsed by different categories. Basic search operations can be performed on these notifications. They can be searched by date, categories, tags and title.

2.2.2 Motivation

This is era of technology. Everything is being automated. Traditional paper based file records are being converted to online archives. These archives provide a lot of functionality like searching of documents. For example, MIT, Digital Archive is used in MIT to archive official documents. Online archives are a great way to get information when needed in no time. In the Department of Computer Science, Quaid-i-Azam University Islamabad, no such online archive exists. The motivation behind this system is to facilitate Department of Computer Science in archiving official notifications. This system is designed for the Department of Computer Sciences, QAU Islamabad. In traditional paper files, searching of notifications is a challenging and time-consuming thing. This system will enable computer science department to search notifications in matter of seconds. This project will be a step toward advancement.

2.2.3 Related Work

Archives come in all shapes and sizes. There are national and international archives, state archives, city archives, community archives, business archives and many more. There are archives for different types of government records and archives that contain the personal records of people and organizations. College and university archives are archives that preserve materials relating to a specific academic institution. Such archives may also contain a special collection. Special collections are institutions containing materials from individuals, families, and organizations deemed to have significant historical value. Topics collected in special collections vary widely, and include medicine, law, literature, fine art, and technology. Often a special collections repository will be a department within a library, holding the library's rarest or most valuable original manuscripts, books, and/or collections of local history for neighboring communities. College and university archives exist first to serve their parent institutions and alumni, and then to serve the public.

Examples

Below are few examples of archives.

- **Pakistan Research Repository**

Pakistan Research Repository is a project of the Higher Education Commission to promote the international visibility of research originating out of institutes of higher education in Pakistan. The aim of this service is to maintain a digital archive of all PhD thesis produced indigenously to promote the intellectual output of Pakistani institutions. It provides a free, single-entry access point to view the manuscript of research executed, and distribute this information as widely as possible. The repository which is currently being populated with content has already made the full-text of PhD theses available in high-quality digitized format, whilst a further thesis is in process of digitization. Higher Education Commission has introduced a systematic mechanism for the collection and digitization of all the theses produced so far in Pakistan.

Reference Link: eprints.hec.gov.pk/

- **Human Rights Commission of Pakistan (HRCP) archive**

Human Rights Commission of Pakistan (HRCP) has launched an online searchable archive (www.hrcparchive.com), which contains newspaper clippings, documents and data pertaining to human rights issues and democratic development in Pakistan. Each year hundreds of students, journalists, writers, civil society activists and researchers from within Pakistan and abroad visit the HRCP's archives at its head office in Lahore. This online archive made it possible to access the required research material in an easier, faster and more efficient way.

Reference Link: http://www.hrcparchive.com/search

- **Stanford University Archives**

Stanford university's archives are used to get different types of information.

Reference Link: http://library.stanford.edu/

- **Harvard University Archives**

The Harvard University Archives is the oldest and largest academic archives in the United States. It collects, preserves, and provides access to a comprehensive record of academic, administrative, student, and social life at Harvard. From 17th century deeds to 21st century web sites, the collections in the Archives comprise over 51,000 feet of University records and publications, personal and faculty archives, and related historical materials that include paper correspondence, minutes and reports, photographs, film, audio and video recordings, and electronic files. The collections support research by scholars of social, intellectual, and academic history; by historians of Harvard, including University academic and administrative departments requiring information from their own early records; by students learning the methodology of historical research and writing; and by the public.

Reference Link: http://library.harvard.edu/

In Pakistani universities, no such archives exist. Official Notifications Archive is designed for Department of Compute Science, Quaid-i-Azam University Islamabad to archive official notifications. This archive will be very helpful in maintaining official notifications.

2.2.4 Purpose

Paper based file records are traditional way to store notifications, documents etc. These file records are of different categories like a file record for student's related documents, file record for faculty related documents, file record for scholarships documents. These file records are maintained to keep a record of the documents. Whenever someone needed the document, they can get the document from the file record. But this thing is time consuming. Official Notifications Archive is solution to this problem. The purpose of the project is to store these documents online in the same way as they are stored in file records in the form of an archive. This archive will help to search notifications easily. This will be a web based archive of official notifications. This will make searching, viewing of official notifications a lot easier than traditional paper files.

2.2.5 Scope

Official Notifications Archive is a web based archive to store official notifications. First, official notifications will be collected. They will be converted to image format to feed in the archive. Administrator must have account to use the system. Administrator can add notifications as images in the archive. Notifications will be tagged under different categories defined by the user. The faculty members should create an account on the system to access notifications. They can perform basic search operations on the notifications. They can download the notifications.

2.2.6 Interfaces

Interfaces for the system includes software interfaces, hardware interfaces, user interfaces and communication interfaces.

1 Software Interfaces

Since the application is web based. It can be easily viewed on desktop computers, laptops and smartphones through web browsers. The required software to run the application are:

- Any updated web browser meeting the today's web technology requirements.

2 Hardware Interfaces

The hardware interfaces required to run the application are following. The application can be run on any on the following.

- Desktop computers
- Laptops
- Smart Phones

3 User Interfaces

User can interact with the application through mouse, or if his device has touch functionality then the user can interact using touch gestures.

4 Communication Interfaces

Communication interface for system is web interface.

2.2.7 Software Product Features

Software has two types of users. One is administrator who will add notifications to the archive etc. and the others are Faculty Members. Faculty members will only search and browse the archive.

Administrator Functionality

1. Sign Up
2. Login
3. Add Category
4. Add Notification
5. Update Notification
6. Delete Notification
7. Search Notification
8. Browse by Category
9. View Notification
10. Add Personal Notification

Functionality for faculty

1. Sign up
2. Login
3. Search notification
4. Browse by Category
5. View Notification
6. Download Notification
7. Upload My Notification

2.2.8 Use Cases Description

Use cases description is given below.

Use Cases

Below are the identified use cases for the whole system.

1. Sign Up
2. Login
3. Add Category
4. Update Category
5. Add Notification
6. Update Notification
7. Delete Notification
8. Search Notification
9. Browse by Category
10. View Notification
11. Download Notification
12. Add Personal Notification
13. Upload My Notification

Use Cases Description

1. Sign Up

The admin first must sign up to use the system.

Use Case ID	UC-001
Use Case Name	Sign Up
Primary Actor	Admin, Faculty Member
Pre-Condition	The user has accessed the website.
Post-Condition	The user has successfully signed up.
Main Success Scenarios	1. The System prompts the user for a username and password or register new account. 2. The user selects registration option. 3. The System prompts user for registration information, Username, password, etc. 4. The user enters information. 5. System verifies information and creates account.
Alternative Flows	3a. The page is not loaded successfully. 4a. The user enters incorrect information.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever there will be a new user.

Table 2. 1 Sign Up Use Case description

2. Sign in

The admin first must sign in to use the system.

Use Case ID	UC-002
Use Case Name	Sign-in
Primary Actor	Admin, Faculty Member
Pre-Condition	The user has accessed the website and has account on the system.
Post-Condition	The user has successfully signed in.
Main Success Scenarios	1. The System prompts the user for a username and password. 2. The user enters username and password. 3. The user clicks sign in button. 4. System verifies information. 5. User is signed in.
Alternative Flows	2a. The user enters wrong input. 4a. System does not verify information.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever the session expires, user should login.

Table 2. 2 Sign in Use Case description

3. Add Category

This use case will be used to define categories in the system under which notifications will be stored.

Use Case ID	UC-003
Use Case Name	Add Category
Primary Actor	Admin
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully created the category.
Main Success Scenarios	1. The user clicks Add Category Button. 2. The user enters category name. 3. The user clicks on save category button. 4. System saves category.
Alternative Flows	2a. The user enters incorrect category name. 3a. User does not save category.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Every time user need a new category for notifications.

Table 2. 3 Add Category Use Case description

4. Modify Category

This use case will be used to modify the name of the category.

Use Case ID	UC-004
Use Case Name	Modify Category
Primary Actor	Admin
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully created the category.
Main Success Scenarios	1. The user clicks Modify Category Button. 2. The user enters new category name. 3. The user clicks on save category button. 4. System saves category. 5. The modified category is stored in the database.
Alternative Flows	2a. The user enters incorrect category name. 3a. User does not save category.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Each time user wants to change name for category

Table 2. 4 Modify Category Use Case description

5. Add Notification

This use case will be used to add the notifications in the system.

Use Case ID	UC-005
Use Case Name	Add Notification
Primary Actor	Admin
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully added notification.
Main Success Scenarios	1. The user clicks Add Notification Button. 2. A new window is appeared asking for the notification details. 3. User clicks on upload image button. 4. If the notification consists of more than one images, user clicks on add more image button. 5. The user defines tags for the notification and select category from the list. 6. The user clicks save notification button to save the notification. 7. If the user does not want to save the notification, he will click cancel button. 8. By clicking save notification button, the notification will be saved in the database.
Alternative Flows	3a. User upload file that is not supported.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever there is a new notification to add.

Table 2. 5 Add Notification Use Case description

6. Update Notification

This use case will be used to add the notifications in the system.

Use Case ID	UC-006
Use Case Name	Update Notification
Primary Actor	Admin
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully updated the notification.
Main Success Scenarios	1. The user searches the notification which he wanted to update. 2. User opens the notification. 3. User clicks on update notification button. 4. User enters new details of the notification. 5. The user clicks save notification button to save the notification. 6. If the user does not want to save the notification, he will click cancel button. 7. By clicking save notification button, the notification will be updated in the database.
Alternative Flows	4a. User enters incorrect data.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever user wants to update notification.

Table 2. 6 Update Notification Use Case description

7. Delete Notification

Delete notification use case will be used to delete any notification from the archive.

Use Case ID	UC-007
Use Case Name	Delete Notification
Primary Actor	Admin
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully deleted the notification.
Main Success Scenarios	1. The user searches the notification which he wanted to delete. 2. User opens the notification. 3. User clicks on delete notification button. 4. User confirms action. 5. Notification is deleted from the archive.
Alternative Flows	2a. User is not able to open notification. 4a. User cancel the action.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever user wants to delete notification.

Table 2. 7 Delete Notification Use Case description

8. Search Notification

This use case will be used when a search will be made on notifications.

Use Case ID	UC-008
Use Case Name	Search Notification
Primary Actor	Admin, Faculty Member
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully searched the notification.
Main Success Scenarios	1. The user type search keywords in search box. 2. One or more than one results displayed. 3. User clicks notifications to open. 4. User successfully searched notification.
Alternative Flows	1a. User types incorrect keywords or characters. 2a. No search result. 3a. Notification is not opened. 4a. User is not able to search notifications.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever the user want to perform search, the use case will be executed.

Table 2. 8 Search Notifications Use Case description

9. Browse by Category

This use case will be used to browse all notifications by categories.

Use Case ID	UC-009
Use Case Name	Browse by Category
Primary Actor	Admin, Faculty Member
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully browsed notification.
Main Success Scenarios	1. The user clicks categories. 2. The user browses all notifications. 3. User clicks notifications to open. 4. User successfully browsed notification.
Alternative Flows	2a. No notification to browse.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever user needs to browse categories.

Table 2. 9 Browse by Category Use Case description

10. View Notification

This use case will be used to view notifications.

Use Case ID	UC-010
Use Case Name	View Notification
Primary Actor	Admin, Faculty Member
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully viewed notification.
Main Success Scenarios	1. The user search or browse the notification. 2. User clicks on notification. 3. Notification is opened. 4. User successfully viewed notification.
Alternative Flows	1a. No notification is found. 2a. Notification is not opened.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever user wants to view notification.

Table 2. 10 View Notification use case description

11. Download Notification

This use case will be used to save the notifications from the archive.

Use Case ID	UC-011
Use Case Name	Download Notification
Primary Actor	Admin, Faculty Member
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	Notification is saved on user’s machine.
Main Success Scenarios	1. The user search or browse the notification. 2. User clicks on notification. 3. User clicks download notification button. 4. Notification will be downloaded.
Alternative Flows	1a. No notification is found.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever user needs to download notification.

Table 2. 11 Download Notification Use Case description

12. Add Personal Notification

This use case will be used to send email to the user.

Use Case ID	UC-012
Use Case Name	Add Personal Notification
Primary Actor	Admin
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	Notification is added for some specific user.
Main Success Scenarios	1. User clicks on Add Notification. 2. User fill in the form. 3. User clicks Add Notification. 4. Notification is added.
Alternative Flows	4a. Notification is not added.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever user needs to download notification.

Table 2. 5 Download Notification Use Case description

13. Add Personal Notification

This use case will be used to add the notifications in the system.

Use Case ID	UC-013
Use Case Name	Add Personal Notification
Primary Actor	Faculty Member
Pre-Condition	The user has accessed the website and has signed in on the system.
Post-Condition	The user has successfully added notification.
Main Success Scenarios	1. The user clicks Add Personal Notification Button. 2. A new window is appeared asking for the notification details. 3. User clicks on upload image button. 4. If the notification consists of more than one images, user clicks on add more image button. 5. The user clicks save notification button to save the notification. 6. If the user does not want to save the notification, he will click cancel button. 7. By clicking save notification button, the notification will be saved in the database.
Alternative Flows	3a. User upload file that is not supported.
Special Requirements	No special requirements are required other than a web browser and internet.
Frequency of occurrence	Whenever there is a new notification to add.

Table 2. 13 Add Personal Notification Use Case description

2.2.9 Use Cases Diagram

Use-Case diagram help to determine the functionality and features of the software from the user's perspective. Use case diagram of Official Notifications Archive is given below.

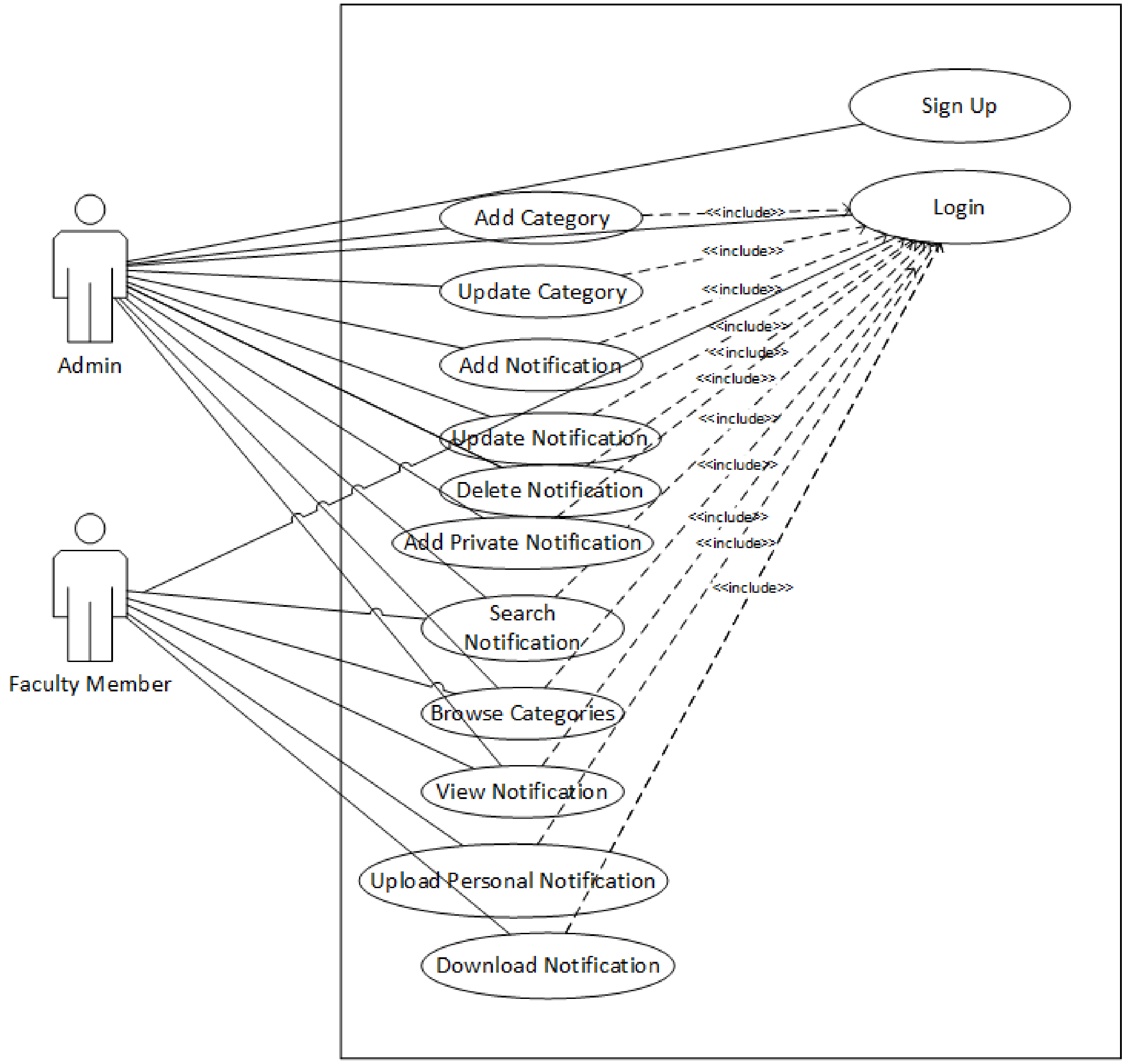


Figure 2. 1 Use Case diagram

2.2.10 Software System Attributes

Reliability

Reliability of system is the ability of a computer program to perform its intended functions and operations in a system's environment, without experiencing failure (system crash). This system should perform all operations without failure. It should give proper search results with precision more than 90 %.

Availability

The system must be available for use when it is needed. The web site should have 99.9 % uptime.

Security

System should be secure. The system should not be used by any unauthorized user.

- System should not allow any fake sign up, login
- Secure database
- No bogus entries to database
- No unauthorized changes in data

Maintainability

The applications will be developed in such a way that it will facilitate maintainability.

Portability

As the system is a web based application. It can be used on smartphone and laptops as well as on desktop computers. The system will provide responsive interface.

Performance

- Application should not crash on any user action
- Application should provide responsive interfaces on all devices
- Applications should have 80 % speed.
- Page load time should be less than or equal to 2 seconds.

2.2.11 Database Requirements

The application will access the MySQL database for the following operations

- Store administrator’s data
- Store faculty member data l
- Store categories
- Store notifications as images.
- Store tags.
- Store private notifications

Entity Relationship Diagram

Entity relationship diagram for system is show below.

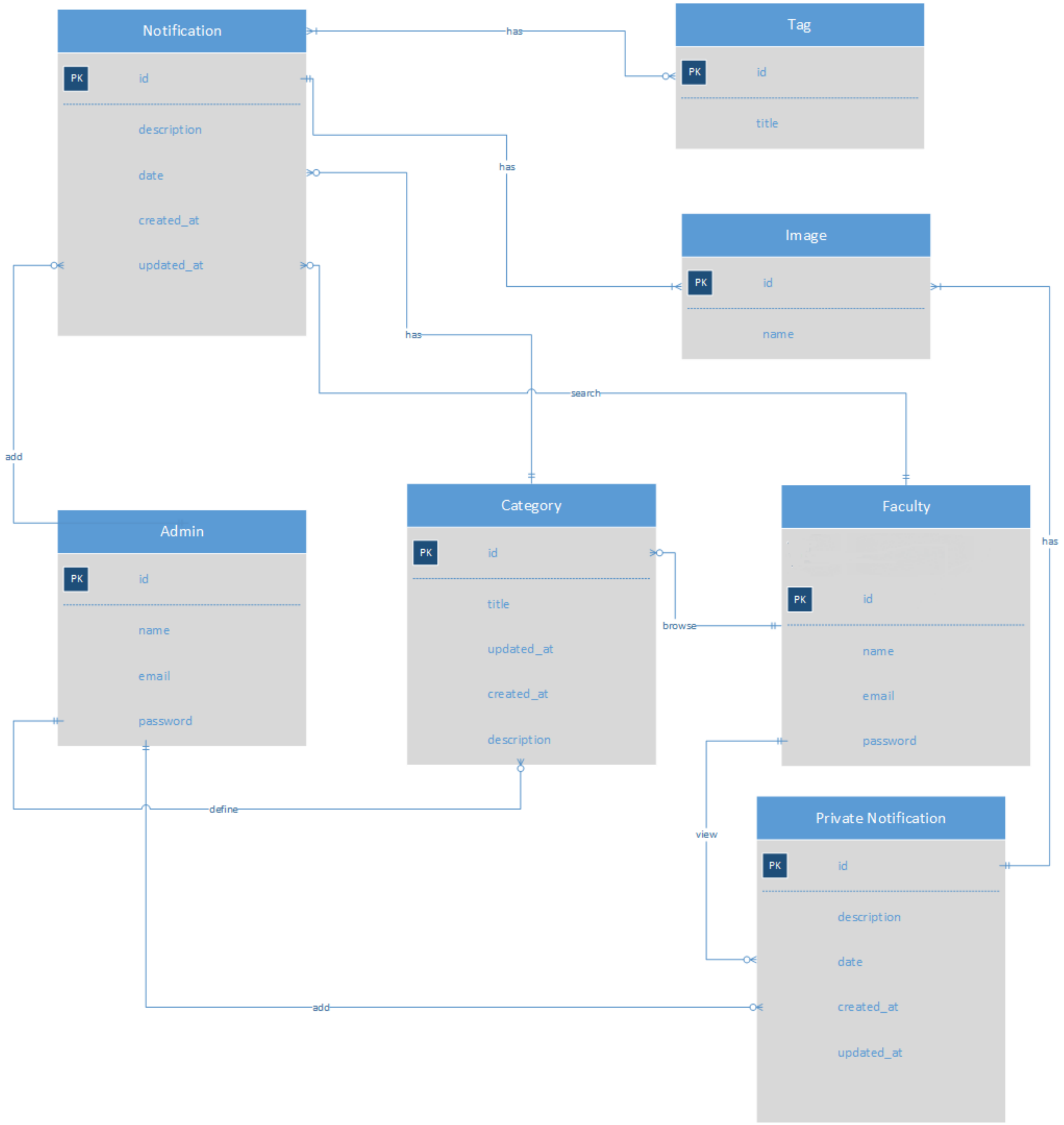


Figure 2. 2 Entity Relationship Diagram

CHAPTER 3

SOFTWARE DESIGN DESCRIPTION

3.1 Introduction

This chapter introduces Software Design in more details. In this chapter we will discuss sequence diagrams, class diagram, database design, software architecture, detailed description of components and user interfaces.

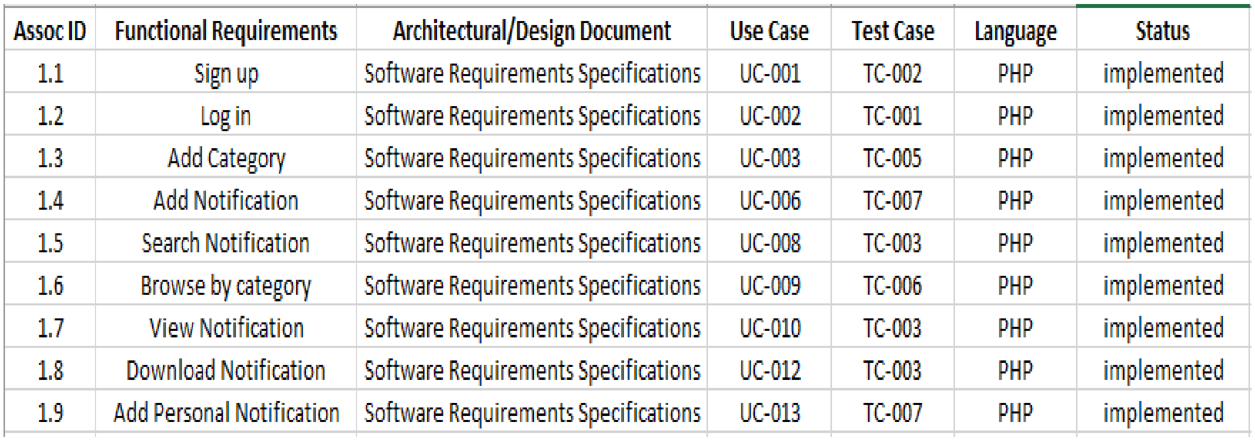
3.1.1 Design Overview

Design of system explains overall design of system. It conveys the purpose of different components and modules of system. Design of system is built in a step by step procedure. It involves software architecture, domain model, class diagram, database design and user interface design.

3.1.2 Requirements Traceability Matrix

Software requirements should always be traceable. Through this matrix each requirement can be traced throughout the development life cycle.

Below table shows traceability matrix.



Assoc ID	Functional Requirements	Architectural/Design Document	Use Case	Test Case	Language	Status
1.1	Sign up	Software Requirements Specifications	UC-001	TC-002	PHP	implemented
1.2	Log in	Software Requirements Specifications	UC-002	TC-001	PHP	implemented
1.3	Add Category	Software Requirements Specifications	UC-003	TC-005	PHP	implemented
1.4	Add Notification	Software Requirements Specifications	UC-006	TC-007	PHP	implemented
1.5	Search Notification	Software Requirements Specifications	UC-008	TC-003	PHP	implemented
1.6	Browse by category	Software Requirements Specifications	UC-009	TC-006	PHP	implemented
1.7	View Notification	Software Requirements Specifications	UC-010	TC-003	PHP	implemented
1.8	Download Notification	Software Requirements Specifications	UC-012	TC-003	PHP	implemented
1.9	Add Personal Notification	Software Requirements Specifications	UC-013	TC-007	PHP	implemented

Figure 3. 1 Traceability matrix

3.2 Software Architectural Design

The software architecture of a software specifies its basic structure. In many ways, it is design created at a high level of abstraction. Architecture deals with properties external to each module. At the architecture level we should think about the important modules and how they interact with other modules [5].

3.2.1 Chosen System Architecture

Official Notifications archive is a web based application. The modules of this application are client, server and database. Chosen architecture for this application is three-tier architecture.

Three-tier Architecture

The three-tier architecture style is often used as a model for web-based applications. The client machine access an application server through a web browser; the application server implements the business logic and communicates with the database. This kind of architecture can also be viewed as a variation to the of the MVC architecture, with the database being the model, and the application server implementing the controller and generating the views that will be displayed by the clients with a web browser.

3 SOFTWARE DESIGN DESCRIPTION

Architecture Diagram

Three-tier architecture of this system has interface tier, business logic and database. Interface tier is from where user interacts with application. Business logic is where the whole application logic works. Database tier is responsible for storing and maintaining data. System architecture diagram for this system is given below.

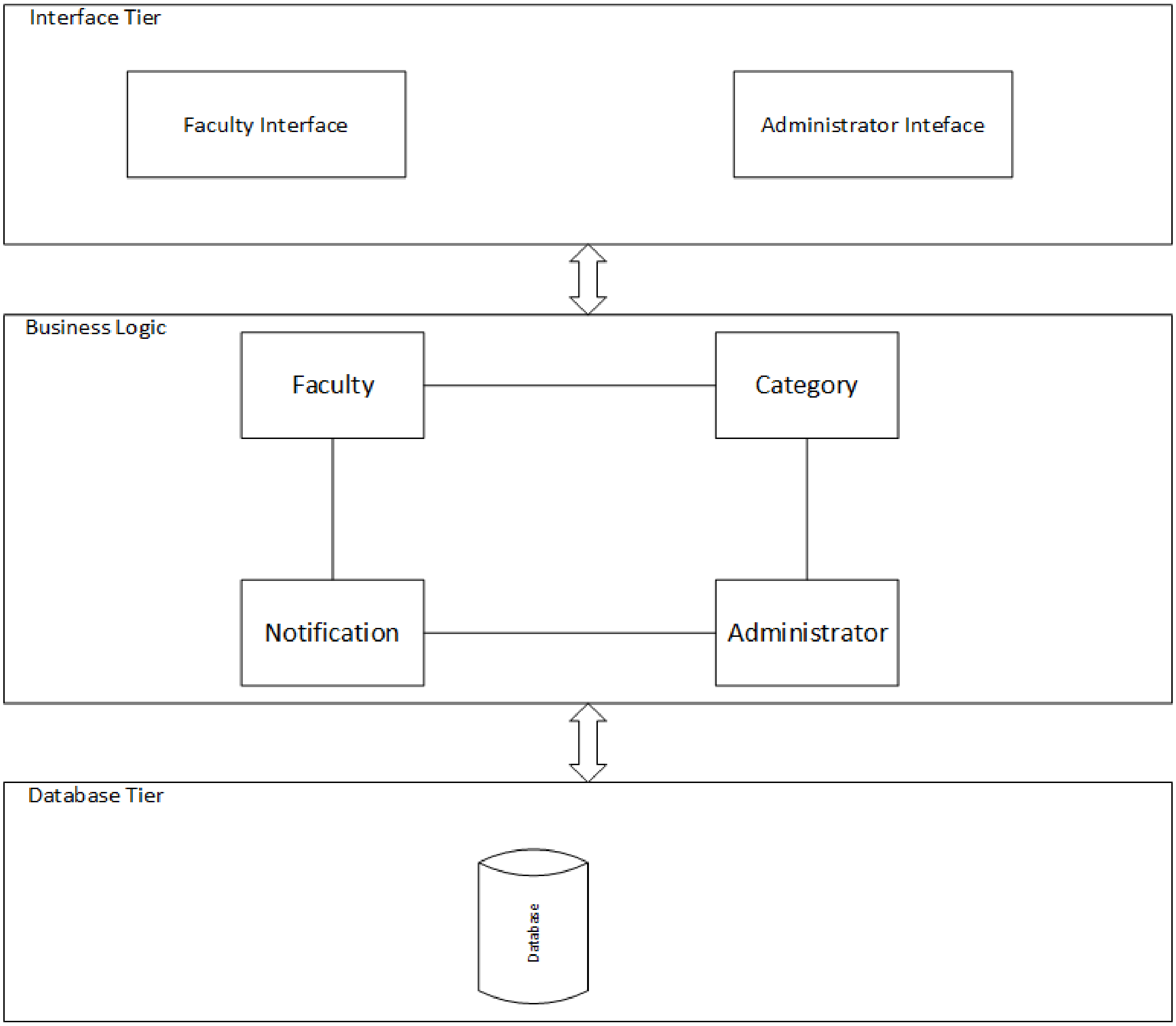


Figure 3. 2 System Architecture Diagram

3.3 Detailed Description of Components

Components of Official Notifications Archive are

1. Login Form
2. Registration Form
3. Notification Form
4. Notification Controller
5. Models
6. Registration Controller
7. Authentication Controller
8. Category Controller
9. Tag Controller

3.3.1 Login Form

Login form is used for login purpose of user.

3.3.2 Registration Form

Registration form is from where new user will be registered.

3.3.3 Notification Form

Notification form component is used to add new form to archive.

3.3.4 Notification Controller

Notification controller component is used to control the notifications. Notifications are sent to controller. Controller will validate notifications and will sent to model to store in database.

3.3.5 Models

Models component talks with database. It will store and retrieve data from database.

3.3.6 Registration Controller

Registration controller component is used to validate registration of user. After validation, it will talk with model component to store new user in database.

3.3.7 Authentication Controller

Authentication controller component is for authentication of user. When user wants to login on system, it will authenticate user and will let user use system.

3.3.8 Category Controller

Category controller component is to manage categories. If new category is created, it will be handled through category controller.

3.3.9 Tag Controller

Tag controller component is used for tags. If a valid tag is added, it will talk with model component. Tag will be stored in database against particular notification.

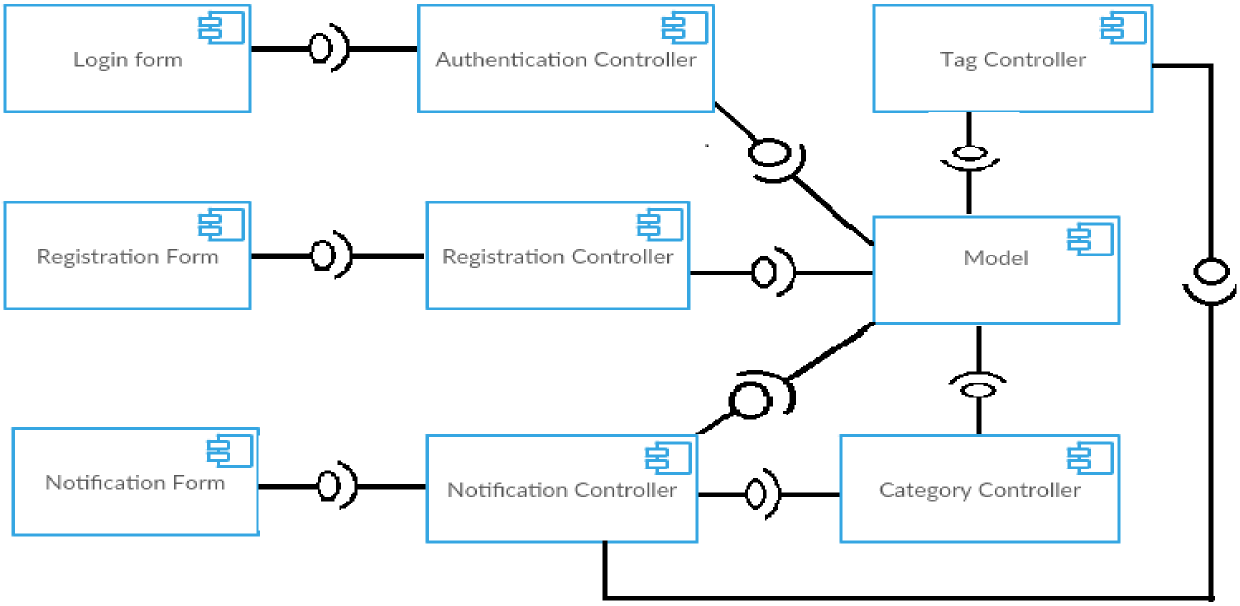


Figure 3. 3 Component diagram

3.4 Detailed Design

Detailed designed for the system includes relational database design, class design and class diagrams, interaction among classes, sequence diagrams and user interface design.

3.4.1 Domain Model

Domain Model helps to identify relevant concepts and ideas of a domain. Domain models identifies conceptual classes of the domain. These classes are bases of software design.

Conceptual Classes

Conceptual classes for the Official Notifications Archive are given below.

1. Notification
2. Category
3. Administrator
4. Faculty
5. Tag
6. Image

3 SOFTWARE DESIGN DESCRIPTION

Domain Model Diagram

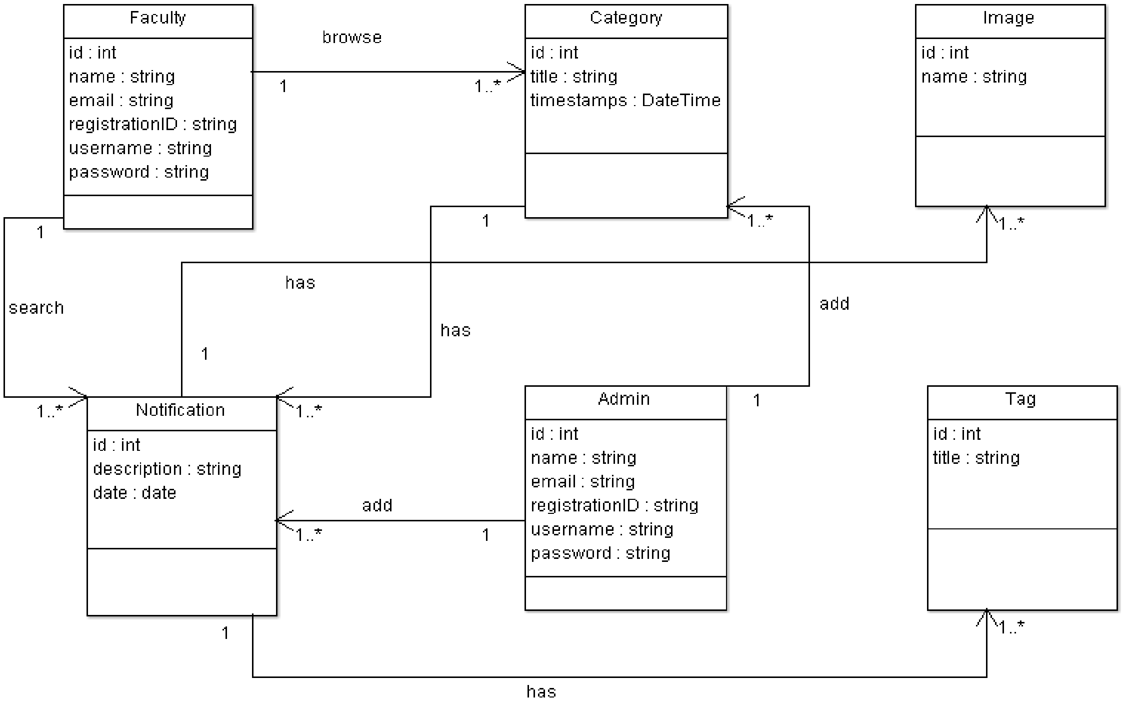


Figure 3. 4 Domain Model

3.4.2 Sequence Diagrams

1. Sign up

User will open website in his browser. There is a Create Account Button. He will click this button to initialize sign up process. He will fill form and submit this for. Registration controller class will validate and process this sign up process. After processing the data will be stored in database system through model class. Below is the sequence diagram for sign up process.

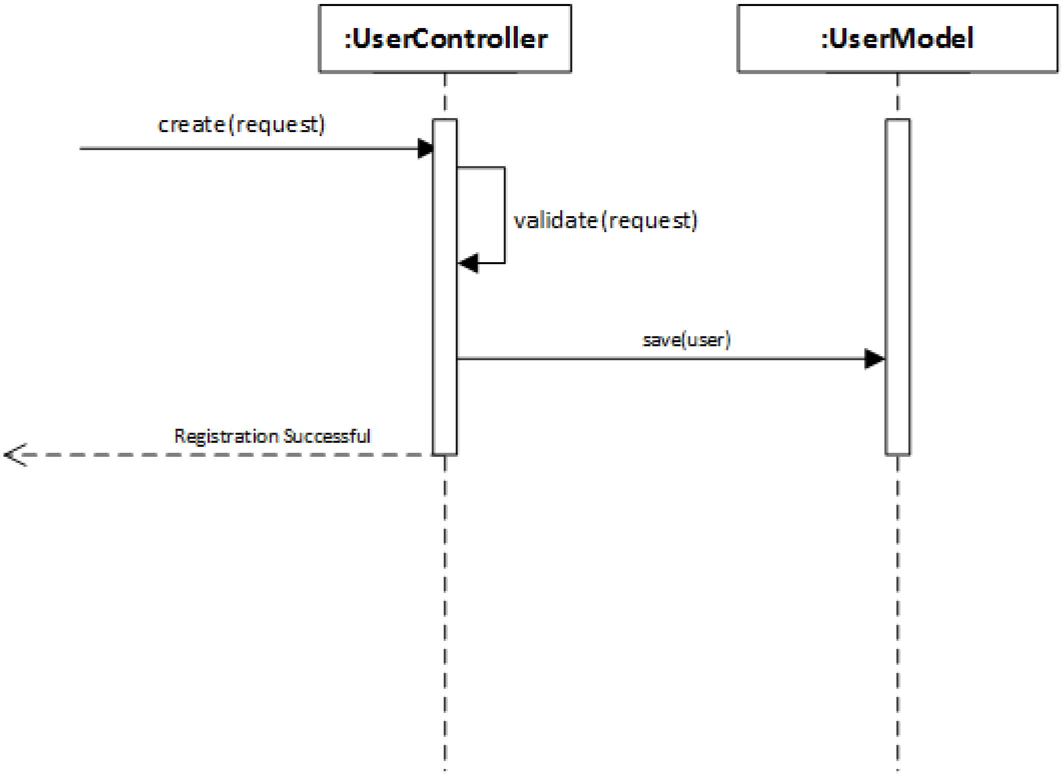


Figure 3. 5 Sign up sequence diagram

2. Log in

User has to log in to use system. He will fill his login form and submit. This form data will be processed by login controller class. After authenticating data from model class, user will be given a message either his login was successful or not.

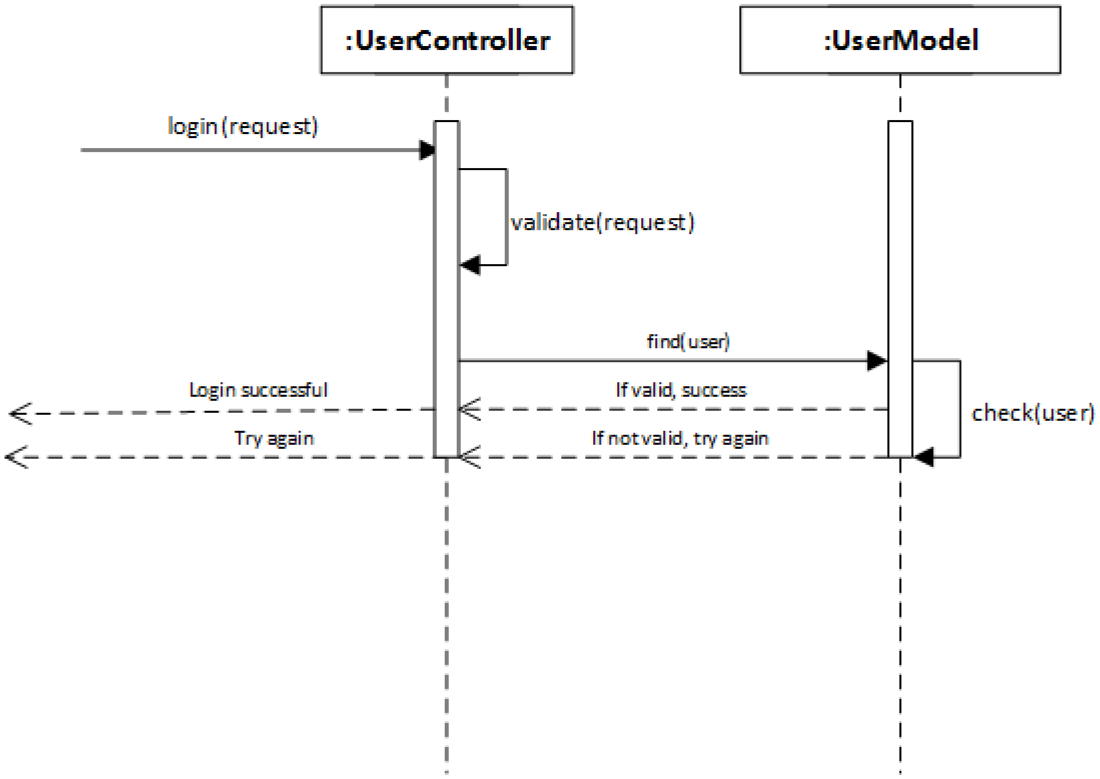


Figure 3. 6 Log in sequence diagram

3. Add Notification

After signing in to system, user can add notifications to archive. He will click New Notification Button. He will fill form for notification and upload image file. After submission notification controller class will validate this data. After validation this notification will be stored in database through model class.

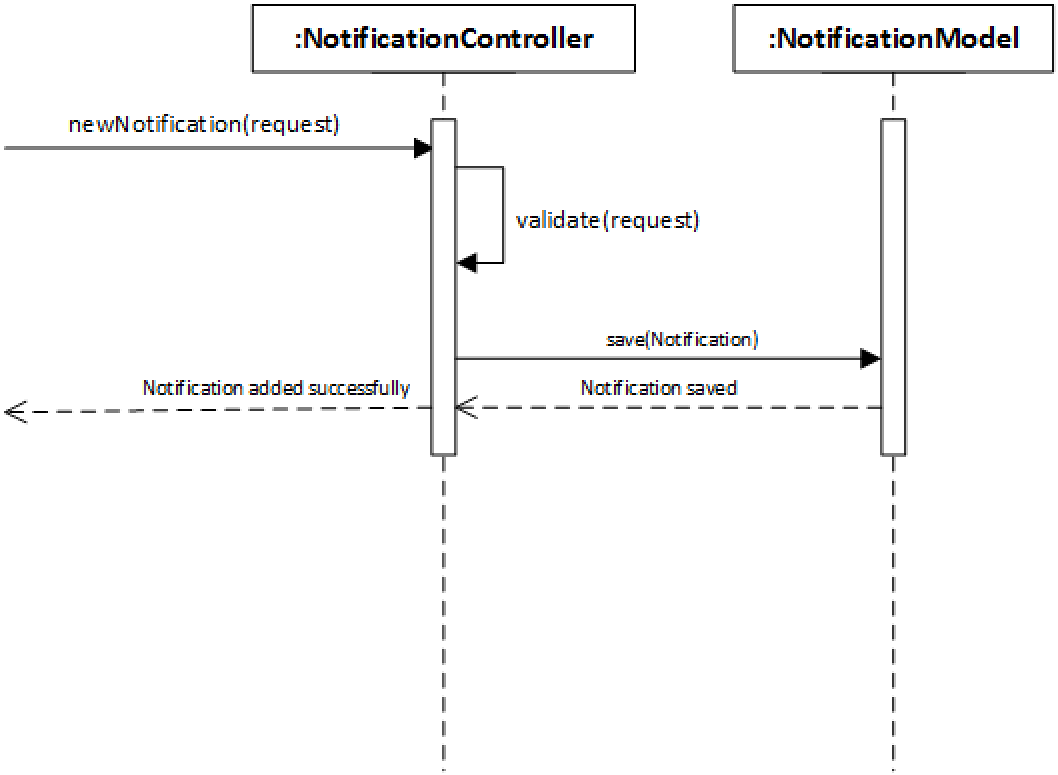


Figure 3. 7 Add Notification sequence diagram

4. Add Category

User will click on Add New Category button. He will be given a form for category. He will submit this form to category Controller class. After validation this category will be stored in database through model class.

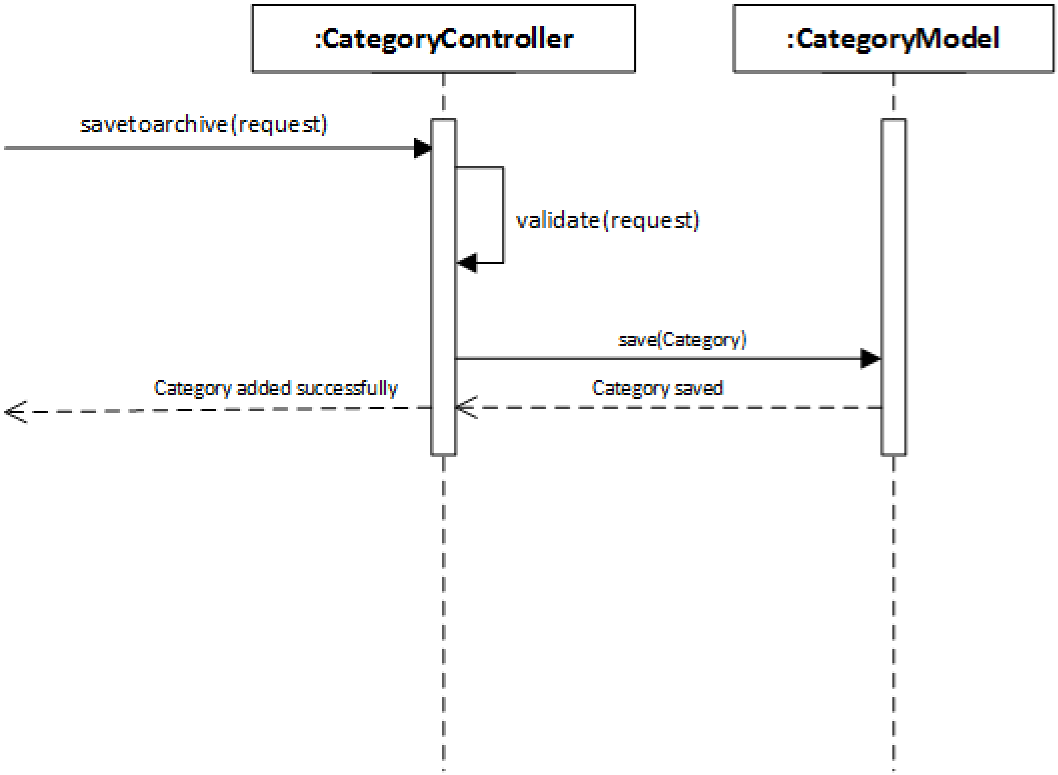


Figure 3. 8 Add Category sequence diagram

5. Browse Category

User can browse categories one by one. He will click on Browse Categories button. He will click on a specific category. This category will be sent to categories controller class. Controller class will contact model class to get data from database. Below is the sequence diagram for browse category.

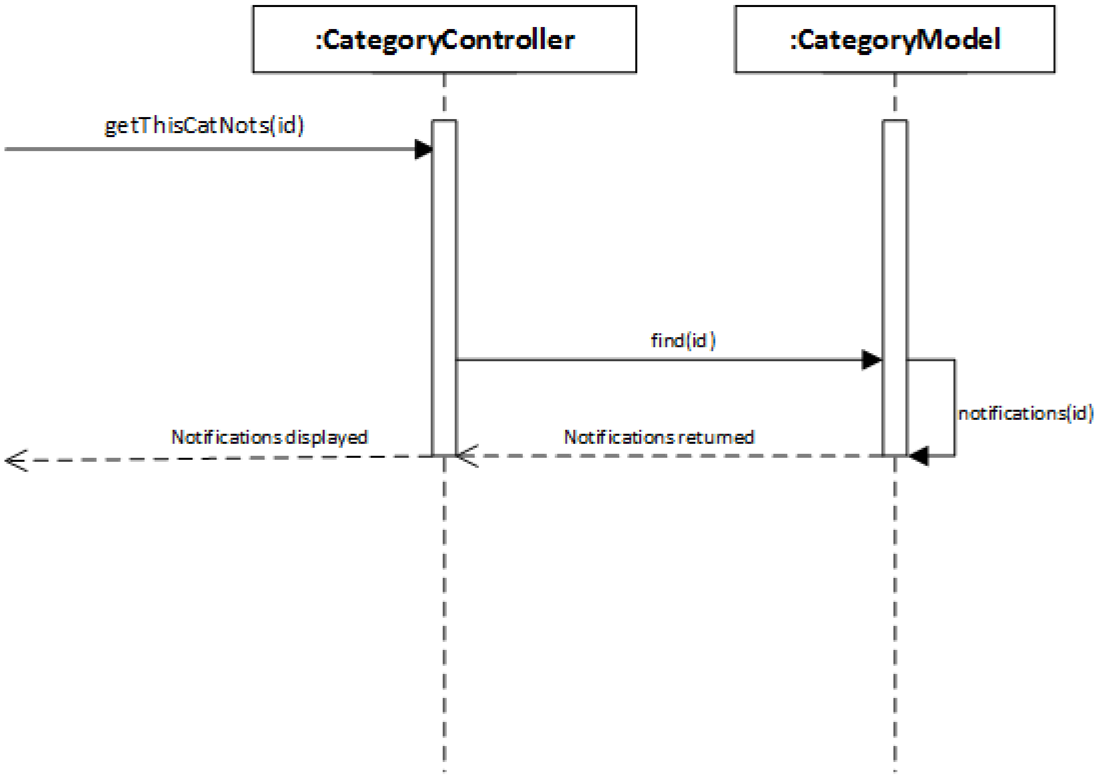


Figure 3. 9 Browse Category sequence diagram

6. Search Notification

User can search notifications. He will enter search keyword on search field. He will press search button. This keyword will be send to search controller. Controller will match search keyword against different data. The result will be given to user. Below is sequence diagram for search notifications.

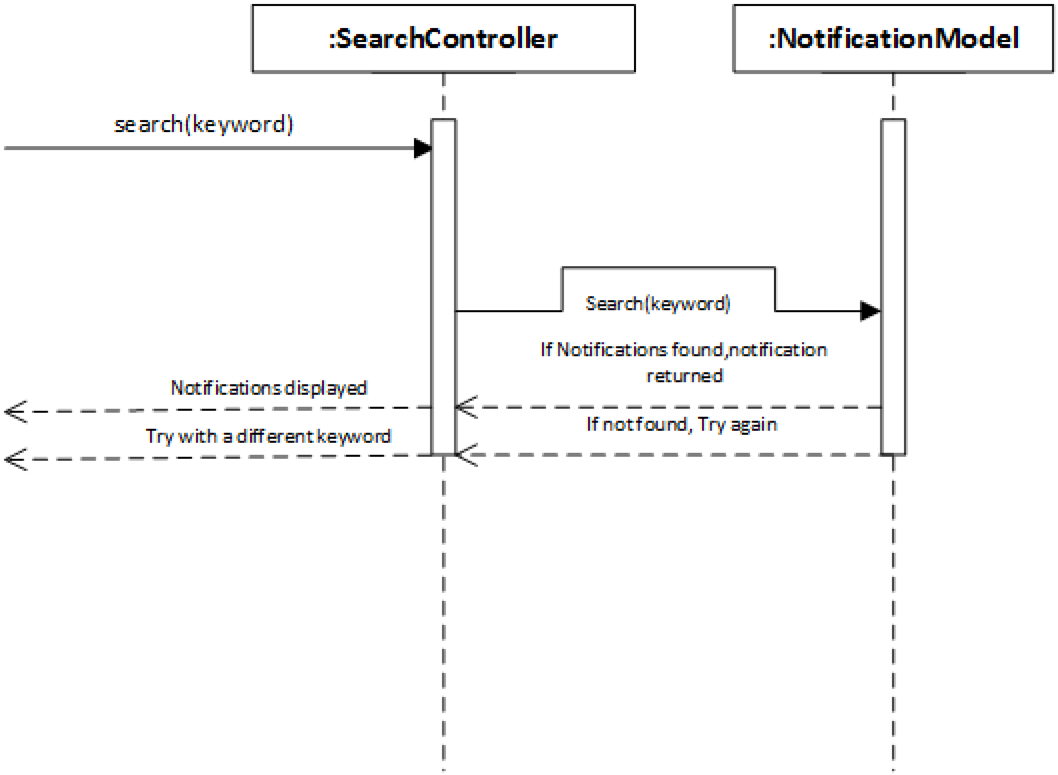


Figure 3. 10 Search Notification sequence diagram

7. View Notification

User has to click on a specific notification to view it. This notification id will be passed to notification controller class. It will return notification to user. Below is sequence diagram for viewing notification.

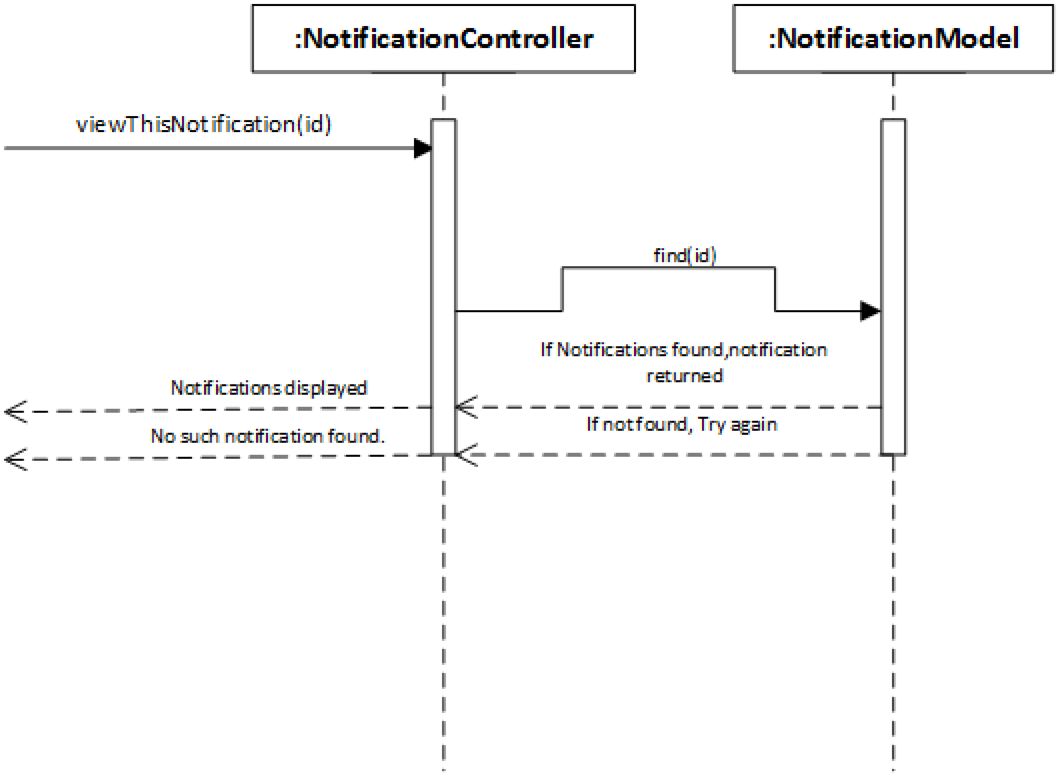


Figure 3. 11 View Notification sequence diagram

8. Delete Notification

User has to click on a specific notification. There will be a button for deletion. User will click Delete button. This notification id will be passed to controller class for deletion. Upon completion the process, user will be given success message.

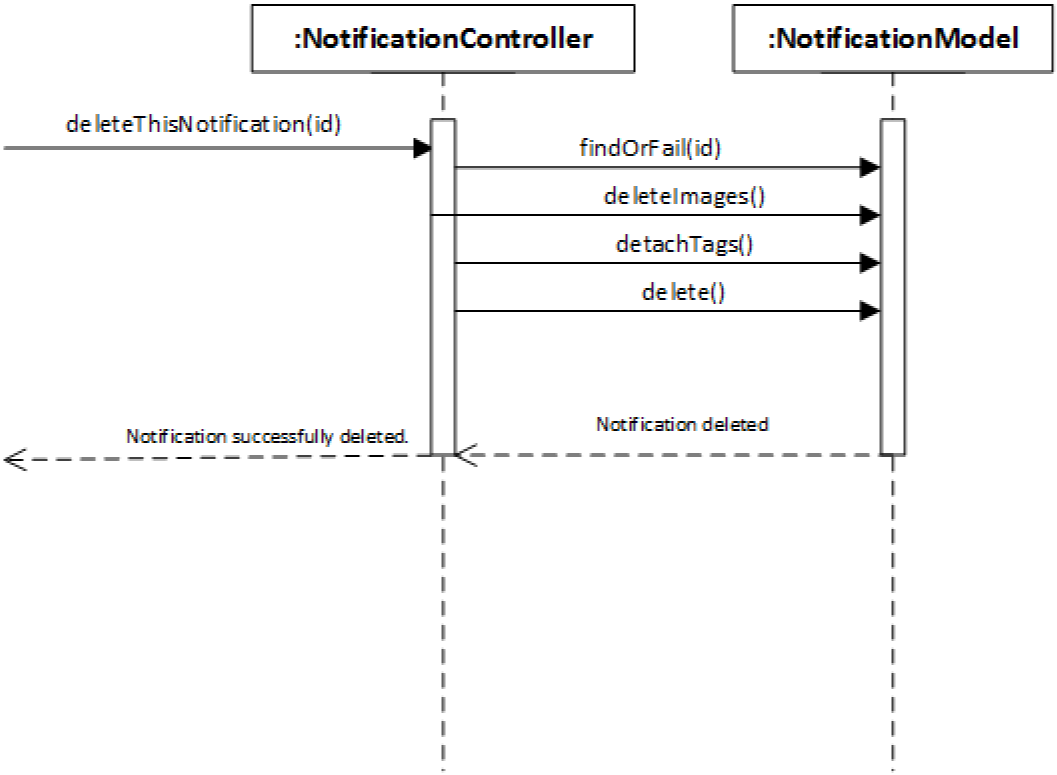


Figure 3. 12 Delete Notification sequence diagram

3.4.3 Class Diagram

Each class has private data members and public methods. Getters and Setters are used to access data members.

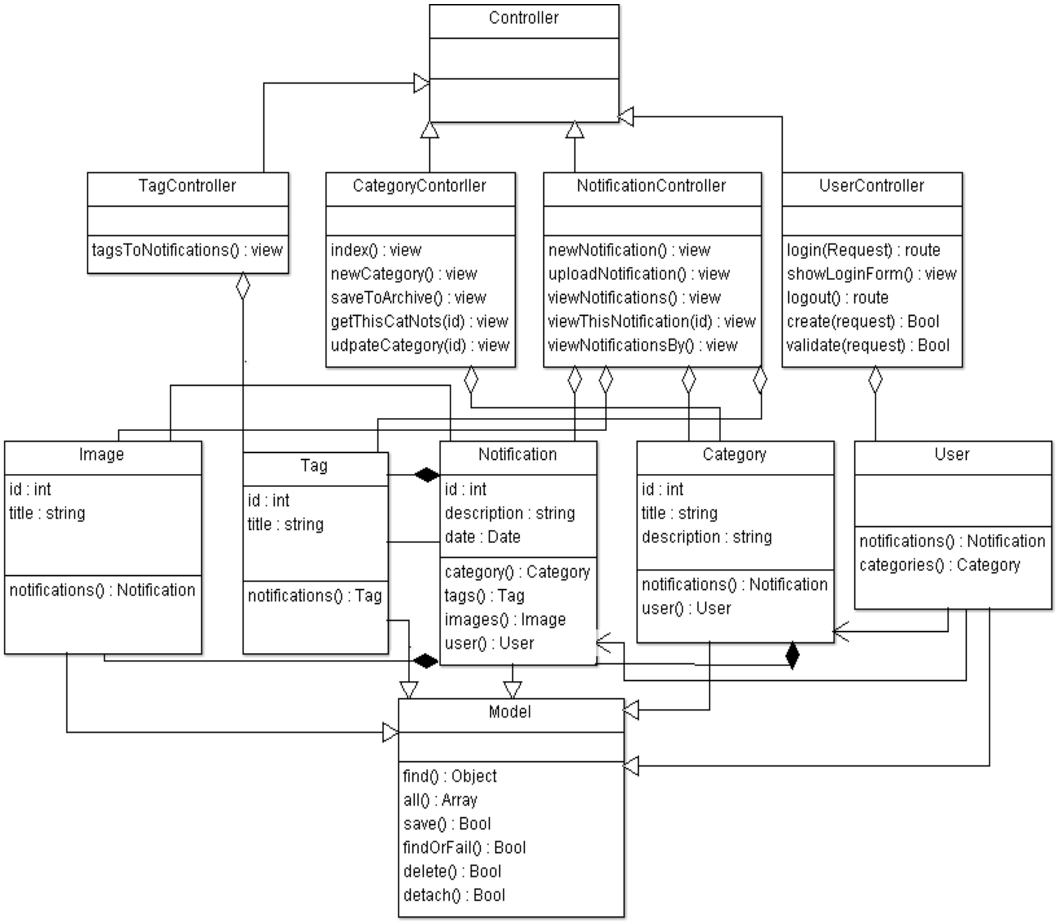


Figure 3. 13 Class Diagram

3.5 User Interfaces Design

The user interface(UI) is the part of the software most visible to the user and is one of the most important to get right. It includes prototypes of screen images of the system. There are two approaches for prototyping system design.

Low Fidelity

In low fidelity prototyping, screens images are drawn by hand.

High Fidelity

In high fidelity prototyping, screen images a designed with variety of screen design software like visual basic, pencil tool etc.

3.5.1 User Interface Screens

Log in UI

User should login to use the system. He will enter username and password and click login.

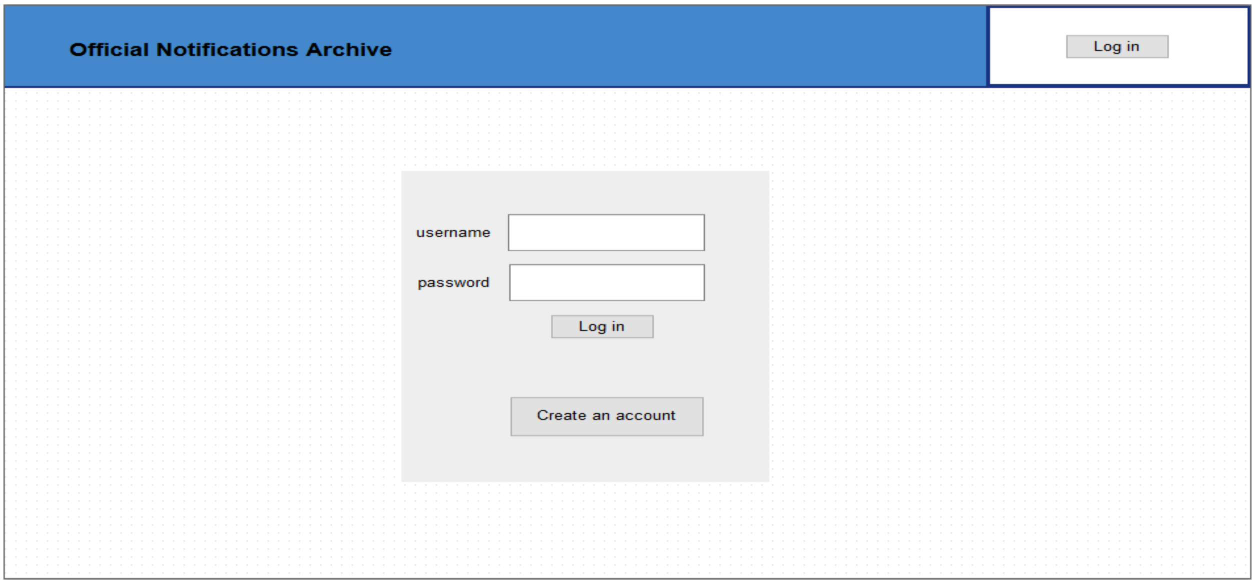


Figure 3. 14 Login UI

Sign up UI

This screen is responsible for new user account. This contains a form asking user information. User will fill this form and click on Create Account button. Upon validation of data, user will be sent a confirmation email. After confirming user email, user will be asked to login on system. When he will be successfully logged in on system. He can user system for further operations.

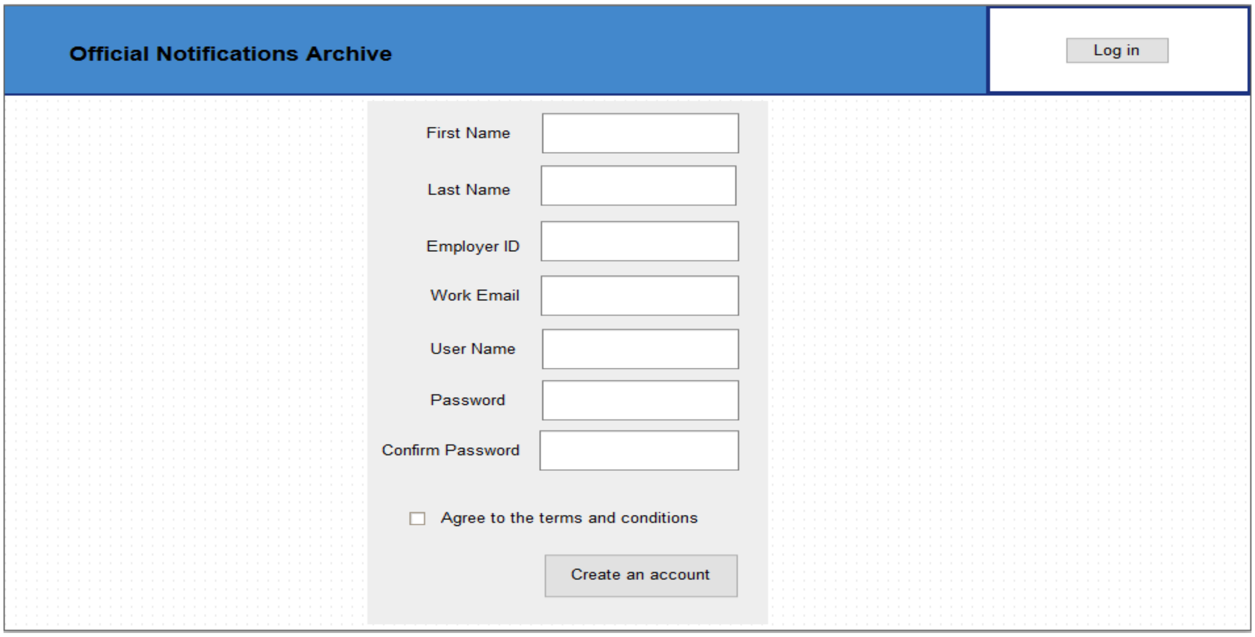


Figure 3. 15 Create account UI

Search Notification UI

User can search notification from archive by keywords and date. User will enter keyword in search field. If his keyword matches with notifications. He will be given results.

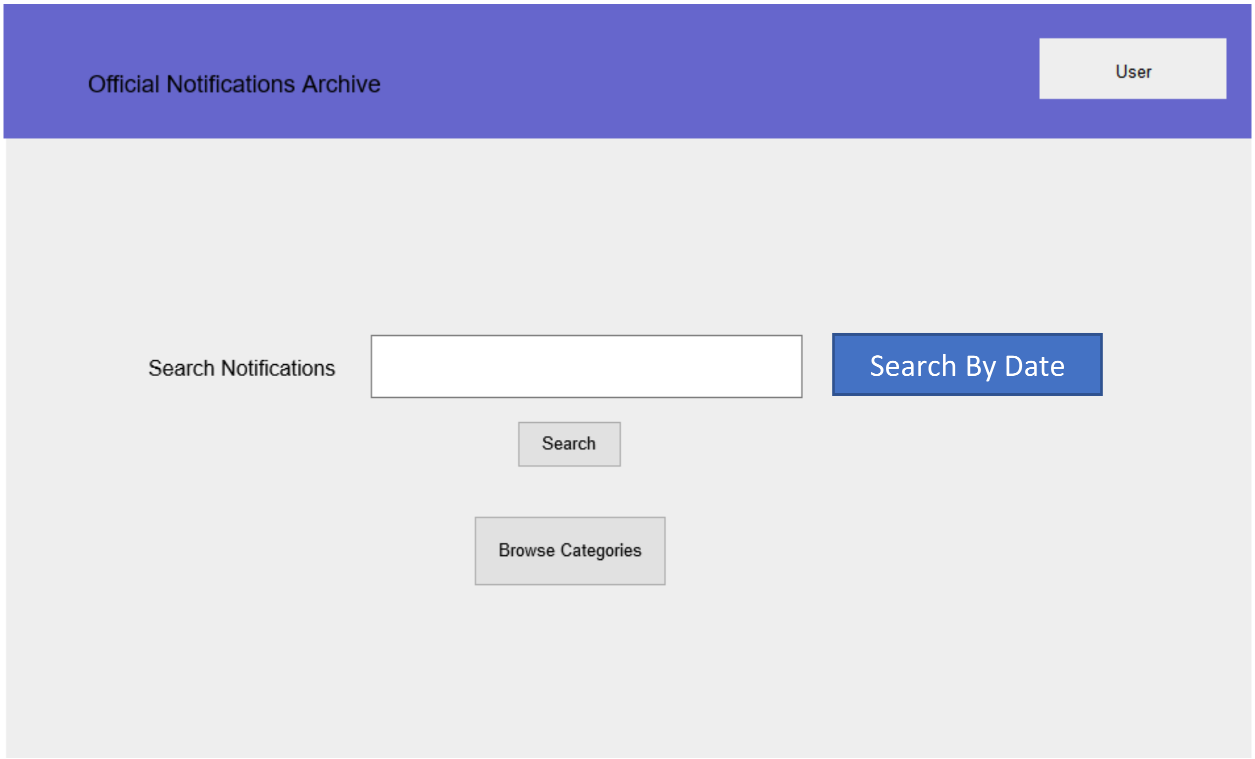
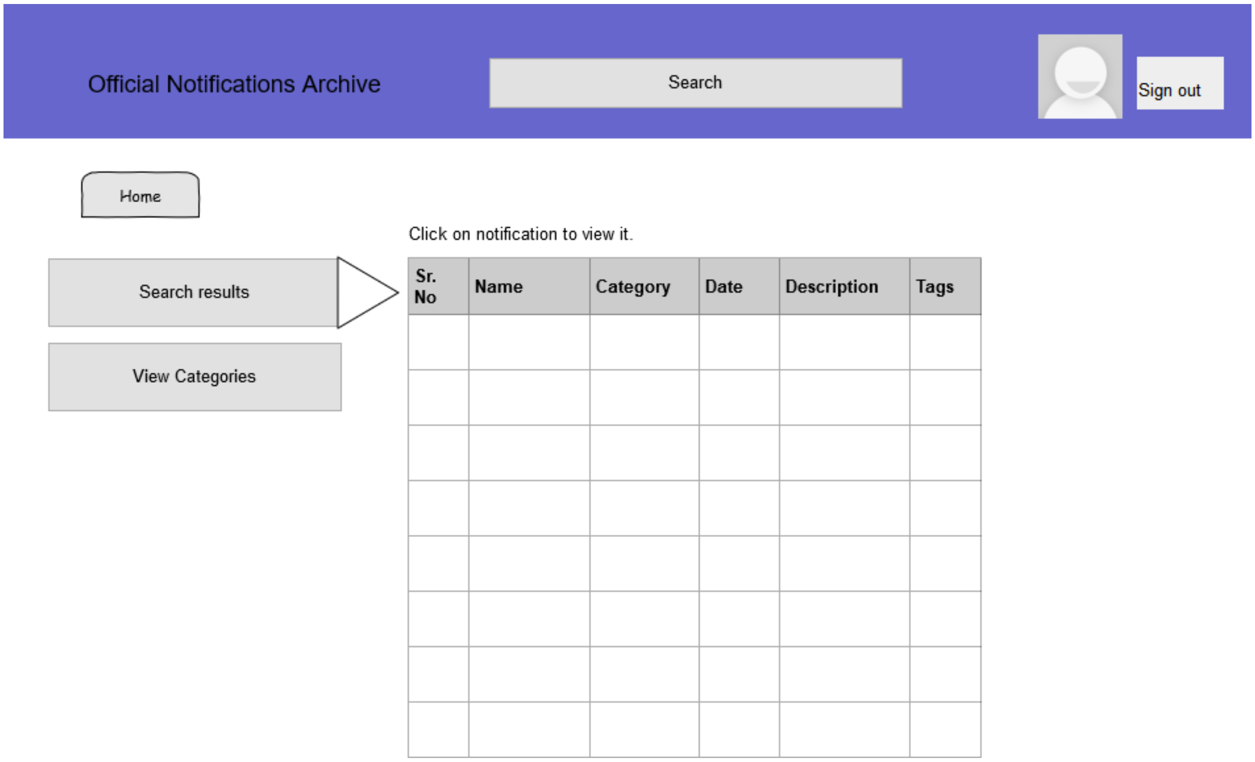


Figure 3. 16 Search Notification UI

Search Results UI

This screen will display search results in a table form. All notifications matching user search criteria will be displayed on this page. User further can filer notifications.



Click on notification to view it.

Sr. No	Name	Category	Date	Description	Tags

Figure 3. 17 Search results UI

Browse Categories UI

This screen will display all categories in one place. User can further click on a specific category.

This screen also has search field and Home button.

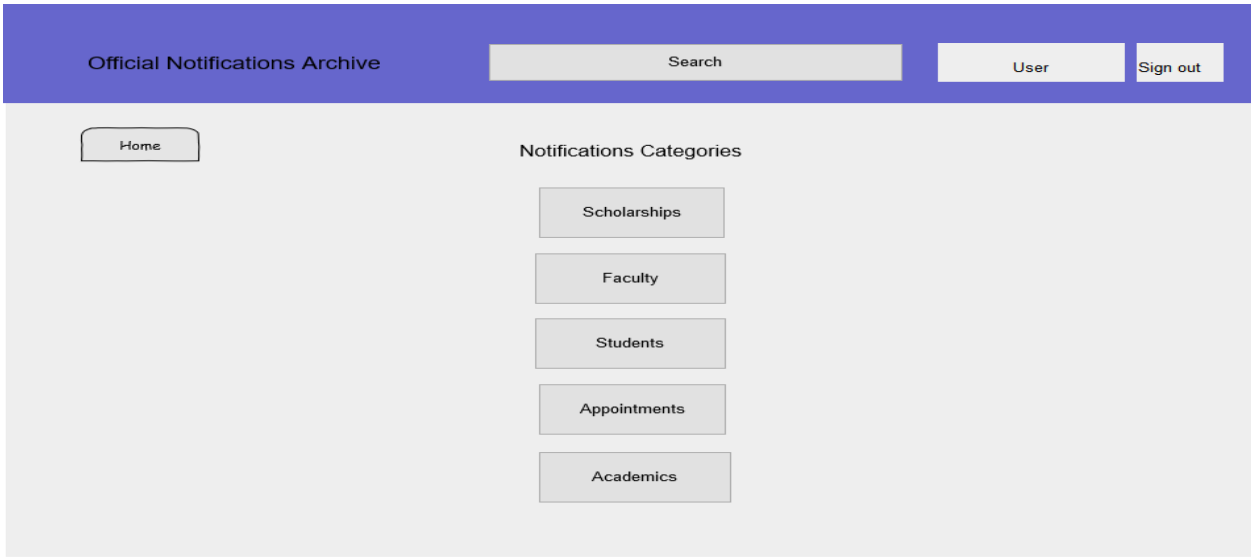
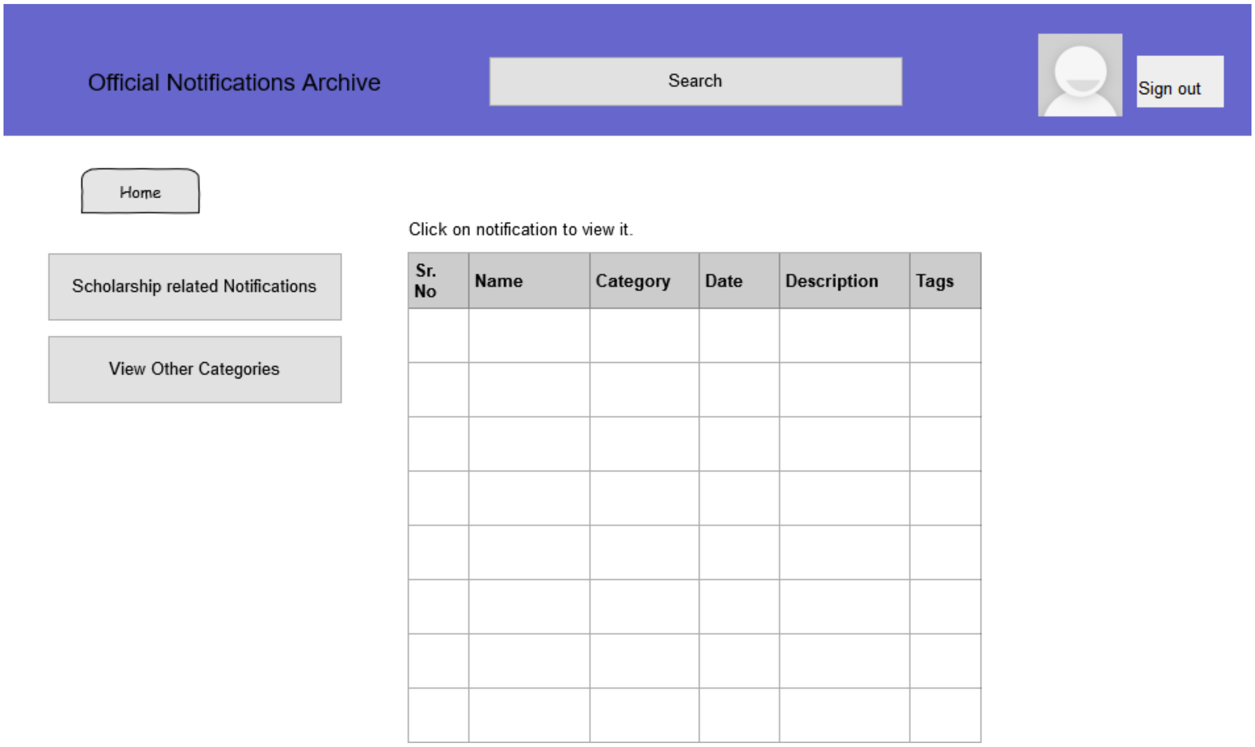


Figure 3. 18 Category View UI

Category Browse UI

This screen is to show contents of categories. All notifications under a specific category. All notifications will be displayed on this screen.


Official Notifications Archive
Search
Sign out
Home
Scholarship related Notifications
View Other Categories

Click on notification to view it.

Sr. No	Name	Category	Date	Description	Tags

Figure 3. 19 Category browse UI

CHAPTER 4

SOFTWARE IMPLEMENTATIONS

4.1 Introduction

In this chapter, we will discuss system implementation, approaches taken, tools used, framework and language which was used in implementation of this project.

4.2 Programming Language

Official Notifications Archive is a web based application. Languages used in this project are following.

1. HTML
2. CSS
3. Java Script
4. JQuery
5. PHP

Front End Design Framework

Front end design framework used for designing user interfaces is bootstrap framework.

PHP MVC Framework

The system is built using model view controller approach. Framework used for this purpose is Laravel 5.4 Framework.

4.3 Laravel Framework

Laravel Framework is PHP MVC Framework. It is very simple and elegant tool for developing web applications.

4.4 Software and Tools used

Following software are used in the implementation process.

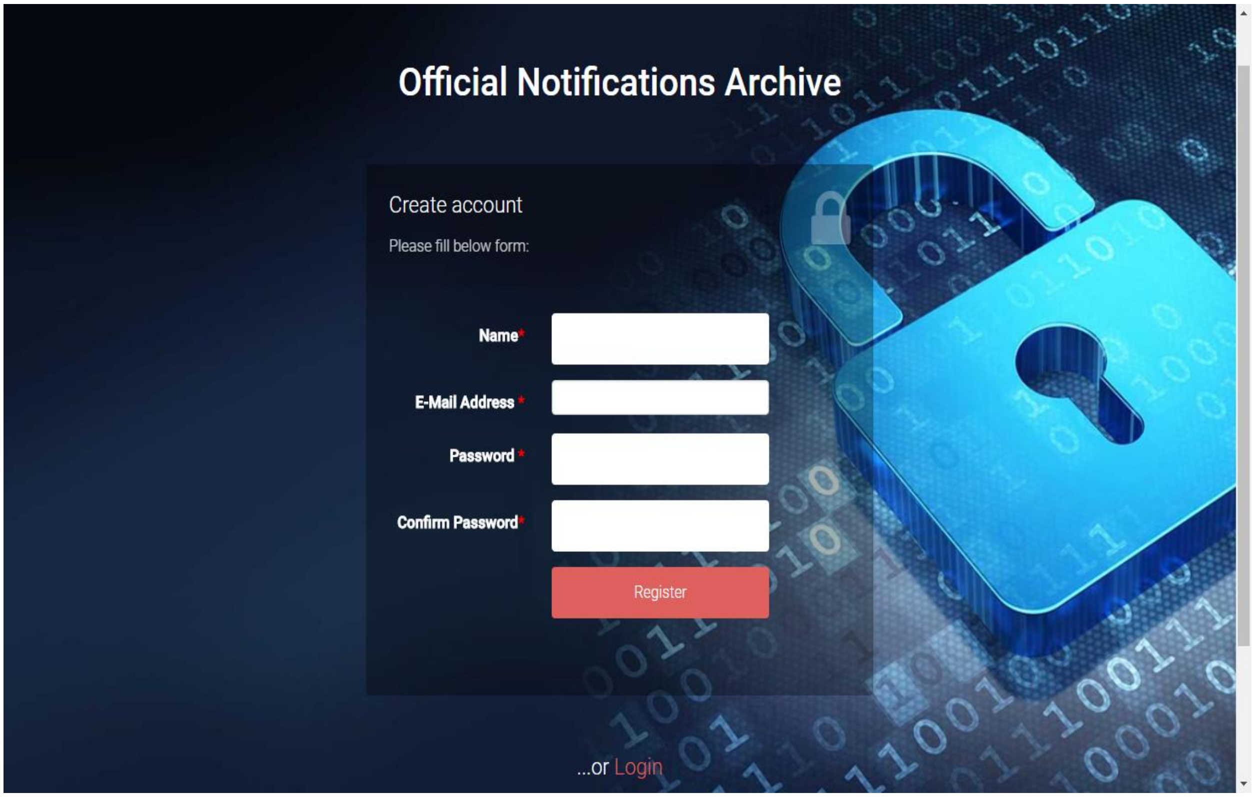
1. **Sublime Text for HTML and PHP coding**
2. **Wamp Server as a test server**
3. **Google Chrome, Mozilla Firefox for testing website**

4.5Screens

4.5.1 Screens for Admin

Following are admin screens

1. Sign up



Official Notifications Archive

Create account

Please fill below form:

Name*

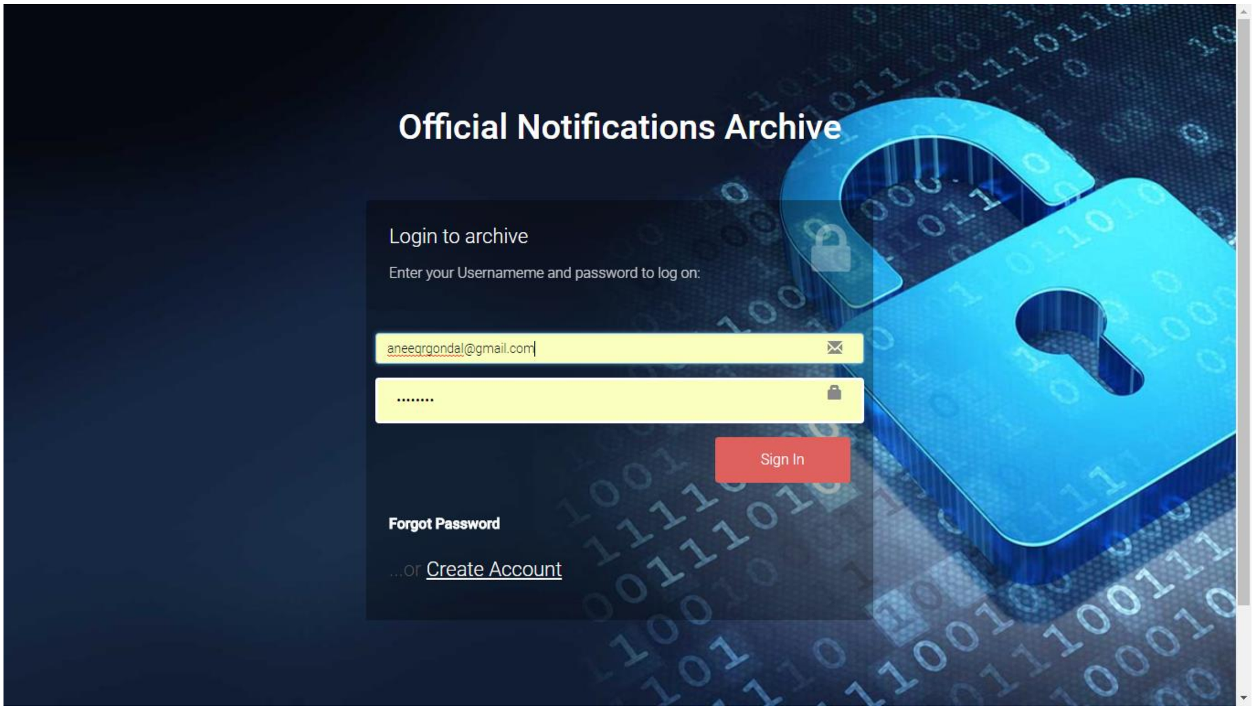
E-Mail Address *

Password *

Confirm Password*

Figure 4. 1 *Sign Up Screen*

2. Login



Enter your Usernameme and password to log on:

aneeqrgondal@gmail.com

....

Figure 4. 2 Sign in screen

3. Home Screens

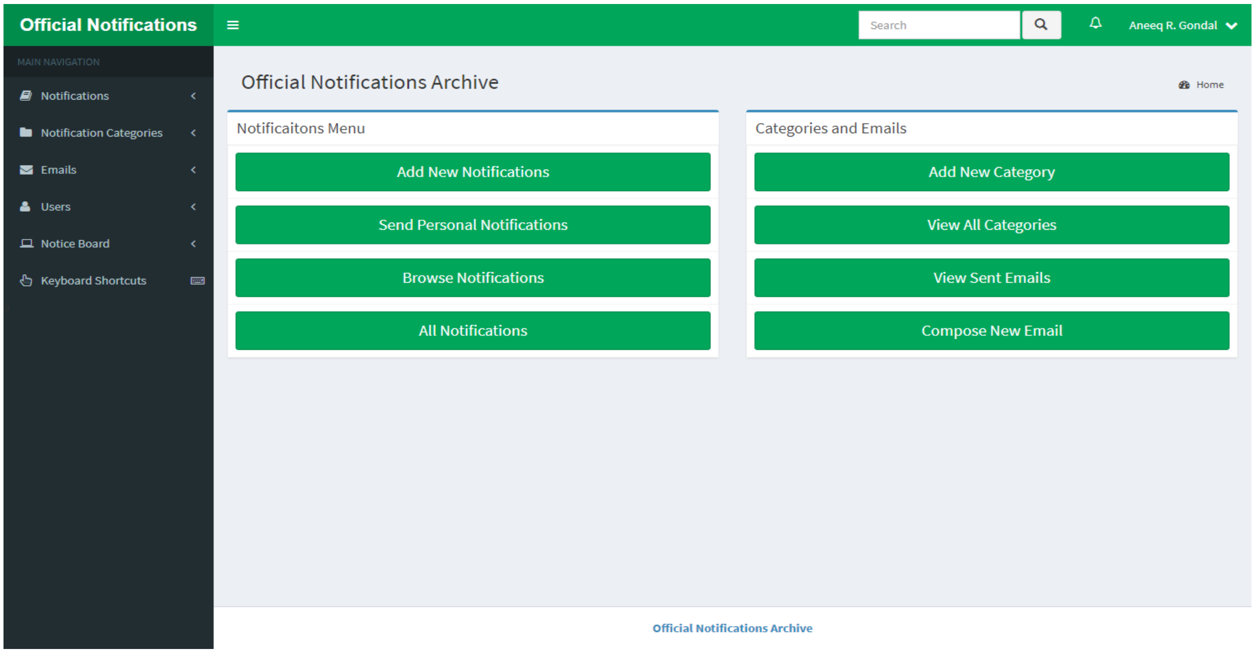


Figure 4. 3 Home Screen

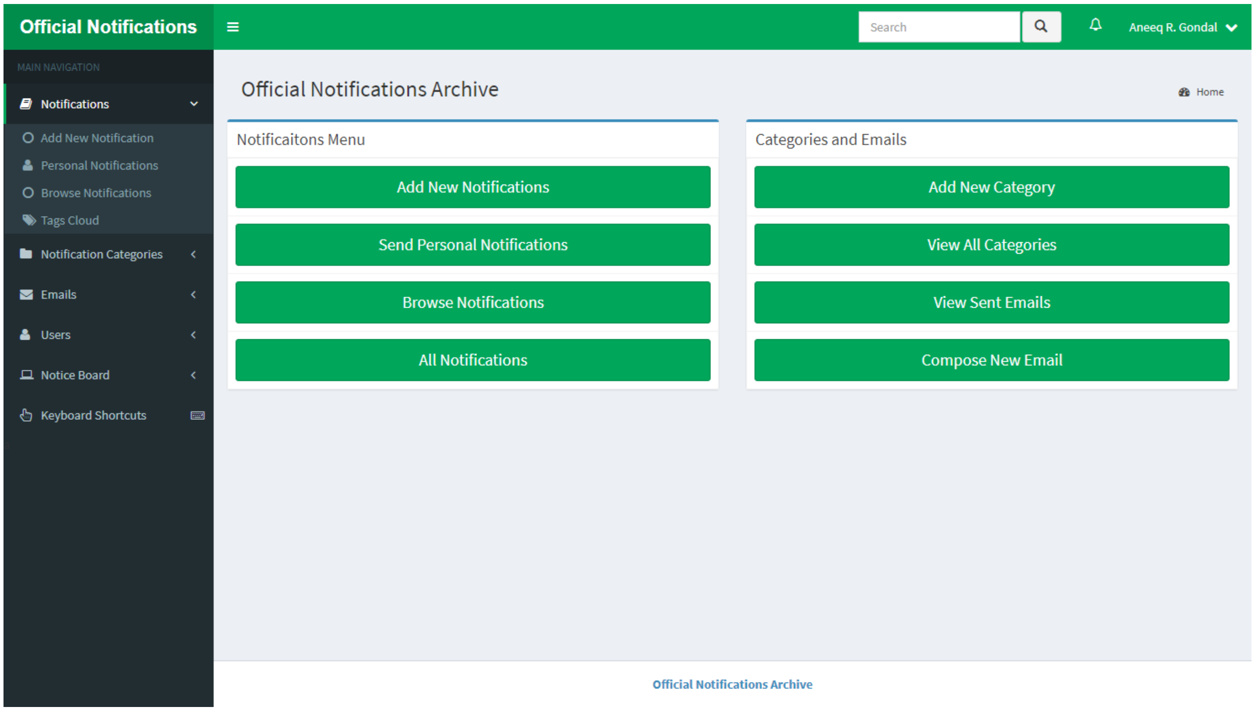
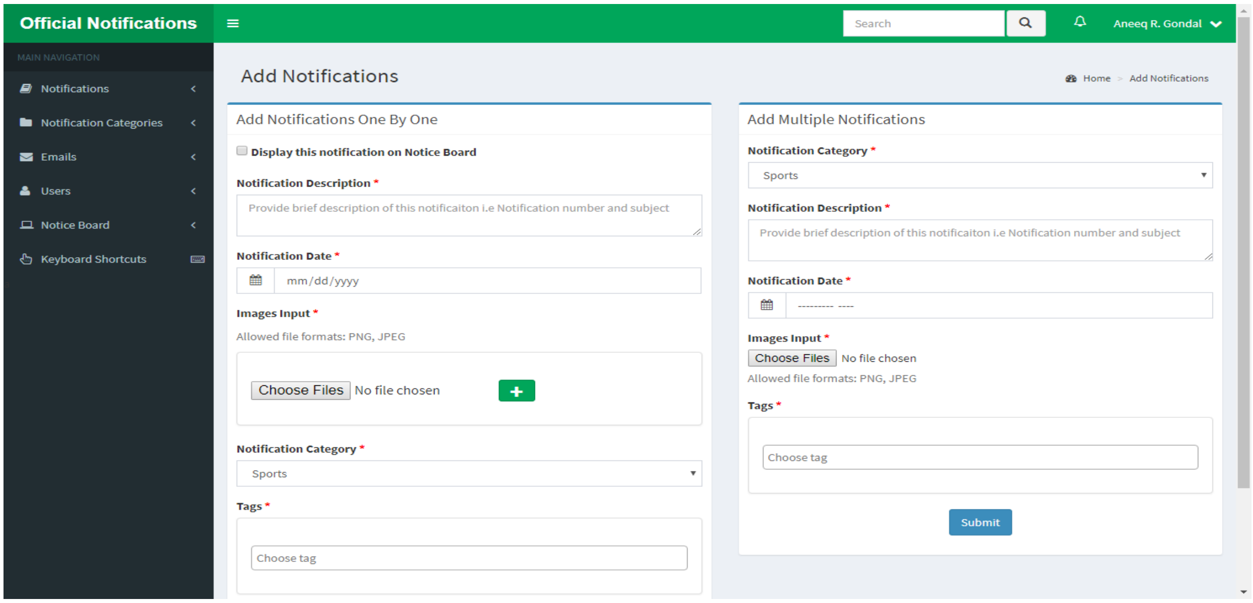


***Figure 4. 4** Home Screen 2*

4. Add new Notification



Add Notifications One By One

□ Display this notification on Notice Board

Notification Description *
Provide brief description of this notificaiton i.e Notification number and subject

Notification Date *
mm/dd/yyyy

Images Input *
Allowed file formats: PNG, JPEG
Choose Files No file chosen

Notification Category *
Sports

Tags *
Choose tag

Add Multiple Notifications

Notification Category *
Sports

Notification Description *
Provide brief description of this notificaiton i.e Notification number and subject

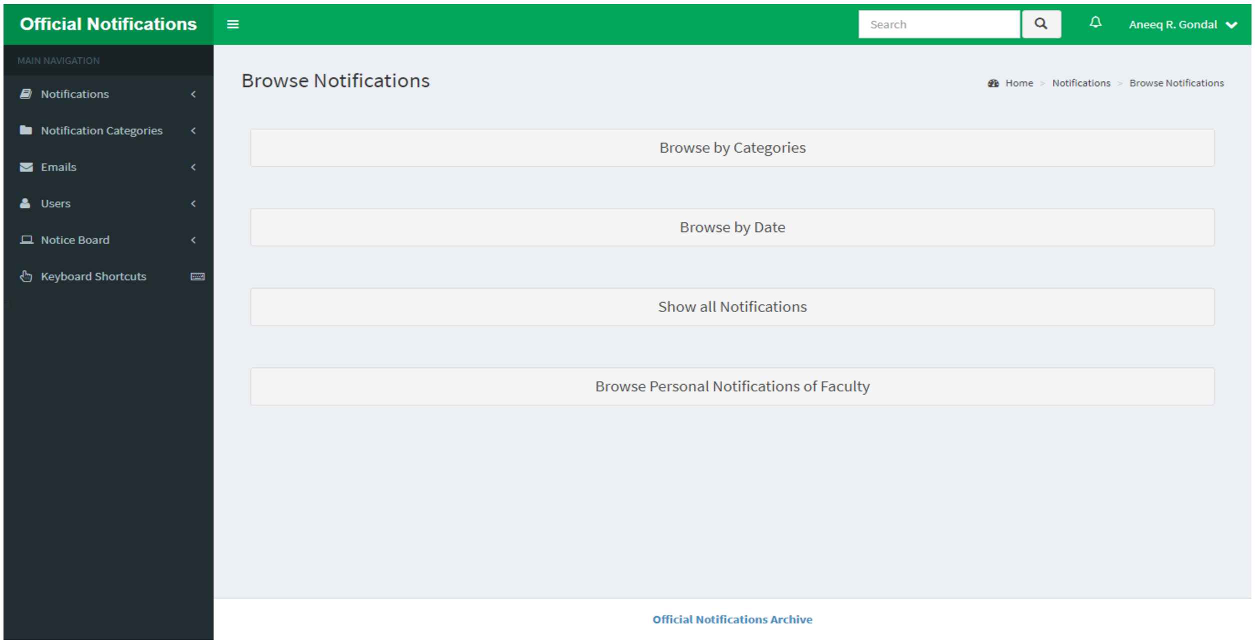
Notification Date *
---------- ----

Images Input *
Choose Files No file chosen
Allowed file formats: PNG, JPEG

Tags *
Choose tag

***Figure 4. 5** Add Notification Screen*

5. View Notifications



Browse Notifications

Browse by Categories

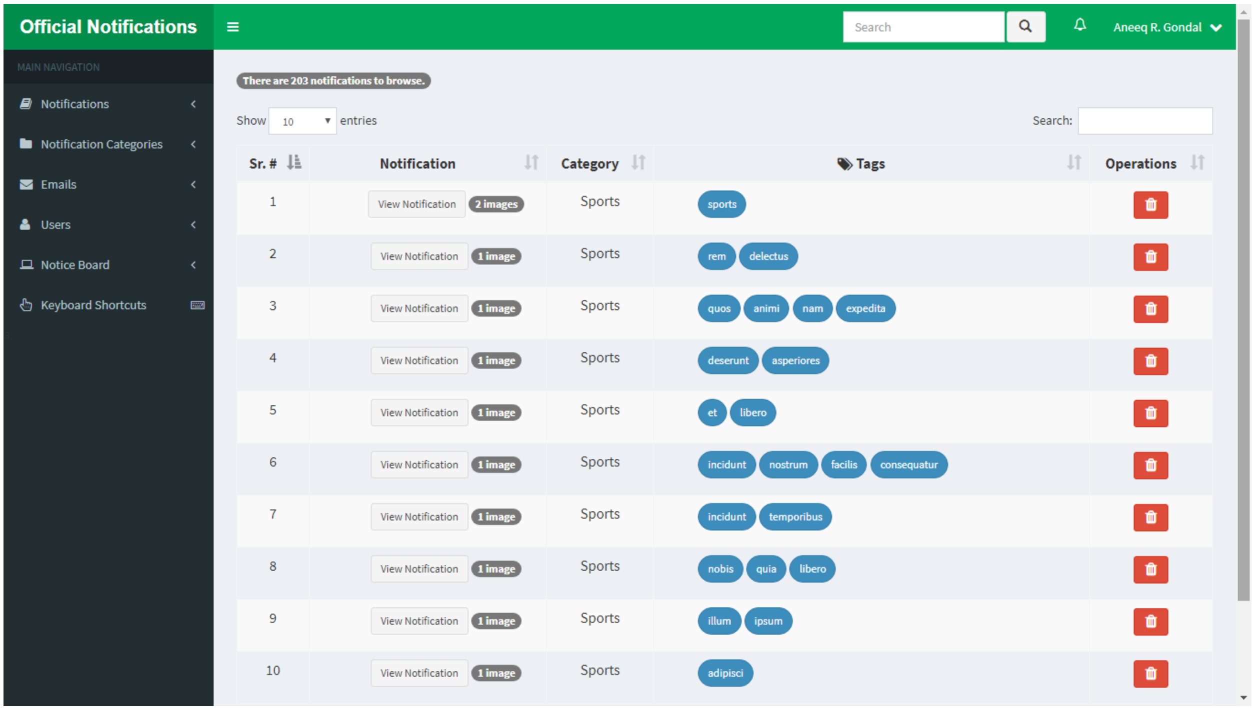
Browse by Date

Show all Notifications

Browse Personal Notifications of Faculty

Official Notifications Archive

Figure 4. 6 Browse Notifications Screen



There are 203 notifications to browse.

Show 10 entries

Search:

Sr. #	Notification	Category	Tags	Operations
1	View Notification 2 images	Sports	sports	🗑
2	View Notification 1 image	Sports	rem delectus	🗑
3	View Notification 1 image	Sports	quos animi nam expedita	🗑
4	View Notification 1 image	Sports	deserunt asperiores	🗑
5	View Notification 1 image	Sports	et libero	🗑
6	View Notification 1 image	Sports	incidunt nostrum facilis consequatur	🗑
7	View Notification 1 image	Sports	incidunt temporibus	🗑
8	View Notification 1 image	Sports	nobis quia libero	🗑
9	View Notification 1 image	Sports	illum ipsum	🗑
10	View Notification 1 image	Sports	adipisci	🗑

Figure 4. 7 All Notifications Screen

6. Add Category

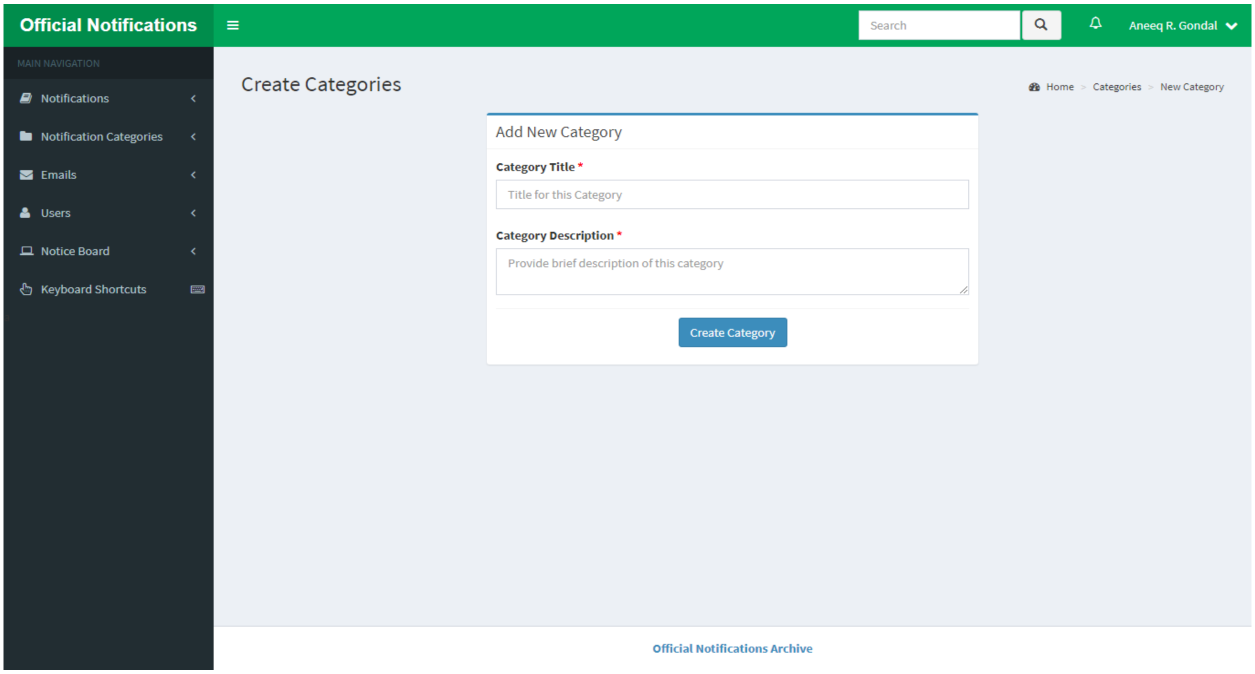
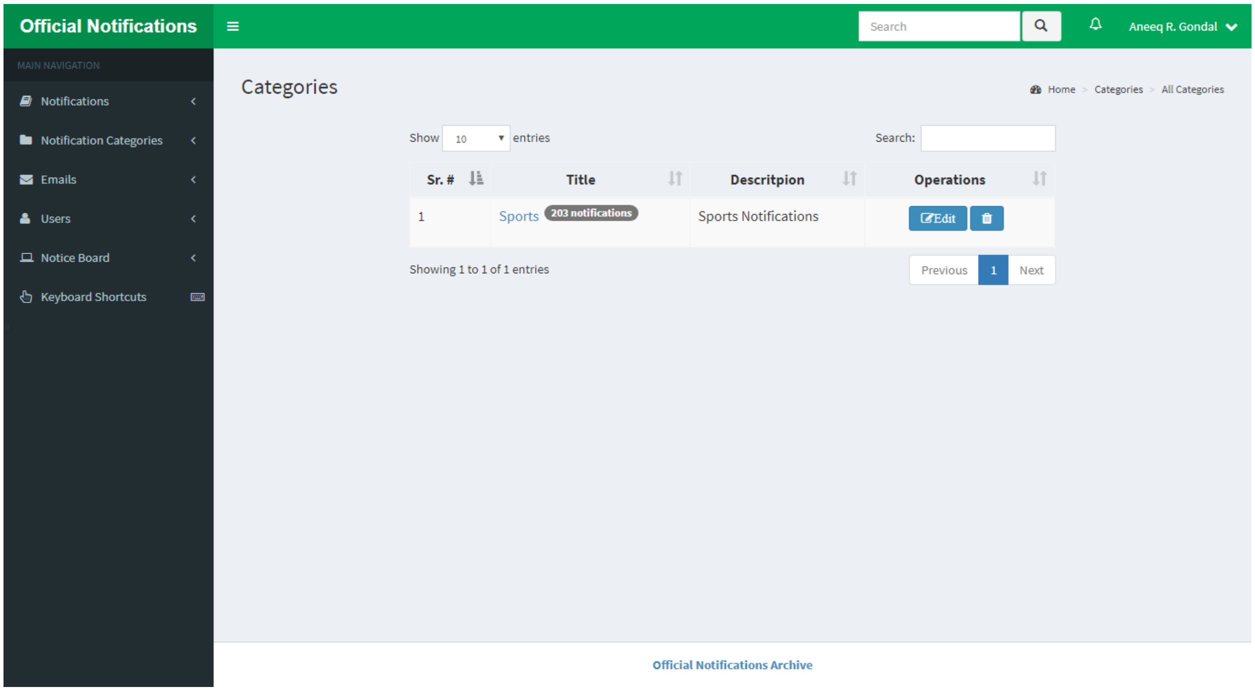


Figure 4. 8 Add Category Screen

7. View Categories



Sr. #	Title	Descritpion	Operations
1	Sports 203 notifications	Sports Notifications	Edit

Showing 1 to 1 of 1 entries

Figure 4. 9 All Categories

8. Email Notifications

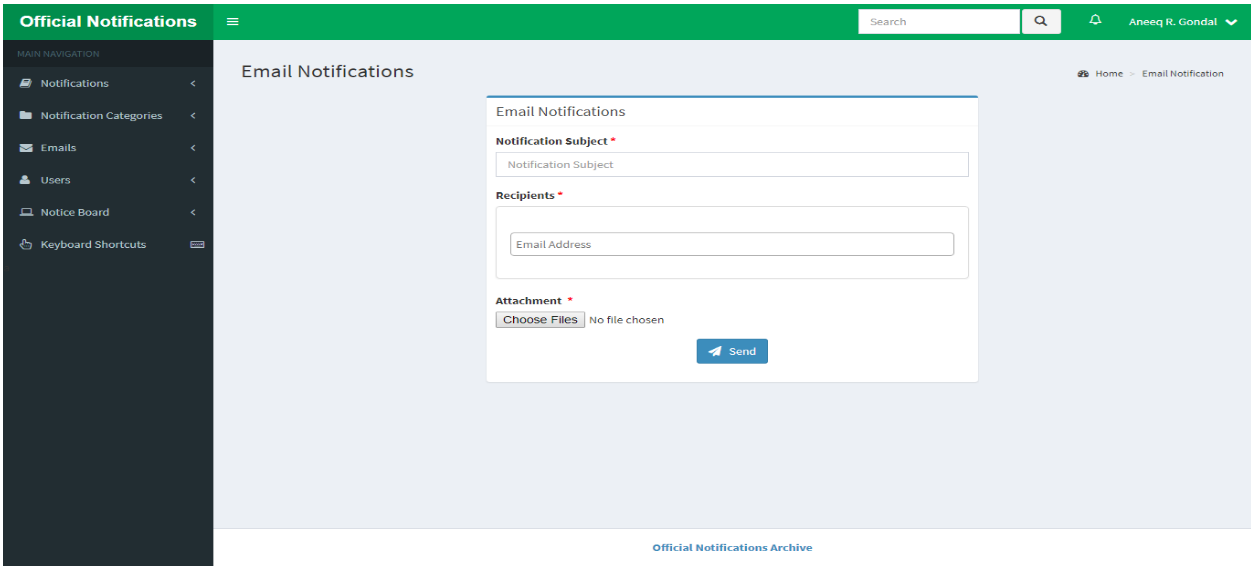


__Figure 4. 10__ Send Email Screen

4.5.2 Screens for Faculty

1. Sign Up

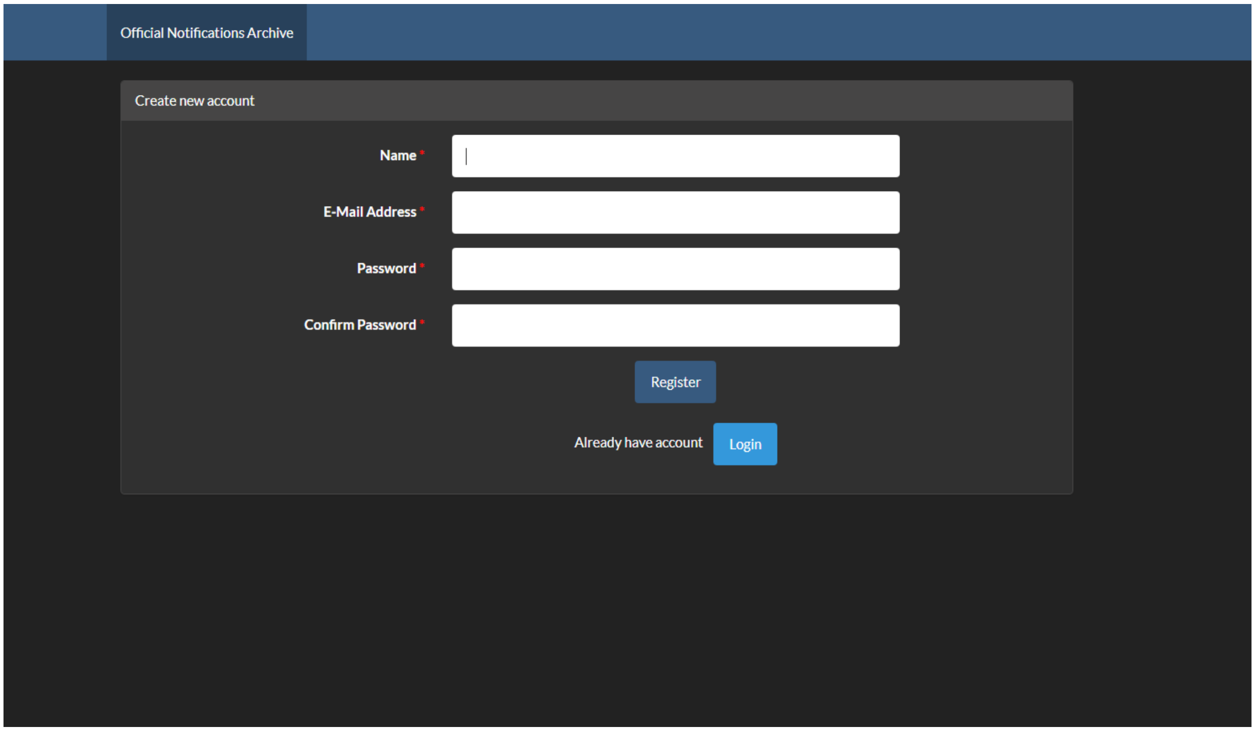
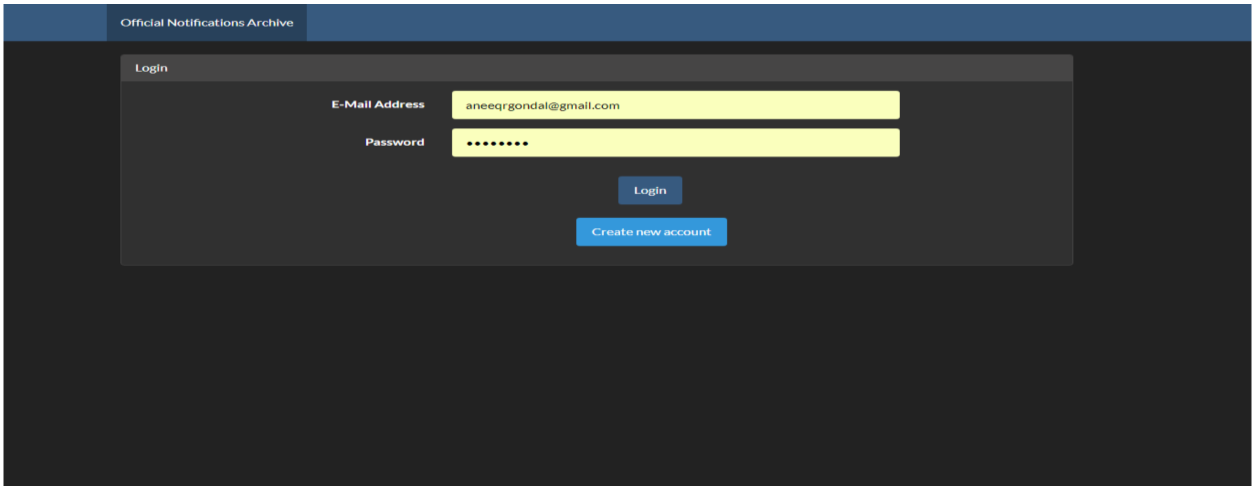


__Figure 4. 11__ Register Screen

2. Sign in

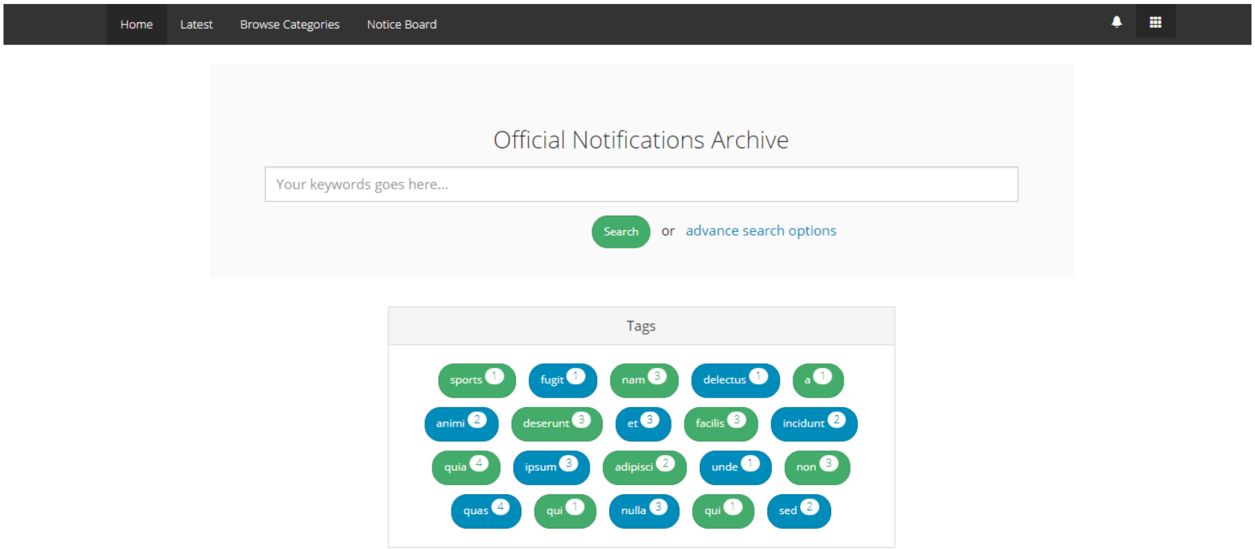


E-Mail Address: aneeqrgondal@gmail.com

Password: ••••••••

Figure 4. 12 Sign in Screen

3. Search



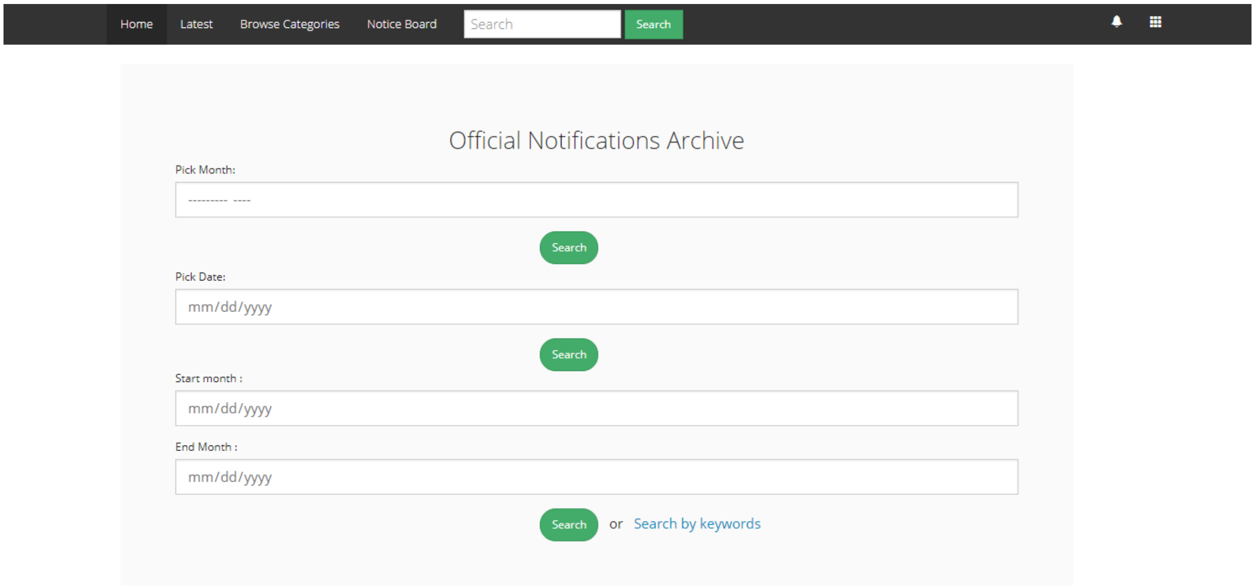
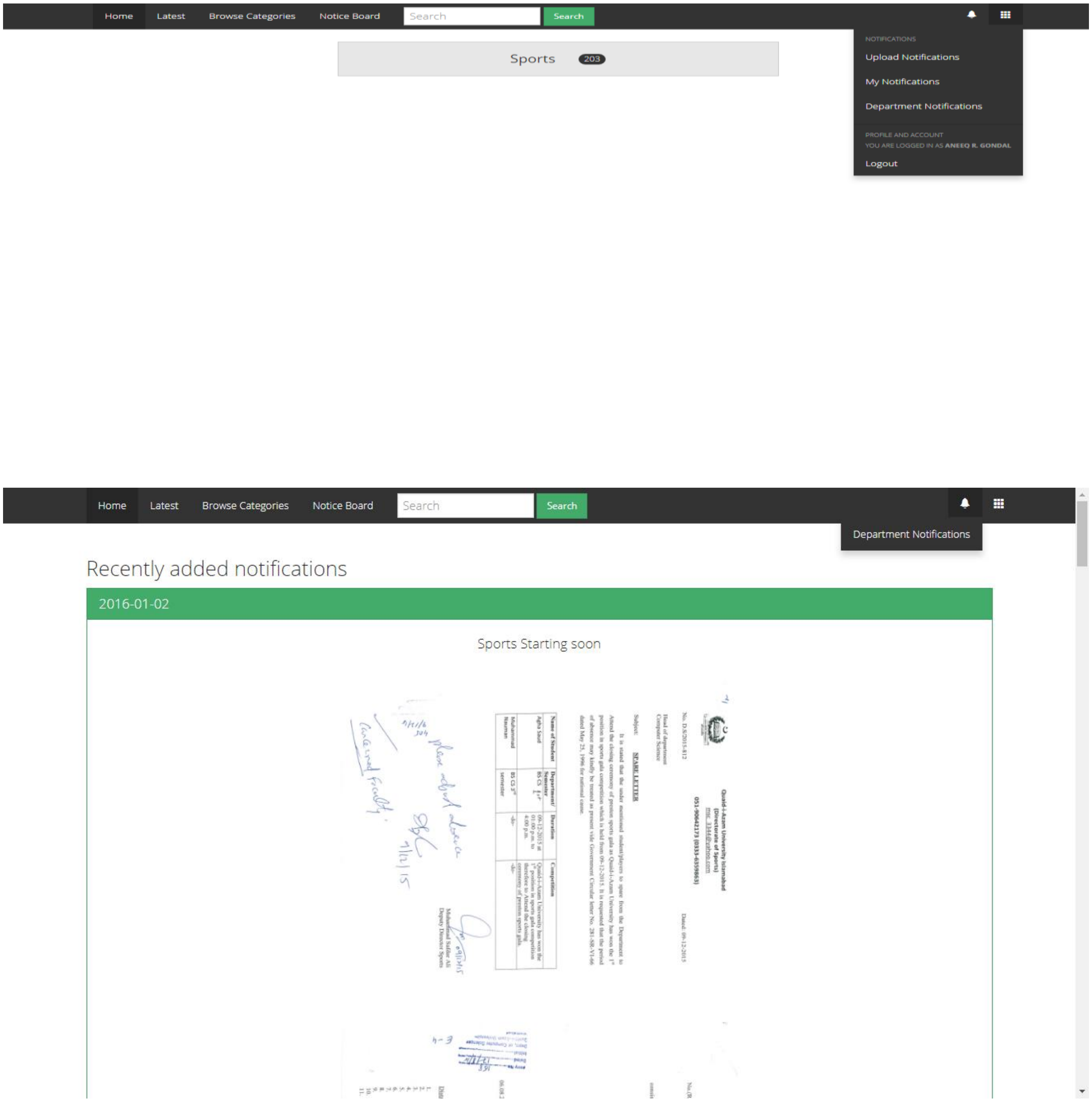


***Figure 4. 13** Search Screens for users*

4. Browse By Categories



Recently added notifications

2016-01-02

Sports Starting soon

Quaid-i-Azam University Islamabad
(Directorate of Sports)
msc_3344@yahoo.com
051-90642173 (0333-6359863)

No. D.S/2015-812

Dated: 09-12-2015

Head of department
Computer Science

Subject: SPARE LETTER

It is stated that the under mentioned student/players to spare from the Department to Attend the closing ceremony of prestine sports gala as Quaid-i-Azam University has won the 1st position in sports gala competition which is held from 09-12-2015. It is requested that the period of absence may kindly be treated as present vide Government Circular letter No. 281-SR-V1-66 dated May 25, 1966 for national cause.

Name of Student	Department/ Semester	Duration	Competition
Agha Saud	BS CS 1st	09-12-2015 at 01:00 p.m. to 4:00 p.m.	Quaid-i-Azam University has won the 1st position in sports gala competition therefore to Attend the closing ceremony of prestine sports gala.
Muhammad Nauman	BS CS 3rd semester	-do-	-do-

09/12/15
Muhammad Safdar Ali
Deputy Director Sports

Please adjust absence
9/12/15

Concerned Faculty.

Figure 4. 14 Browse Categories Screen

CHAPTER 5

SOFTWARE TEST DOCUMENTATION

5 SOFTWARE TEST DOCUMENTATION

5.1 Introduction

This chapter is about testing of system. Testing is the process to verify that system works correctly, according to requirements.

5.2 System Overview

Official Notifications Archive is a web based application having two types of users. One is Faculty of Compute Science Department. The other type of user is administrator who manages archive. Admin can add, update, delete and search notifications while faculty can only search or browse notifications. Notifications are added as image files. These notifications have different categories and tags. Notification can be searched by date, category and tag.

5.2.1 Test Approach

We are using Requirements Testing Technique to test our requirements. The objectives of requirement testing are following.

- Implement user requirements
- Maintain correctness over extended processing periods.
- Ensure that all features of application are being implemented.

Link:

http://www.msqaa.org/Best_Practices/Quality_Control/FunctionalSystemTestingTechniqueCategories.pdf

5.3 Test Plan

Test Plan is a formal document that describes our strategy or approach of testing the software.

5.3.1 Features to be tested

Features to be tested for Official Notifications Archive are

1. Login
2. Sign Up
3. Searching
4. Adding Notifications
5. Add Category
6. Browse Category

5.4 Test Cases

Test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. Test cases for Official Notifications Archive are given below. The process of developing test cases can also help find problems in the requirements or design of an application.

5.4.1 Log in

User must login on system to use the system. To login user has unique id and password.

1 Purpose

The Purpose of this test case is to test the functionality of login feature. If login works, user must be redirected to main menu system from where he can perform further operations.

2 Inputs

Inputs for this test case are username and password.

3 Expected outputs and pass/fail criteria

If the valid username and password is entered, user must see successful login message. If the user is logged in on the system, test case is passed otherwise it the test case will be failed.

4 Test Procedure with correct data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-001
Test Case Summary	Modify Category
Related Use Case	UC-002
Prerequisite	• User is shown login page.
Test Procedure	1. User click on username field. 2. User enters username. 3. User clicks on password field. 4. User enters password. 5. User clicks Login button.
Test Data	**Username:** AteeqRGondal **Password:** ************
Expected Result	1. User should see successful login message.
Actual Result	Successful log in message is displayed to the user.
Status	Pass
Remarks	Login feature of the system works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 1 Test Case TC-001

5 Test Procedure with incorrect data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-001
Test Case Summary	This test case to test login feature.

Related Use Case	UC-002
Prerequisite	- User is shown login page.’
Test Procedure	1. User click on username field. 2. User enters username. 3. User clicks on password field. 4. User enters password. 5. User clicks Login button.
Test Data	**Username:** AteeqGondal **Password:** *****
Expected Result	2. User should see unsuccessful login message.
Actual Result	Unsuccessful log in message is displayed to the user.
Status	Pass
Remarks	Login feature of the system works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 2 Test Case TC-001

5.4.2 Sign up

User has to create account on system to use the system. User must have unique registration id and active email address. User will be asked to confirm his/her email address.

1 Purpose

The Purpose of this test case is to test the functionality of registration feature. If registration works, user must be see successful registration message.

2 Inputs

Inputs for this test case are name, email address, registration ID, username and password.

3 Expected outputs and pass/fail criteria

If the valid registration id which is unique for every person in university and email is entered, user should be registered. If the user successfully created new account on the system, test case is passed otherwise it the test case will be failed.

4 Test Procedure with correct data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-002
Test Case Summary	This test will test the registration feature of system.
Related Use Case	UC-001
Prerequisite	• User is shown Create New Account page.
Test Procedure	1. User clicks on Name field. 2. User enters Name in valid format. 3. User clicks on Registration ID field. 4. User enters Registration ID. 5. User clicks Email field. 6. User enters Email. 7. User clicks on username field. 8. User enters username. 9. User clicks password field. 10. User enters password in allowed format. 11. User clicks on Create Account button.
Test Data	**Name:** Ateeq-Ur-Rehman Gondal **Registration ID: 04071313033** **Email Address** ateeqrgondal@gmail.com **Username: AteeqRGondal** **Password:** ********
Expected Result	3. User should see successful registration message.
Actual Result	Successful registration message is displayed to the user.
Status	Pass
Remarks	Registration feature of system works correctly.

Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 3 Test Case TC-002

5 Test Procedure with incorrect data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-002
Test Case Summary	This test will test the registration feature of system.
Related Use Case	UC-001
Prerequisite	• User is shown Create New Account page.
Test Procedure	1. User clicks on Name field. 2. User enters Name in valid format. 3. User clicks on Registration ID field. 4. User enters Registration ID. 5. User clicks Email field. 6. User enters Email. 7. User clicks on username field. 8. User enters username. 9. User clicks password field. 10. User enters password in allowed format. 11. User clicks on Create Account button.
Test Data	**Name:** 1234564789 **Registration ID: 04071313033** **Email Address** ateeqrgondal@gmail.com **Username: AteeqRGondal** **Password:** ******
Expected Result	4. User should see successful registration message.
Actual Result	Successful registration message is displayed to the user.
Status	Pass
Remarks	Registration feature of system works correctly.

Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 4 Test Case TC-004

5.4.3 Search Notification

User can perform search operation by a keyword, date, category. He enters keyword to search notification. If the keyword is matched with any tag or category, result will be displayed to the user.

1 Purpose

The Purpose of this test case is to test search functionality of system. User should see results based on his search criteria.

2 Inputs

Inputs for this test case is keyword or date

3 Expected outputs and pass/fail criteria

If keyword is matched with any notification, result should be displayed. If user enters a keyword and he is given result based on the keyword, test case is passed. If user enters a relevant keyword but does not see any result, test case is failed.

4 Test Procedure with correct data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-003
Test Case Summary	This test will test search functionality of system.
Related Use Case	UC-009
Prerequisite	• User is shown search place.
Test Procedure	1. User clicks on Search field.

	2. User enters keyword or date. 3. User clicks on Search Button.
Test Data	**Keyword: student**
Expected Result	5. User should see some results.
Actual Result	Following notifications matched your keyword.
Status	Pass
Remarks	Searching works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 5 Test Case TC-003

5 Test Procedure with incorrect data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-003
Test Case Summary	This test will test search functionality of system.
Related Use Case	UC-009
Prerequisite	• User is shown search place.
Test Procedure	4. User clicks on Search field. 5. User enters some numbers as keyword or date. 6. User clicks on Search Button.
Test Data	**Keyword:** student
Expected Result	6. User should see some results.
Actual Result	No notification matched your criteria.
Status	Pass

Remarks	Searching works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 6 Test Case TC-003

5.4.4 Add Notification

Admin can add new notification to archive. He will have image of notification.

1 Purpose

The Purpose of this test case is to test adding notification functionality of system.

2 Inputs

Inputs for this test case is new notification.

3 Expected outputs and pass/fail criteria

If notification is added to system, user should see success message. If notifications is not added to system, test case will be failed.

4 Test Procedure with correct data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-004
Test Case Summary	This test will test adding notification functionality of system.
Related Use Case	UC-006
Prerequisite	• User is shown Add New Notification section.
Test Procedure	1. User clicks on Add New Notification Button. 2. User enters subject of notification. 3. User enters category of notification. 4. User upload image file of notification. 5. User clicks on Upload this Notification button.

Test Data	**Notification Image:** notification0001.jpg **Notification Subject:** Apply for scholarships.
Expected Result	1. Notification added to archive successfully.
Actual Result	Notification added to archive successfully.
Status	Pass
Remarks	Adding notification works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 7 Test Case TC-004

5 Test Procedure with incorrect data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-004
Test Case Summary	This test will test adding notification functionality of system.
Related Use Case	UC-006
Prerequisite	• User is shown Add New Notification section.
Test Procedure	1. User clicks on Add New Notification Button. 2. User enters subject of notification. 3. User enters category of notification. 4. User upload image file of notification. 5. User clicks on Upload this Notification button.
Test Data	**Notification Image:** notification0001.txt **Notification Subject:** Apply for scholarships.
Expected Result	1. Incorrect image format.
Actual Result	Incorrect image format. This file type is not allowed.
Status	Pass

Remarks	Adding notification works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 10,2017
Date of Execution	July 10,2017

Table 4. 8 Test Case TC-004

5.4.5 Add Category

Admin can define categories of notifications. Notifications are stored under these categories.

1 Purpose

The Purpose of this test case is to test adding category functionality of system.

2 Inputs

Inputs for this test case is Category title.

3 Expected outputs and pass/fail criteria

If category is added to system, user should see success message. If category is not added to system, test case will be failed.

4 Test Procedure with correct data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-005
Test Case Summary	This test will test adding category functionality of system.
Related Use Case	UC-003
Prerequisite	• User is shown Add New Category section.
Test Procedure	6. User clicks on Add New Category Button. 7. User enters title of category. 8. User enters description of category. 9. User clicks on Add Category button.
Test Data	**Category title:** Scholarships

	Category Description: In this category, scholarship related notifications will be stored.
Expected Result	1. Category added to archive successfully.
Actual Result	Category added to archive successfully.
Status	Pass
Remarks	Adding Category works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 9 Test Case TC-005

5 Test Procedure with incorrect data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-005
Test Case Summary	This test will test adding category functionality of system.
Related Use Case	UC-003
Prerequisite	• User is shown Add New Category section.
Test Procedure	10. User clicks on Add New Category Button. 11. User enters title of category. 12. User enters description of category. 13. User clicks on Add Category button.
Test Data	**Category title:** 123456ABCD **Category Description:** In this category, scholarship related notifications will be stored.
Expected Result	2. Incorrect category title.
Actual Result	Incorrect category title.
Status	Pass

Remarks	Adding Category works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 10 Test Case TC-005

5.4.6 Browse a Category

Categories can be browsed. The notifications under category will be displayed.

1 Purpose

The Purpose of this test case is to test browse categories functionality of system.

2 Inputs

No input is required for this test case. User just has to click Browse Category Button

3 Expected outputs and pass/fail criteria

If category contains notifications, user should see them.

4 Test Procedure with correct data

Test procedure describes step by step test execution. Table below shows procedure for this test case.

Test Case ID	TC-006
Test Case Summary	This test will test browsing category functionality of system.
Related Use Case	UC-009
Prerequisite	• User is shown Browse Category section.
Test Procedure	1. User clicks on browse category button. 2. User views notifications.
Test Data	**Category title:** Scholarships

Expected Result	Notifications under Scholarship category should be displayed.
Actual Result	User sees some notifications.
Status	Pass
Remarks	Browse Category functionality works correctly.
Created By	Ateeq R. Gondal
Date of Creation	July 03,2017
Date of Execution	July 03,2017

Table 4. 11 Test Case TC-006

5.5 Test Results

Table below shows results for tests.

Test Cases	Frequency	Passed	Failed
TC-001	5	5	0
TC-002	5	5	0
TC-003	5	5	0
TC-004	5	5	0
TC-005	5	5	0
TC-006	5	5	0

CHAPTER 6

CONCLUSIONS

6 CONCLUSIONS

6.1 Introduction

In this chapter we will conclude the whole project.

6.2 Conclusion

The purpose of this project is to archive official notifications. These notifications can be stored under different categories and tags. Notifications can be browsed by categories, tags and date. Admin add notifications to the archive. Notifications can be added one by one. If admin wants to add multiple notifications in the archive, he can add multiple notifications. Text from notifications can be extracted using OCR. This OCR text is used for searching purpose. User can search notifications by keywords. Users can upload personal notifications also in the archive. Admin can email notifications to the user.

6.2.1 Future Enhancement

This archive is web based application. It is possible to enhance the application in the future. Many other features can be added in this application.

- It can be android application. Admin just has to capture image of notifications using his camera and he can upload to the archive.
- Artificial Intelligence can be integrated in this project.
- This project can be expanded to make online system of notifications where less paper work will be involved.
- This project can lead to virtual notice board for students.
- A new module can be added for sharing study related stuff among students.

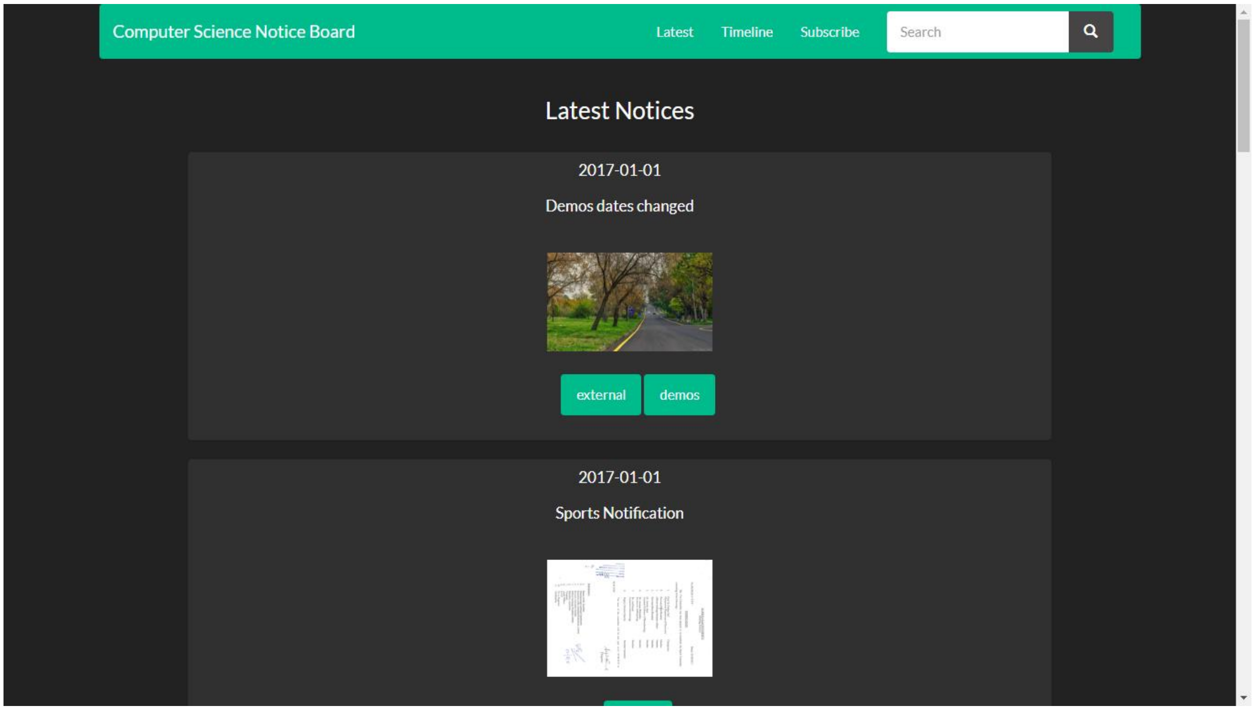
6.3 Extra Work

As extra work, I implemented notice board.

6.3.1 Notice Board

I also worked on notice board things where student can visit online notice board. They can subscribe notice board to get notified when there is something new posted on notice board of Computer Sciences Department. Student can search notice board, view latest notifications on notice board and can browse timeline. Below are the screens of notice board.

6.3.2 Notice Board Screens



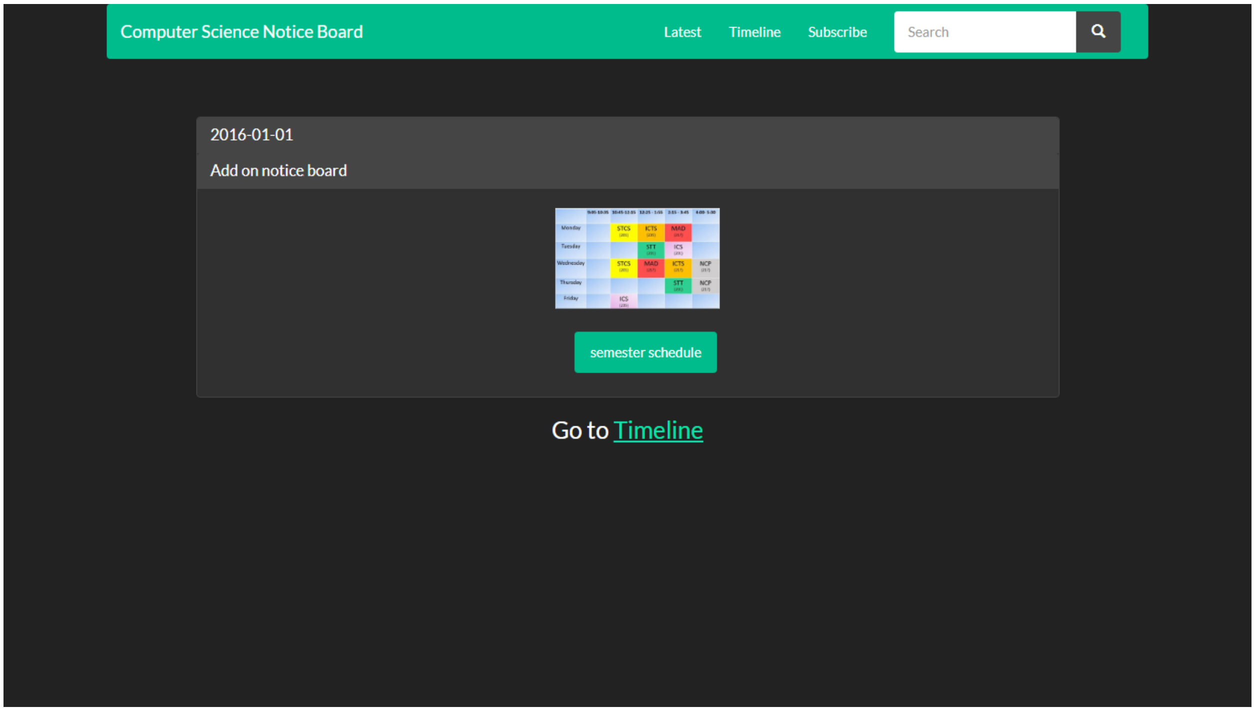
Latest Notices

2017-01-01

Demos dates changed

2017-01-01

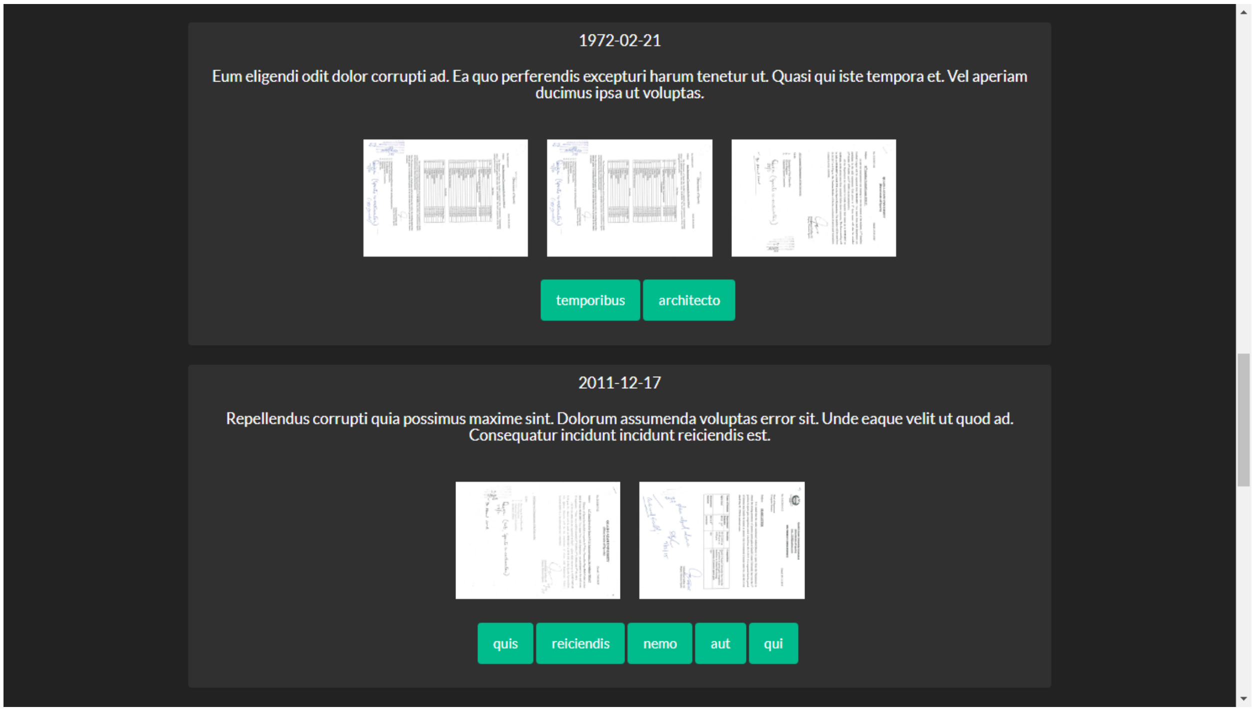
Sports Notification



2016-01-01

Add on notice board

Go to Timeline

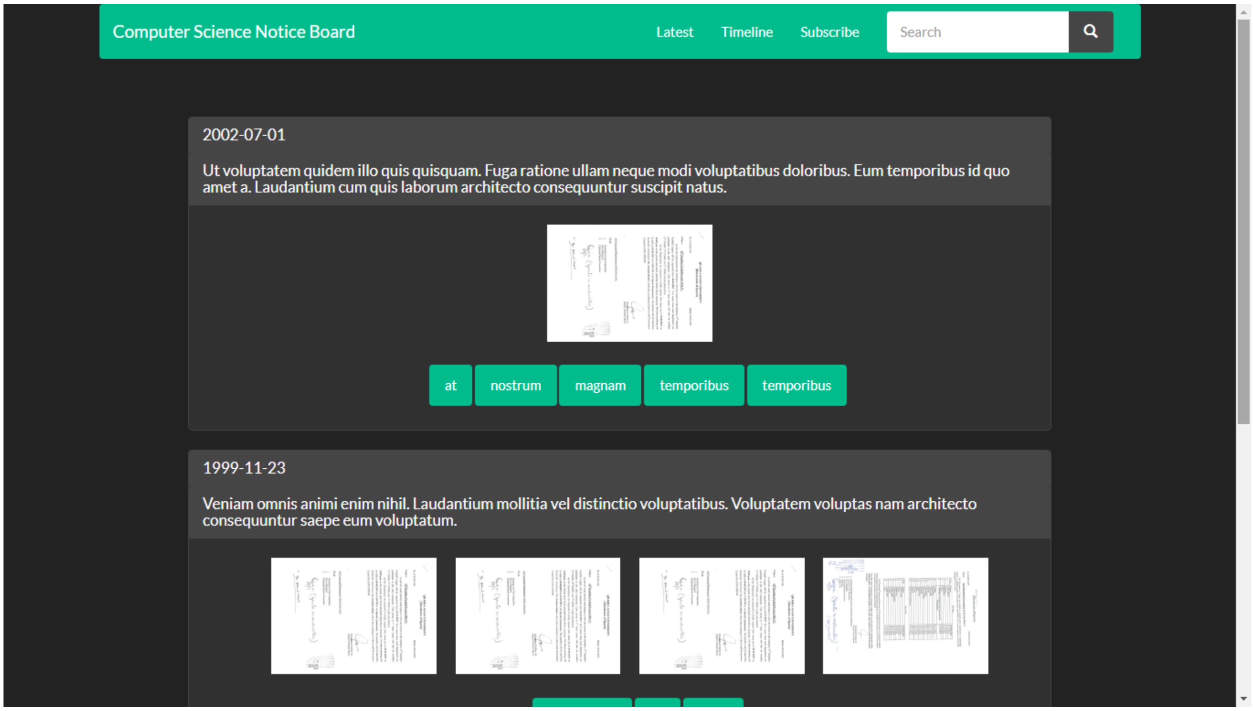


1972-02-21

Eum eligendi odit dolor corrupti ad. Ea quo perferendis excepturi harum tenetur ut. Quasi qui iste tempora et. Vel aperiam ducimus ipsa ut voluptas.

2011-12-17

Repellendus corrupti quia possimus maxime sint. Dolorum assumenda voluptas error sit. Unde eaque velit ut quod ad. Consequatur incidunt incidunt reiciendis est.

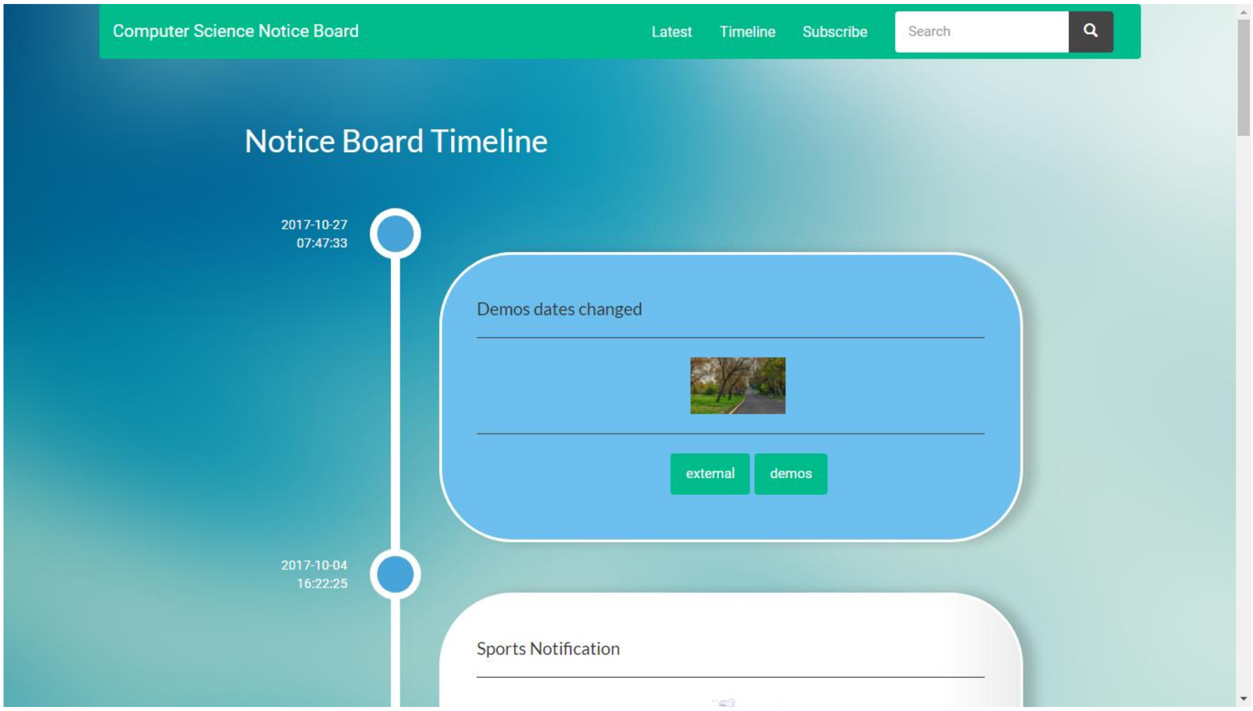


2002-07-01

Ut voluptatem quidem illo quis quisquam. Fuga ratione ullam neque modi voluptatibus doloribus. Eum temporibus id quo amet a. Laudantium cum quis laborum architecto consequuntur suscipit natus.

1999-11-23

Veniam omnis animi enim nihil. Laudantium mollitia vel distinctio voluptatibus. Voluptatem voluptas nam architecto consequuntur saepe eum voluptatum.



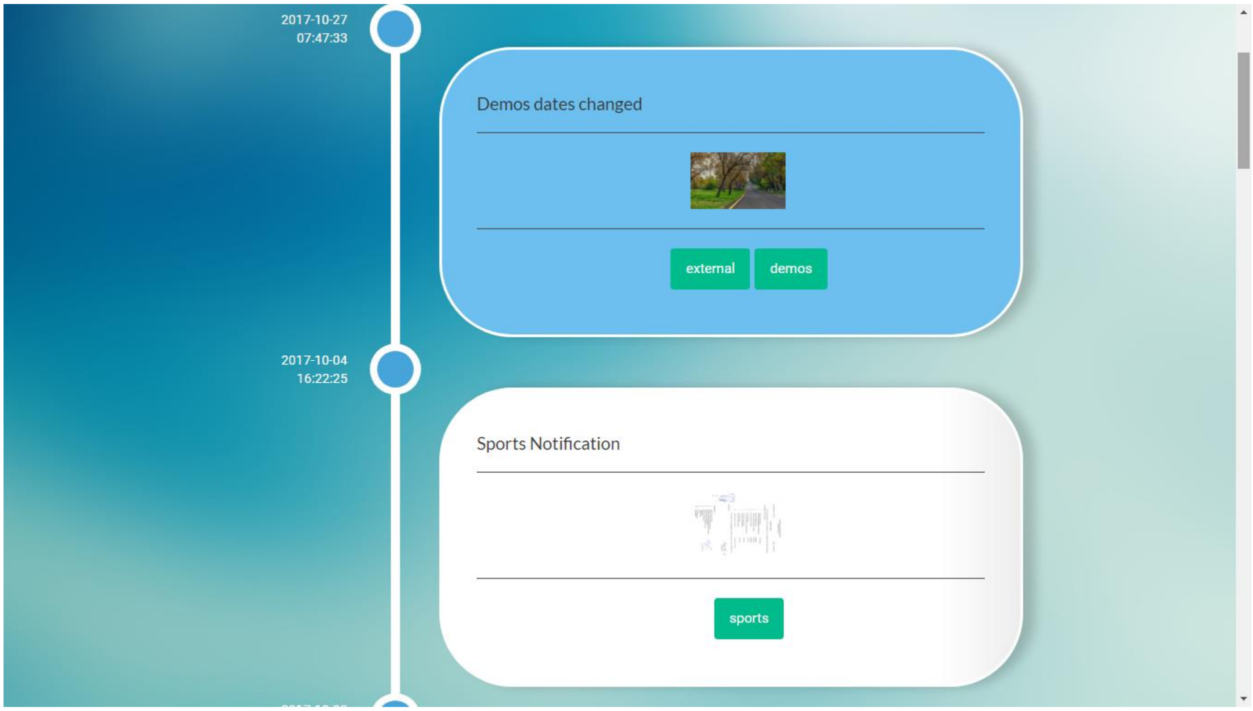
Notice Board Timeline

2017-10-27 07:47:33

Demos dates changed

2017-10-04 16:22:25

Sports Notification

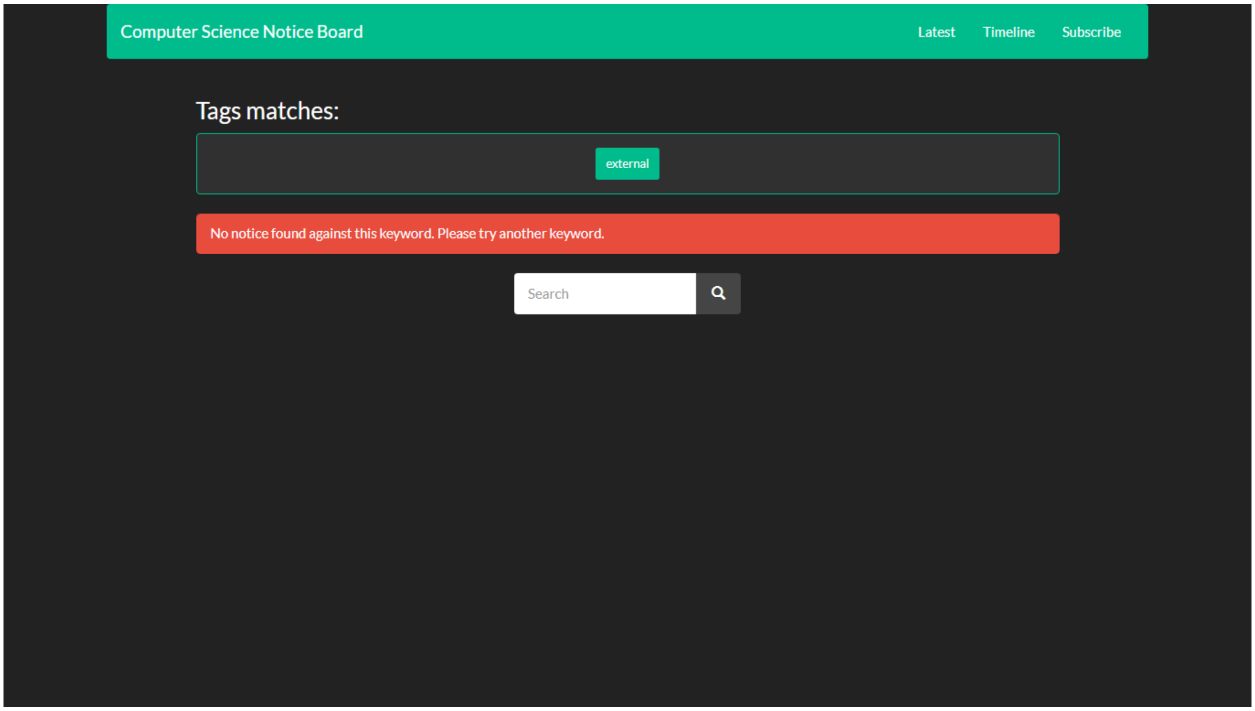


2017-10-27
07:47:33

Demos dates changed

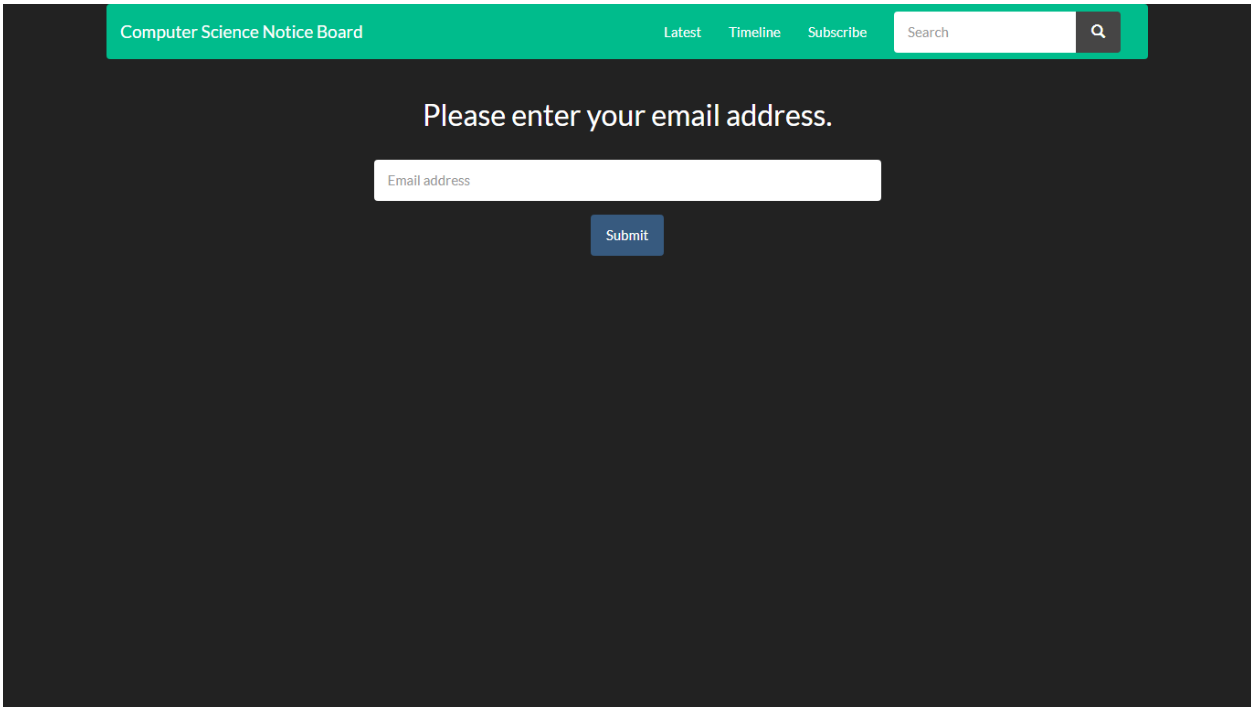
2017-10-04
16:22:25

Sports Notification

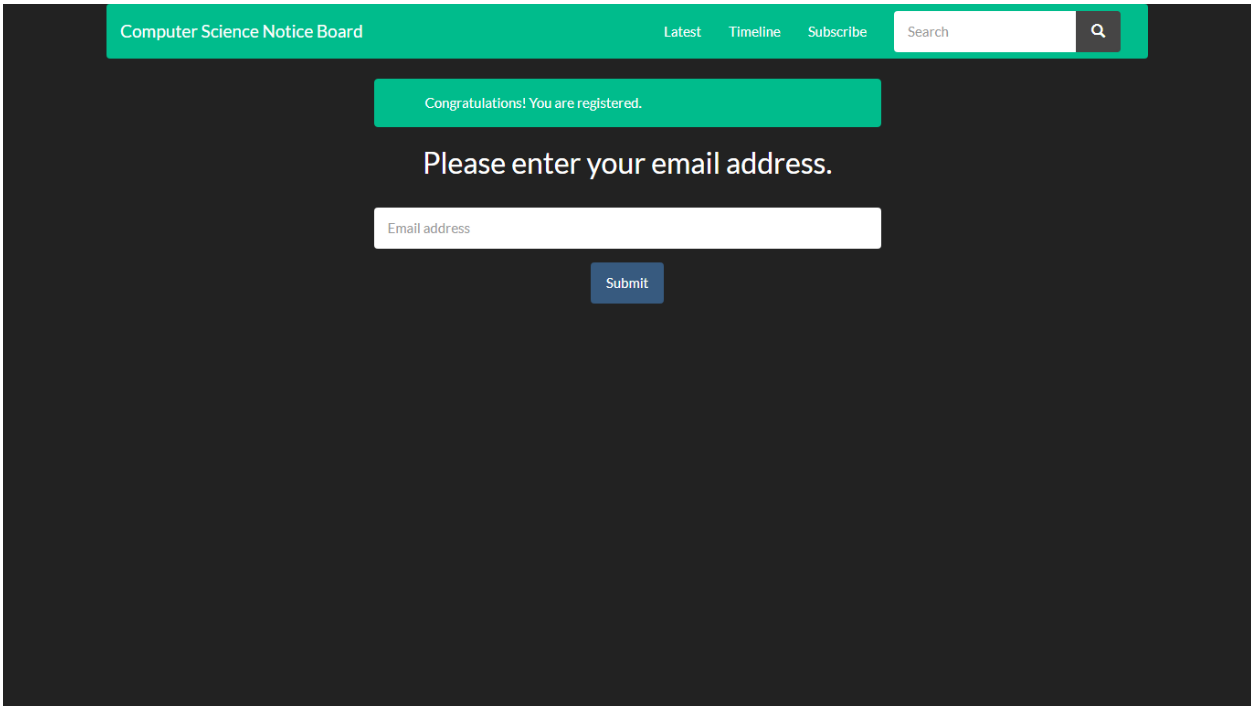


Tags matches:

No notice found against this keyword. Please try another keyword.



Please enter your email address.



Congratulations! You are registered.

Please enter your email address.

References

[1] https://www.agilealliance.org/agile101/

[2] Software Engineering, A practitioner’s approach, 7th Edition by Roger S. Pressman

[3] http://www.dictionary.com/browse/archive

[4] http://www2.archivists.org/sites/all/files/UsingArchives_Final.pdf

[5] Software Engineering, 3rd Edition, by Frank Tsui, Onrlando Karam, Barbara Bernal

[6] http://softwaretestingfundamentals.com/test-case/

[7]http://www.msqaa.org/Best_Practices/Quality_Control/FunctionalSystemTestingTechniqueCategories.pdf