Software Requirements Specification

# for

**SR Portal**

**Course:** CS253

**Course Instructor:** Prof. Amey Karkare

Prepared by

**Group:** 8

|  |  |  |
| --- | --- | --- |
| ANCHIT | 180085 | [anchit@iitk.ac.in](mailto:anchit@iitk.ac.in) |
| BHUVNESH | 180200 | [bamoriya@iitk.ac.in](mailto:bamoriya@iitk.ac.in) |
| NISHTHA | 180489 | [nishthaa@iitk.ac.in](mailto:nishthaa@iitk.ac.in) |
| SHREYA | 180735 | shreyak@iitk.ac.in |

Version: 1.1 Date: dd/mm/2020

Table of Contents

Table of Contents I

Revisions ii

1 Introduction 1

1.1 Document purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References 1

2 Overall Description 1

2.1 Product Overview 1

2.2 Product Functionality 2

2.3 Design and Implementation Constraints 3

2.4 Assumptions and Dependencies 4

3 Specific Requirements 4

3.1 External Interface Requirements 4

3.2 Functional Requirements 4

3.3 Use Case Model 6

4 Other Non-functional Requirements 13

4.1 Performance Requirements 13

4.2 Safety and Security Requirements 14

4.3 Software Quality Attributes 14

5 Other Requirements 15

Appendix A – Data Dictionary 15

Appendix B - Group Log 16

# 

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.1 | Anchit  Bhuvnesh  Nishtha  Shreya | This is the first release of the software. | xx/xx/2020 |

# Introduction

## 1.1 Purpose

This SRS describes the software functional and non-functional requirements for <project>. The purpose of this web application is to build an online system for accepting PhD and MTech applications and checking the validity of the data on the application with the corresponding data in the csv table as collected from Pingala.

## 1.2 Scope

The purpose of this project is to ease the process of checking and verifying the applications every year and to create a convenient and easy-to-use web-based application to reduce man-handling of nearly two thousand files in merely days. The system is based on a relational database with its application correcting and storing functions and a hierarchical login system with maximum powers vested in the hands of the admin. The database will support the data on csv file and an online file system will store the collected pdf applications. Above all, we hope to provide a comfortable and interactive user experience and optimum time utilization.

## 1.3 Intended Audience and Reading Suggestions

This project is built for college purposes and is restricted within the college premises. It’s mainly intended for the staff members to verify and check the data of the PhD and MTech applications. The major client is therefore the professors and assigned Teaching Assistants (TAs). It has been implemented under the guidance of college professors.

This SRS contains a user level description of the project in section 2, along with a detailed list of prioritized requirements in section 3 and a brief description of the change management process thereafter.

## 1.4 Definitions, Acronyms and Abbreviations

Refer to Appendix A - Glossary

## 1.5 Document Conventions

## No document conventions are being used at this time.

## 1.6 References

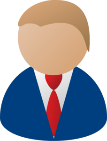
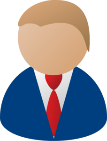
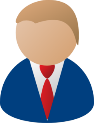
None

# Overall Description

## 2.1 Product Overview

The assessment and verification of the various Ph.D. and MTech applicants is time consuming and nearly two thousand applicants apply each year whose files are manually handled. Therefore, we have come up with a platform just to serve the purpose. <project> uses a hierarchical login system in which maximum powers are in the hands of the Admin and the TA’s are allotted bunch of files and corresponding information in the csv table and they are given the task of verifying those allotted files. No unauthorized user is allowed to work on the portal and people have powers depending on the role assigned to them i. e. Admin, TA, or Manager. The TAs’ can submit the corrections (if any) to the manager or admin who can then make necessary changes on the csv file. The following diagram in Figure 2.2 gives a very basic overview of how the different people relate in the system and the very basic roles played by them.

**Uploads corrections in csv**



**Admin**

Assign registered user as TA

Assign registered user as manager

**Manager**

**TA**

Allot pdf files

Suggest errors in case of discrepancy

Verify files sent by TAs

Note: the admin may also assign people as admin and there may be more than one admin in the system

**Figure 2.1**

*Basic features and how the relate*

## 2.2 Product Functionality

The major features of the software include:

* Stores pdf files in an online file system and csv table in database
* Allows admin to delete accounts and approve registered people as TAs, managers or admins
* People can register but no person can work unless his account is activated by admin
* Admin can also internally create accounts for others
* Both admin and manager:
  + Have a dashboard to review verified applications
  + Can allot files to TAs
  + Can make announcements for everyone in the portal
  + Can verify the corrections submitted by TAs and make corrections in csv files

*(Note: although both can do, it has to be done by either one of them and not by both for every file)*

* TAs’ workspace displays csv data of the allotted files and the corresponding pdf files
* TAs can submit errors (*if discrepancy found between pdf and csv*) to manager/admin
* Database stores the errors submitted and data of registered users

The following context diagram in Figure 2.2 illustrates the external entities and system interfaces.

Needs approval of admin to work

Work as manager, TA or admin

**Project Name**

**Admin**

**Online File System**

**People**

**Database**

**TA**

**Manager**

Register on portal

Assigned by admin

Allot pdf files to TAs

Verify/edit files sent by TAs

make corrections in csv

Delete/activate accounts of people

Approve as admin, manager or TA

Verify/edit files sent by TAs

Make corrections in csv

Allot pdf files to TAs

Manage allotted applications

Submit errors (if any)

Needs admin approval for login

Stores pdf files of applicants

Data of applicants (as collected from Pingala)

Data of registered users

Error messages provided by TAs

Announcements made by admin

Make announcements

Make announcements

**Figure 2.2**

*Context diagram for <project>*

## 2.3 Design and implementation constraints

|  |  |
| --- | --- |
| **Constraint** | **Description** |
| C-1: | Since the code is SQLite compatible, so it won’t work if you are using any other database to access the software. |
| C-2: | Use SQLAlchemy to access SQLite database. |
| C-3: | Use flask framework to access the software. |
| C-4: | The UI is not compatible with Internet Explorer and all the features won’t work on that web browser. |

## 2.4 Assumptions and dependencies

|  |  |
| --- | --- |
| **Assumption** | **Description** |
| AS-1: | It’s hard to judge the usability withouttesting the product in the field with the end clients. (*For instance: The space for storing files online will depend on our purchase*.) |
| AS-2: | System assumes that the parameters it takes as input are valid. |
| AS-3: | System assumes that the columns provided in the csv table are same the sample csv given. (*as they are coded accordingly*) |
| AS-4: | System assumes that only pdf files are uploaded in the portal. |
| AS-5: | System assumes that the names on the pdf files are same as the application number else it won’t open the correct files. |
| AS-6: | System assumes that users have JavaScript enabled. |

# Specific Requirements

## External Interface Requirements

* + 1. **User Interfaces**

UI-1: Web application shall permit complete navigation and selection using the keyboard alone, in addition to using mouse and keyboard combinations.

UI-2: However, if the application is run on the web in mobile or tablet, the pdf files won’t show in the workspace of TAs.

* + 1. **Hardware Interfaces**

To use input devices like touch screen monitor, mouse, keyboard etc we need various hardware interfaces.

## Functional Requirements

The functional requirements describe the functionality that the system is supposed to perform. It depends on the type of software, expected users and the type of system where the software is used. The functional requirements include the technical details, data manipulation, data processing, data integration, security requirements, performance, data migration, and conversion. Functional requirements for the Submit web application are listed below in Table. Functional requirements are labelled as FR.

|  |  |
| --- | --- |
| **Function** | **Description** |
| FR-1: | The system shall allot different powers to a person working depending on whether he is TA, admin or manager. |
| FR-2: | The system shall allow anyone to register and make an account on the portal. |
| **Functional Requirements for admin** | |
| FR-3: | The system shall allow the admin to approve the registered user as admin, manager or TA. (*Note: there may be more than one admin*) |
| FR-4: | The system shall allow only the account approved by admin to work else the user won’t be able to access any feature. |
| FR-5: | The system shall allow the admin to internally create and delete account for users. |
| FR-6: | The system shall allow the admin to view details of all registered users except their passwords and allows admin to change password of a user in case needed. |
| FR-7: | The system shall allow the admin to view all files in workspace and to view the files sent by TAs in the dashboard. |
| FR-8: | The system shall allow admin to verify the files sent by TAs and upload the corresponding corrections in the csv table. |
| FR-9: | The system shall allow the admin to upload csv table and pdf files in the portal. |
| FR-10: | The system shall allow the admin to allot the pdf files equally to TAs for verification. |
| FR-11: | The system shall allow the admin to make announcements for everyone in the portal. |
| **Functional Requirements for manager** | |
| FR-12: | The system shall allow the manager to view all files in workspace and to view the files sent by TAs in the dashboard. |
| FR-13: | The system shall allow manager to verify the files sent by TAs and upload the corresponding corrections in the csv table. |
| FR-14: | The system shall allow the manager to upload csv table and pdf files in the portal. |
| FR-15: | The system shall allow the manager to allot the pdf files equally to TAs for verification. |
| FR-16: | The system shall allow the manager to make announcements for everyone in the portal. |
| **Functional Requirements for TA** | |
| FR-17: | The system shall allow TAs to manage the applications allotted to them. |
| FR-18: | The system shall allow the TA to view only the data of allotted applicants and their corresponding pdf files. |
| FR-19: | The system shall allow the TA to view csv table columns and corresponding pdf, on clicking the application number, in their workspace. |
| FR-20: | The system shall allow the TA to suggest any cases of discrepancy in the comments text area and submit the suggestions to manager and admin. |
| **Other General Functional Requirements** | |
| FR-21: | The system shall store the data of csv table in the database and the pdf files in an online file system. |
| FR-22: | The system shall store the announcements made by manager and the comments submitted by TAs in the database as well. |
| FR-23: | The system shall store the data of the registered users in the database. |
| FR-24: | The system provides a dashboard for admin and manager but not for TAs. |
| FR-25: | The system shall allow sending a password recovery mail in case password is lost but the link works for thirty minutes only. |
| FR-26: | The system shall allot the pdf files automatically and equally among all TAs. |
| FR-27: | The system shall reset whole database accessible to admin only to be used once admission process is over. |
| FR-28: | Once file has been verified “View” button changes to “Verified” to alert another manager and admin that someone has already verified that file. |

## Use Case Model

This section gives the written description of the list of actions or event steps of how the user will perform tasks, typically the interactions between a role and a system, to achieve a goal. It outlines a user’s point of view and a system’s behaviour as it responds to a request. In our system, there are three types of users: Administrator, Manager and Teaching Assistant. The following diagram in Figure 3.1 gives the use case model for SR Portal.

* + 1. **UC 01 - Registration**

**Author** –Nishtha

**Purpose** - This use case describes the process of registering in the portal.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - High

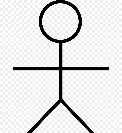
**Preconditions** - There are no preconditions to this use case

**Post conditions** - After successful registration, you need approval of the admin to work after which actors will be able to do work according to the role assigned.

**Actors** – Any person can register thus actor may be admin, manager or teaching assistant.

**Flow of Events**

1. **Basic Flow** - This use case allows a person to register to the portal using username, email address and password and then he waits for the approval of admin to do further functioning.
2. **Alternative Flow** - In case of invalid email address or an already existing username, it gives and error and does not register. Also in case the password and confirm password field don’t match, the registration does not occur.



Registration

View/manage allotted files & data

Submit corrections to manager/admin

Approval of users

Create/delete accounts for others

Access details of all users

Change passwords for others

Make announcements

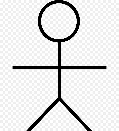
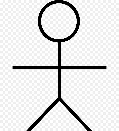
Verify files sent by TAs

Upload corrections to csv file

Upload csv file to portal

Upload pdf files to portal

Allotting files among TAs



TA

Manager

Admin

<<include>>>

<<include>>>

<<include>>>

<<include>>>

<<extend>>>

Login

<<extend>>>

**Figure 3.1**

*Use case diagram and use cases for <project>*

* + 1. **UC 02 - Login**

**Author** –Nishtha

**Purpose** - This use case describes the process of logging in into the portal.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - High

**Preconditions** - The user must be registered to the portal.

**Post conditions** - After logging in only the user may be able to access the system. But for anyone to work his registration must be approved by the admin.

**Actors** – Any registered person can login thus actor may be admin, manager or teaching assistant.

**Flow of Events**

1. **Basic Flow** - This use case allows you to access the portal by logging in using email and password. After logging in, you may get to your workspace and other features depending on the role assigned.
2. **Alternative Flow** -You may be asked to login again in case of incorrect email or password.
3. **Exceptions** - You may not be able to login in case not registered or not approved by admin.
   * 1. **UC 03 - Creating and deleting accounts**

**Author** –Anchit

**Purpose** - This use case describes creating and deleting accounts of users by admin.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Low

**Preconditions** - The user should be an admin and he must be logged in.

**Post conditions** - The user account is added or delete as per admin requirement.

**Actors** – Admin

**Flow of Events**

1. **Basic Flow** - The admin logs in and opens the users table where he can add or delete accounts for users.
2. **Exceptions** - An admin can (or) can’t delete account of other admins. (if can’t to yd point hata dena)
   * 1. **UC 04 - Change password for other users**

**Author** –Nishtha

**Purpose** - This use case describes changing password of users by admin.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Low

**Preconditions** - The user to perform this shall be admin only and he must be logged in.

**Post conditions** - The password of the specific account is changed.

**Actors** – Admin

**Flow of Events**

1. **Basic Flow** - The admin logs in and opens the users table where he can change the password for the required user.
2. **Alternative Flow** - If an error occurs due to any reason, the user may himself change password via password reset mail.
   * 1. **UC 05 - Approval of users**

**Author** –Bhuvnesh

**Purpose** - This use case describes the process of approving users.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - High

**Preconditions** - The admin should be registered and logged in. The users should be registered to the portal.

**Post conditions** - The approved users can log in and work on the system as per the role assigned.

**Actors** – Admin

**Flow of Events**

1. **Basic Flow** - The admin goes to the users table where he finds all the users and approves and assigns role to the required users.

**Includes -** It includes UC 01 - Registration. This approval is only done after the users have registered to portal.

* + 1. **UC 06 - Access details of users**

**Author** –Bhuvnesh

**Purpose** - This use case describes the accessing of details of all users.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Medium

**Preconditions** - The admin shall be logged in.

**Post conditions** - Admin has access to details of all registered users.

**Actors** – Admin

**Flow of Events**

1. **Basic Flow** - On logging in and going to the users table, the admin can access the details of all users registered on the system.
2. **Exceptions** - Except that admin is not allowed to view passwords of the users which are hashed.
   * 1. **UC 07 - Make announcements**

**Author** –Bhuvnesh

**Purpose** - This use case describes the process of making announcements in the portal.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Low

**Preconditions** - The actor should be manager/admin and should be logged in.

**Post conditions** - There is an announcement for everyone in the system.

**Actors** – Admin and manager.

**Flow of Events**

1. **Basic Flow** - The actor may log in and go to the announcements page and post an amercement for all.
   * 1. **UC 08 - Verify files sent by TAs**

**Author** –Bhuvnesh

**Purpose** - This use case describes the process of verification of files sent by TAs.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Medium

**Preconditions** - The files to be verified shall be sent by TAs and the actor shall be logged in.

**Post conditions** - The verified files are correct and the data corresponds in both csv and pdf and correction (if any) therefore can be uploaded in csv.

**Actors** – Admin and manager

**Flow of Events**

1. **Basic Flow** - The actor may see upon the corrections sent by TAs and check if they are correct and mark the file verified in the dashboard which removes it from his dashboard and it is verified.
2. **Alternative Flow** - The actor can view all files in workspace and he may verify files himself in case he does not receive files from TAs.

**Includes -** It includes UC 14 - Submit suggestions to manager/admin. The manager/admin shall verify that the corrections submitted are correct.

* + 1. **UC 09 - Upload corrections to csv**

**Author** –Anchit

**Purpose** - This use case describes the process of uploading corrections to the csv file.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - High

**Preconditions** - The files are verified by TA and admin/manager and the actor is logged in. The csv should be uploaded to the portal.

**Post conditions** - The csv file has corrected information of applicants.

**Actors** – Admin and manager.

**Flow of Events**

1. **Basic Flow** - The manager/admin shall log in and upload corrections to the csv table.

**Includes -** It includes UC 12 - Upload csv file to portal.

* + 1. **UC 10 - Allot files among TAs**

**Author** –Anchit

**Purpose** - This use case describes the process of allotting pdf files among TAs.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - High

**Preconditions** - The pdf and csv files shall be uploaded to the portal and the actor may be logged in.

**Post conditions** - The files are automatically allotted equally among all the TAs.

**Actors** – Admin and manager

**Flow of Events**

1. **Basic Flow** - The actor should log in and click the allocate button to allot the pdf files and corresponding csv information among the TAs.
   * 1. **UC 11 - Upload pdf files to the portal**

**Author** –Anchit

**Purpose** - This use case describes the process of uploading pdf files to the portal.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Medium

**Preconditions** - The pdf files should be available with the same name as the application number.

**Post conditions** - The pdf files are uploaded online in the system database.

**Actors** – Admin and manager

**Flow of Events**

1. **Basic Flow** - The actor may log in and upload the pdf files for the PhD and MTech applicants by clicking on the upload button.
   * 1. **UC 12 - Upload csv file to the portal**

**Author** –Shreya

**Purpose** - This use case describes the process of uploading the csv file to the portal.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Medium

**Preconditions** - The csv file should be available with the same number of columns as the sample csv.

**Post conditions** - The csv data is uploaded in the system database.

**Actors** – Admin and manager

**Flow of Events**

1. **Basic Flow** - The actor may log in and upload the pdf files for the PhD and MTech applicants by clicking on the upload button.
   * 1. **UC 13 - View/manage allotted files and data**

**Author** –Shreya

**Purpose** - This use case describes the process of managing and checking the allotted data by the TAs.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - High

**Preconditions** - The files should be uploaded and allocated and the TAs should be assigned.

**Post conditions** - The TAs have access only to allocated data and files of applicants.

**Actors** – Teaching Assistants

**Extends** – It is an extension of UC 14 - Submit suggestions to admin/manager

**Flow of Events**

1. **Basic Flow** - The TAs may log in and open their workspace to find the data on one side and corresponding pdf files which he may view in the other half of the page for verifying.
2. **Exceptions** - If the files are not allotted the admin/manager may still have access to those but not the TAs.

**Includes -** It includes UC 10 - Allot files among TAs

* + 1. **UC 14 - Submit corrections to admin/manager**

**Author** –Shreya

**Purpose** - This use case describes the process of submitting corrections to the admin/manager by TA.

**Requirements Traceability** – Identify all requirements traced to this use case

**Priority** - Medium

**Preconditions** - The files should be allotted among TAs and the TA should have thoroughly verified the file.

**Post conditions** -The errors and cases of discrepancies are sent to admin/manager for reviewing.

**Actors** –Teaching Assistant

**Extends** – It is an extension of UC 09 - Uploading corrections to csv

**Flow of Events**

1. **Basic Flow** - The data in the csv columns and the submitted pdf is checked by the TA in the workspace and in case of discrepancy among the two, the corrections are written in the text area and submitted to admin/ manager.
2. **Exceptions** - In case files are not allotted, the TA does not have access but the admin/ manager can still do the task.

# Other Non-functional Requirements

The non-functional requirements describe the specific behaviour or functions that the system is supposed to perform. It can also be referred to as the quality attributes of the system architecture. Response time, scalability, reliability, maintainability, usability, etc., are some of the examples of non-functional requirements.

## Performance Requirements

PR-1: System assumes that the check-box “Allow pages to choose their own fonts, instead of your selection above” is marked in case you are working in Firefox else UI won’t work properly.

PR-2: All Web pages generated by the system shall be fully downloadable in no more than 10 seconds over a 200KBps modem connection.

## Safety and Security Requirements

SE-1: Users shall be required to log in to the system for all operations.

SE-2: The system shall permit only staff members who are authorized by the administrators to create or edit brand.

SE-3: The system shall permit users who are not project managers or administrators i.e. TAs to view only their workspace, not the information of other users.

SE-4: The system shall permit managers to view everything that admin can except the user information page which is accessible for admin only.

SE-5: The system shall allow the password to be recovered via a password recovery mail that is valid for thirty minutes from the time the password is sent.

## Software Quality Attributes

### **4.3.1 Availability**

The system shall have a 24/7 availability. The database has rollback segments, and a centralized backup can be performed to act upon a system recovery from a failure.

### **4.3.2 Security**

The system is quite secure and should allow only authorized access to users.

The system uses password hashing to verify the integrity of user’s password sent during login, against the 32-bit stored hash.

Actual password of user is never stored and with password hashing it is practically impossible for anyone to steal other’s password.

### **4.3.3 Portability**

This software is a web-based application; thus, it should work on other platforms and operating systems. But the UI may not be fully functional in all browsers thus there may occur some issues with portability.

# Appendix A - Data Dictionary

|  |  |
| --- | --- |
| **Abbreviation** | **Abbreviation Definition** |
| AS | Assumption |
| Admin | Administrator |
| C | Constraint |
| csv | Comma-separated values |
| FR | Functional Requirements |
| KBps | Kilobyte per second |
| MTech | Masters in Technology |
| OTP | One-time password |
| Pdf | Portable document format |
| PhD | Doctorate of Philosophy |
| PR | Performance Requirement |
| SE | Security Requirement |
| SQL | Structured Query Language |
| TA | Teaching Assistant |
| UI | User Interface |

# Appendix B - Group Log

**Meeting 1**

Date: 20/02/20

Duration: 1.5hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Discussed about various features of the project and how to implement them | Anchit, Bhuvnesh, Nishtha, Shreya |
| Made rough sketch of UI and decided the framework | Anchit, Bhuvnesh, Nishtha, Shreya |
| Divided workload and tasks | Anchit, Bhuvnesh, Nishtha, Shreya |

**Meeting 2**

Date: 10/02/20

Duration: 1.5hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Discussed where to start from, which database to use and on which framework to work on | Anchit, Bhuvnesh, Nishtha, Shreya |
| Watched online tutorial and videos on Flask and sqlite3 | Anchit, Bhuvnesh, Shreya, Nishtha |

**Meeting 3**

Date: 12/02/20

Duration: 1.5hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Created signup page so that every should have distinct username and email id | Anchit, Bhuvnesh |
| Created login page so that they can login after registering | Shreya, Nishtha |
| Discussed on admin manager and TA writes and discussed how to move further | Anchit, Shreya, Bhuvnesh, Nishtha |

**Meeting 4**

Date: 23/02/20

Duration: 4hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Database tables User, Post and connected them to flask routes using Sqlalchemy | Anchit, Bhuvnesh |
| Worked on code to allocate PhD files to Tutors. | Anchit, Bhuvnesh, Shreya, Nishtha |
| Implemented role-based access control and assigned different powers to different users | Anchit |
| Created new route “/people” to view all users and their profile | Nishtha, Shreya |

**Meeting 5**

Date: 24/02/20

Duration: 3hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Created a non-dynamic PhD database table and created new routes for viewing the uploaded csv. | Anchit, Shreya |
| Designed and implemented new html templates | Anchit, Bhuvnesh, Nishtha |
| Implemented allocation code and connected it | Bhuvnesh |
| Worked on database error while updating PhD table new column on csv | Anchit, Bhuvnesh, Shreya |

**Meeting 6**

Date: 25/02/20

Duration: 4.5hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Created new route to upload pdf files of applicants | Anchit, Bhuvnesh |
| Tested new method on creating database directly from csv file | Anchit, Bhuvnesh, Shreya |
| Updated the workspace section to view pdf of respective applicants on clicking their Application reference no. | Anchit, Bhuvnesh, Nishtha |
| Added option to manually allocate TA | Anchit, Bhuvnesh |

**Meeting 7**

Date: 26/02/20

Duration: 3hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Created dashboard section to view all submissions and stats related to files, submissions and verifications | Anchit |
| Secured the routes such that TAs can view only those files which are allocated to them. | Anchit, Bhuvnesh |
| Added new feature of adding any error found during checking by TAs | Anchit, Bhuvnesh |
| Worked on database error | Shreya |

**Meeting 8**

Date: 27/02/20

Duration: 3.5hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Implemented forgot password in which forgot password mail is send to mail id and then he can change it | Anchit, Bhuvnesh |
| Designed and implemented new html templates for forgot password | Anchit, Bhuvnesh, Nishtha |
| Finding bugs while uploading csv file on Database | Anchit, Shreya |
| Started working on SRS. Looked over different SRS template and finalized it | Shreya, Nishtha |

**Meeting 9**

Date: 28/02/20

Duration: 4hr

|  |  |
| --- | --- |
| **Work** | **Contributors** |
| Implemented new method of creating dynamic database tables directly from csv using Pandas and python sqlite3. | Anchit |
| Updated the html files for TAs according to new database | Anchit, Bhuvnesh, Shreya |
| Updated allocation code to allocate same file to 2 different tutors for cross verification | Bhuvnesh |
| Implemented announcement section such that admin and manager can make any announcement and TAs can only view them | Anchit |
| Wrote code to reset the database tables and updated the SRS file. | Shreya, Nishtha |

**Meeting 10**

Date: 29/02/20

Duration: 3.5hr

|  |  |
| --- | --- |
| Work | Contributors |
| Created comment section so that TAs can write any mismatch between CSV and PhD files so that manager can update the CSV file | Anchit, Shreya |
| Created Violation table to record all events of unauthorized access to any route | Anchit |
| Redesigned the workspace section to view individual file on separate route | Anchit, Bhuvnesh |
| Worked on problem of database not updating on without restarting the server | Anchit, Bhuvnesh, Shreya, Nishtha |

**Meeting 11**

Date: 24/02/20

Duration: 3hr

|  |  |
| --- | --- |
| Work | Contributors |
| Wrote code for MTech table | Anchit, Bhuvnesh, Shreya |
| Updated the PhD table with new columns of Reject Reason | Anchit, Bhuvnesh, Nishtha |
| Worked on system wide notification system | Anchit, Bhuvnesh |
| Added feature such that manager can reject application if he founds that TA hasn’t checked properly and notify him with some message | Anchit |

**Meeting 12**

Date: 24/02/20

Duration: 3hr

|  |  |
| --- | --- |
| Work | Contributors |
| Created two different comment section for same file for each TA | Anchit, Bhuvnesh |
| Created the submit section so that he can submit the comment he added with the file | Anchit, Shreya, Bhuvnesh |
| Updated database and dashboard section such that only those files will appear on dashboard, which have been submitted by both TAs | Anchit |
| Bugs Finding | Shreya, Nishtha |
| Added feature of saving the updated database table as csv using Pandas | Anchit |