Final Report - CSCE 606
Software Engineering
PhD Annual Review

1. Project Summary

The PhD annual review system is a web application designed to streamline and enhance the efficiency of the annual PhD review process at Texas A&M University. Our application has 3 main stakeholder PhD students, faculty/reviewing committee members and administrators, the system provides a comprehensive platform for managing the annual PhD review cycle. Our client mentioned that the current system used for annual review is not user friendly and it is laborious to provide review to the student as faculty members have to switch between views in order to see the student documents. The client also wanted us to have a consolidated view for the administrators so that they can view if students have submitted all the documents, has faculty members provided their evaluation and what are the students final evaluation.

We have designed our application by taking the above requirements into consideration. The application empowers the PhD students to submit their documents and information easily. Faculty and committee members benefit from a centralized view of student information, enabling them to submit ratings, review comments and nominate students for rewards seamlessly. The administrator view, which consolidates all relevant information and statuses for each stakeholder, is a crucial aspect of the system. Administrators can use this view to check if students have provided all required documentation, examine committee member evaluations, and calculate final evaluation scores. This addresses the client's complaint about the current system's lack of a single view, allowing faculty members who previously had to switch between views to do their tasks more quickly. The reporting capabilities of the system enable administrators to generate comprehensive reports on student progress and results, allowing data-driven decision-making. The annual PhD review system benefits all stakeholders greatly. It streamlines the tracking of PhD candidates progress, provides a centralized and efficient platform for professors and committee members, and delivers useful information to administrators. This application not only improves the whole experience for PhD students, but it also helps to improve and ensure the quality of the university's PhD programs.

2. User Stories Description

○ Home page for the application

2.○.1 UI for Home page for the application

Points: 1 and Status: Completed
Created the home page for the web application so that the user can login as a student, faculty or as an administrator.

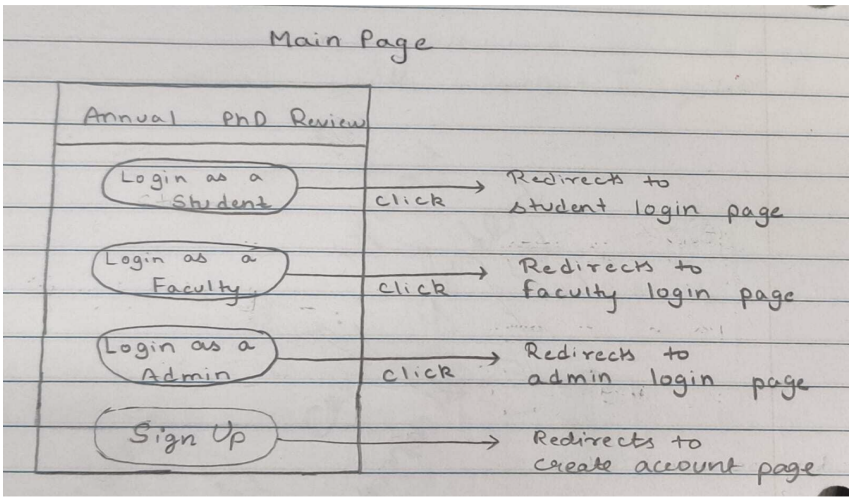

Figure 1: Mock UI for home page

2.○.2 Add styling to the home page for the application

Points: 1 and Status: Completed
Added styling to the webpage so that it looks more user friendly.

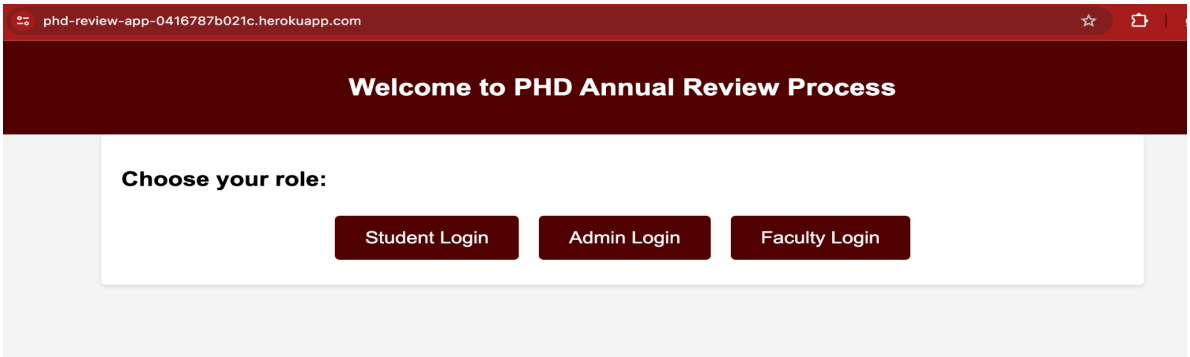

Figure 2: Actual UI for home page

- **Login and signup page for student**

2.○.1 Creating a login and signup page for the student

Points: 1 and Status: Completed
We created a login and signup page for the student where the student can enter their credentials and then sign in to submit their PhD annual review report and documents.

2.○.2 Adding constraints on the credentials

Points: 1 and Status: Completed
We added additional constraints on the user that the emailID must end with tamu.edu and the length of the password should be minimum 8 characters.

2.○.3 Use Bycrpt to store the encrypted password in the database

Points: 1 and Status: Completed
We used bycrpt gem to store the password in an encrypted format rather than storing it in plain text.

2.○.4 CSS Styling for login and signup page for the student

Points: 1 and Status: Completed

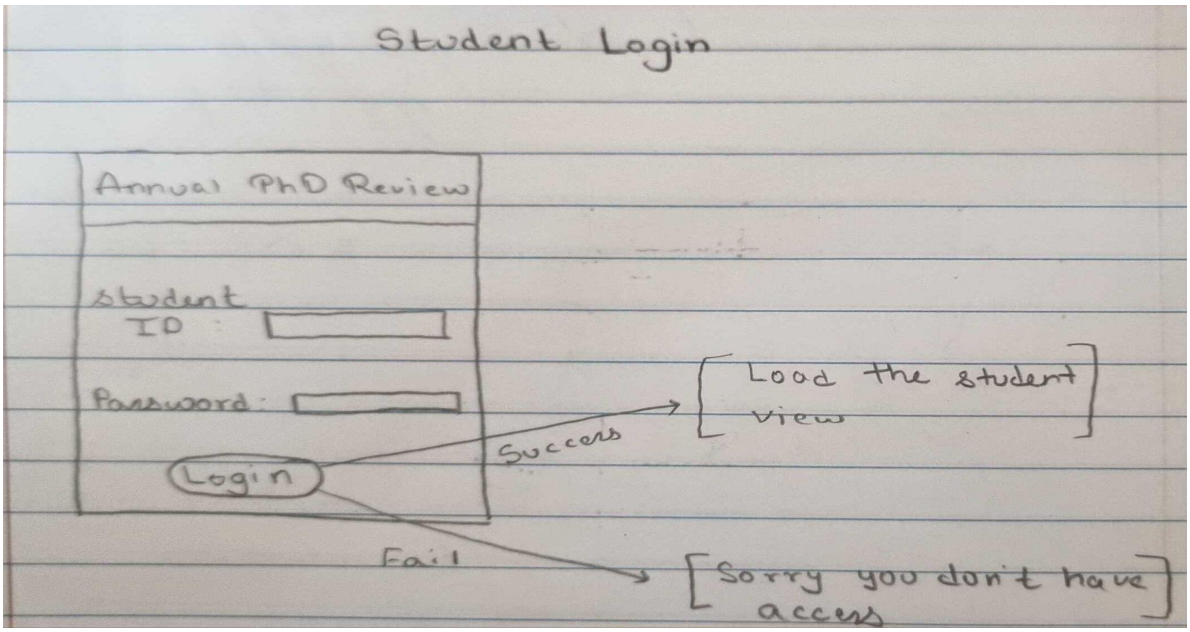

Figure 3: Mock UI for Student Login Page

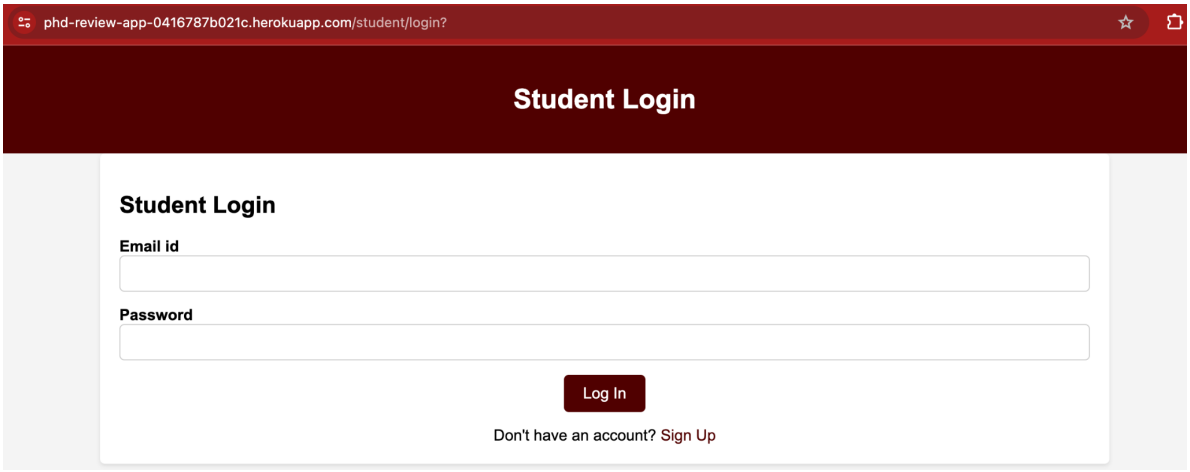

Figure 4: Actual UI for Student Login Page

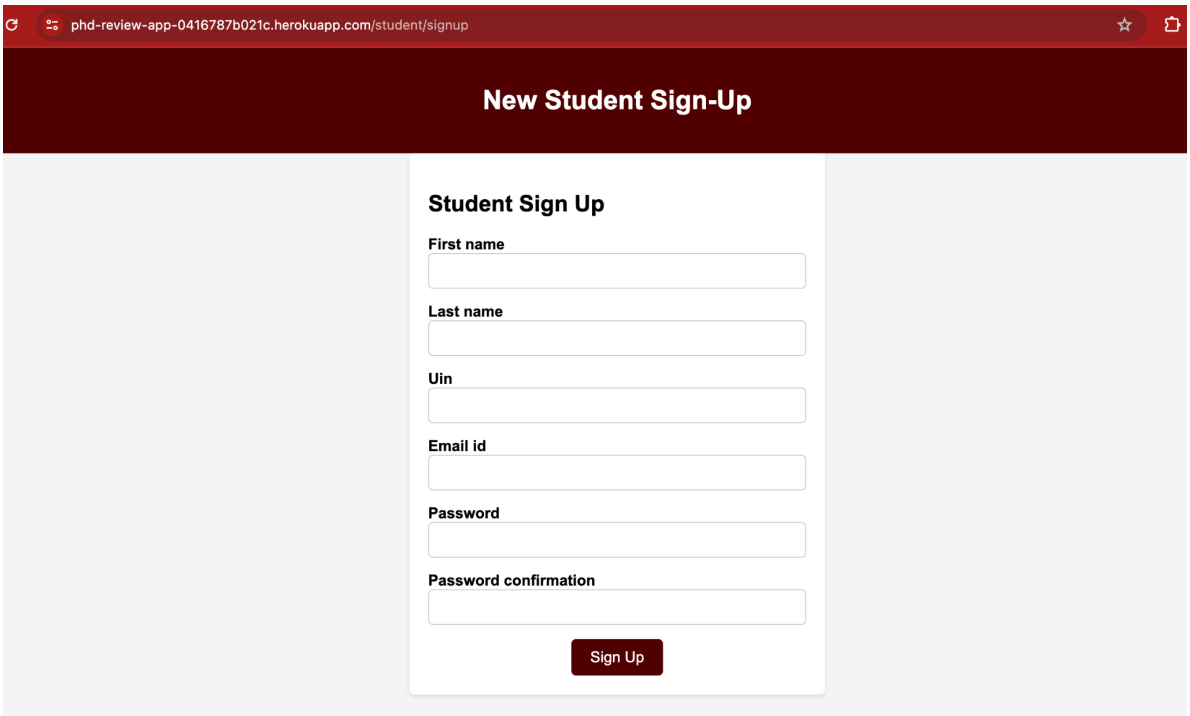

Figure 5: Actual UI for Student Sign up Page

- **Login page for faculty and administrator**

2.○.1 Creating a login for faculty and administrator

Points: 1 and Status: Completed
We created a login page for faculty and administrators so that they can login into the system and see student documents and provide their feedback.

2.○.2 Adding constraints on the credentials

Points: 1 and Status: Completed
We added additional constraints that the emailID must end with tamu.edu and the length of the password should be minimum 8 characters.

2.○.3 Use Bycrpt to store the encrypted password in the database

Points: 1 and Status: Completed
We used bycrpt gem to store the password in an encrypted format rather than storing it in plain text

2.○.4 CSS Styling for login page for faculty and administrator

Points: 1 and Status: Completed

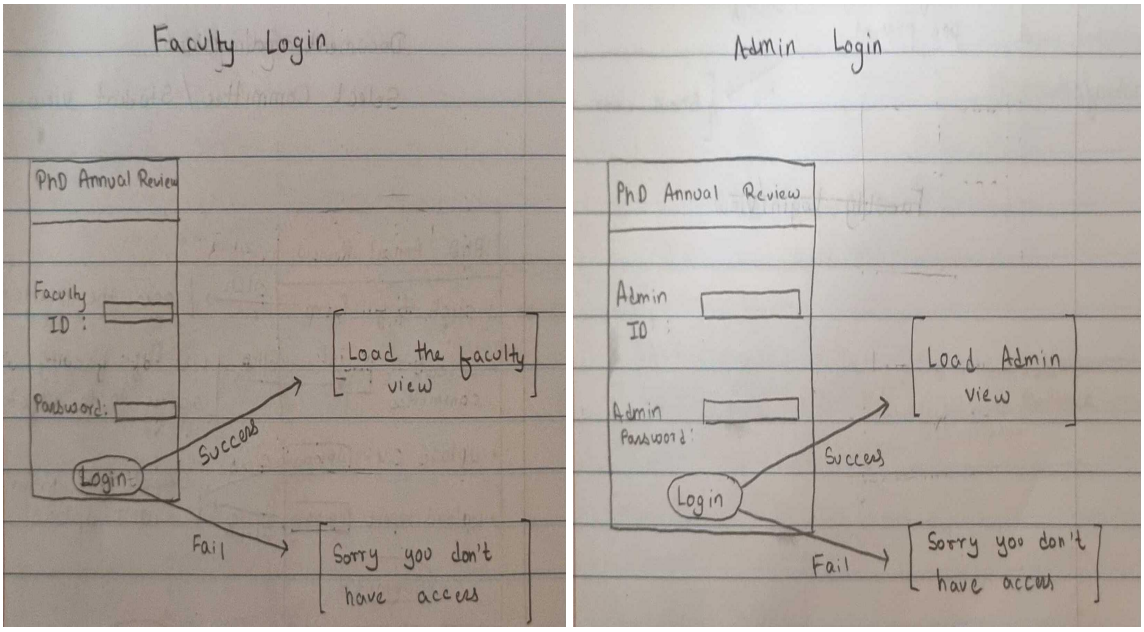

Figure 6: Mock UI for Faculty and admin login

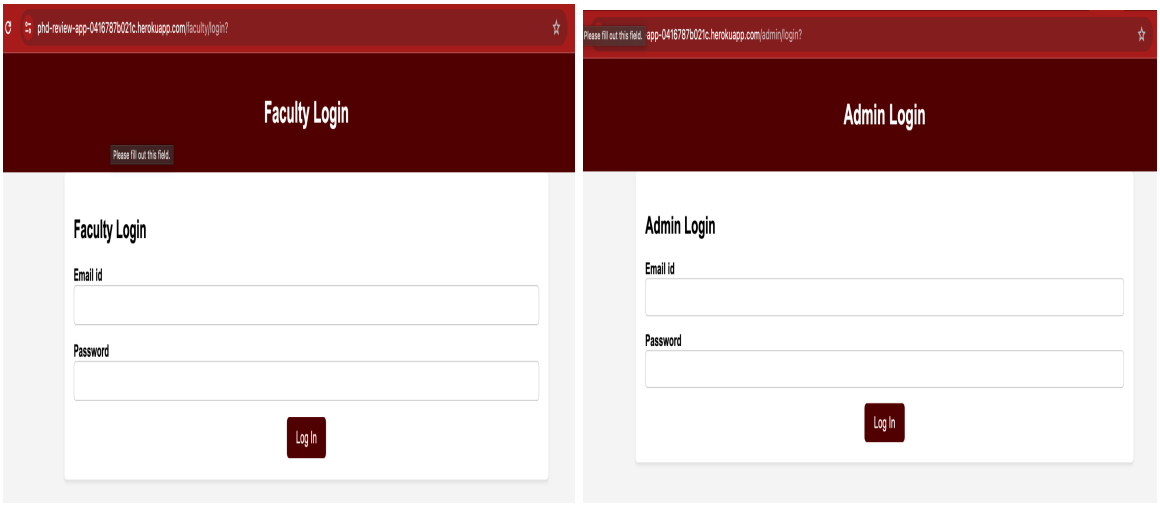

Figure 7: Actual UI for Faculty and admin login

- **Student view**

2.0.1 Created a student dashboard

Points: 1 and Status: Completed
We created a dashboard for students which had various options such as edit committee, document submission, view submission and view comments made by faculty.

2.0.2 Student document submission Page

Points: 4 and Status: Completed
Created a view for the student user where they have to fill a form and submit his documents so that faculty can review and provide feedback.

2.0.3 Student can select his committee members and chair

Points: 2 and Status: Completed
Students can select faculty members who will be a part of their review committee and also select the chair for the committee.

2.0.4 Student can view comments made by the committee members

Points: 2 and Status: Completed
Once the committee members provide a review and comment for the student, the student can view the comments they got from the faculty.

2.0.5 Student can view the documents submitted by them

Points: 1 and Status: Completed
Students can view the submissions they made after submitting the form and their

documents, so that if they submitted any info that is incorrect they can update their response. This also serves as confirmation that the students have submitted the documents successfully.

2.○.6 Student can update their password

Points: 2 and Status: Completed
Our client asked us to add this feature because if students are provided a default password by the university then they can update the password later.

2.○.7 CSS Styling for student view

Points: 2 and Status: Completed

2.○.8 Making some fields in the form conditionally mandatory

Points: 1 and Status: Completed
Our client asked us to make certain fields mandatory in the form based on the selection made by the student..

- **Faculty view**

2.○.1 Created a faculty dashboard

Points: 3 and Status: Completed
We created a dashboard for faculty which has a list of students for which the faculty need to provide a review. When the faculty member clicks on the student name they should see the response submitted by the student and they can provide rating, comments for the student, comment for other committee members and nominate the student for reward.

2.○.2 Faculty can view the assessment they submitted and also update the same

Points: 3 and Status: Completed
When a faculty member submits an assessment for the student then the student should move to the completed list. The faculty member can see the assessment they provided and also update the assessment if they want to.

2.○.3 Faculty can update their password

Points: 2 and Status: Completed
Our client asked us to add this feature because if faculty members are provided with a default password by the university then they can update the password later.

2.○.4 CSS Styling for faculty view

Points: 1 and Status: Completed

- **Admin view**

2.0.1 Created admin dashboard

Points: 2 and Status: Completed
Created an admin dashboard where admin can have a bird eye view of the system, the admin can see if the students are ready for review, which all committee members have provided the feedback, what is the overall evaluation of the student and if the student is nominated for reward for not.

2.0.2 Administrator can download the report

Points: 2 and Status: Completed
Added a download csv feature for the administrator so that the administrator can download the data onto his system and get a better understanding of the data.

2.0.3 CSS Styling for admin view

Points: 1 and Status: Completed

- **Bug fixes**

2.0.1 S3 timeout issue

Points: 3 and Status: Completed
We had a bug where the url generated by S3 will expire in 15 minutes. To fix this now we are fetching the object that is stored in the S3 and generating the URL on the fly.

2.0.2 Bug while filling the student document submission form

Points:1 and Status: Completed
We had a bug where if the student selected None as the option for support in last semester we were getting an error we fixed that and now the feature is working as expected.

- **Logic Development and Setups**

2.0.1 Create Class Diagram for the PhD Annual Review System

Points: 3 and Status: Completed
Following the requirements of the client, we designed the needed classes and the data fields in each. This was followed by creating connections between all the present classes.

2.0.2 Designing DB schema

Points: 6 and Status: Completed
Following the finalization of the class diagram, we designed our database schema based on the requirements of the class diagram.

- **Testing**

2.○.1 Iteration 1: Added cucumber and Rspec test cases for the features developed.

Points: 3 and Status: Completed

2.○.2 Iteration 2: Added cucumber and Rspec test cases for the features developed.

Points: 2 and Status: Completed

2.○.3 Iteration 3: Added cucumber and Rspec test cases for the features developed.

Points: 2 and Status: Completed

2.○.4 Iteration 4: Added cucumber and Rspec test cases for the features developed.

Points: 2 and Status: Completed

2.○.5 Iteration 5: Added cucumber and Rspec test cases for the features developed.

Points: 2 and Status: Completed

- **Miscellaneous**

2.○.1 Updating the schema and test cases based on the feedback from the client.

Points: 2 and Status: Completed

2.○.2 Creating a rake task to populate the database with a csv file.

Points: 2 and Status: Completed

2.○.3 Making style consistent throughout the application

Points: 1 and Status: Completed

2.○.4 Added navigation buttons so that the user can navigate/go back to the previous page

Points: 1 and Status: Completed

2.○.5 Added constraints that all the views can only be accessed if the stakeholder is logged into the system

Points: 1 and Status: Completed

2.○.6 Add logout option on all the views

Points: 1 and Status: Completed

3. Team Roles

Iteration	Product Manager	Scrum Master	Developers
0	Neha Joshi	Shwetima Sakshi	Prachi Surbhi, Harshvardhan Surolia, Shashank Jagtap, Will Harper
1	Neha Joshi	Shwetima Sakshi	Prachi Surbhi, Harshvardhan Surolia, Shashank Jagtap, Will Harper
2	Neha Joshi	Prachi Surbhi	Shwetima Sakshi, Harshvardhan Surolia, Shashank Jagtap, Will Harper
3	Neha Joshi	Prachi Surbhi	Shwetima Sakshi, Harshvardhan Surolia, Shashank Jagtap, Will Harper
4	Neha Joshi	Prachi Surbhi	Shwetima Sakshi, Harshvardhan Surolia, Shashank Jagtap, Will Harper
5	Neha Joshi	Prachi Surbhi	Shwetima Sakshi, Harshvardhan Surolia, Shashank Jagtap, Will Harper

4. Scrum Iteration Summary

- **Iteration 0**

  Story Point: 0
  1) Created Lo-Fi UI mockup for the student, faculty and admin pages.

  2) Gather the requirements from the client and features that have the highest priority.

  3) Understood how the PhD Annual review works and got ourselves familiarized with the process

- **Iteration 1**

  Story Point: 14
  1) Created the login feature for student, faculty and admin login.

  2) Created a signup feature to add new students to the system.

  3) Added condition such as the email should end with tamu.edu and the password should have a minimum length of 8

4) Decided to use Bcrypt to store the password hash in the database instead of storing the password as text

- **Iteration 2**

  Story Point: 13
  1) Finalized the database design for the application, defined how different objects such as student, faculty, documents, assessment are associated with each other.

  2) Implemented student document submission page which has a form and the feature to upload resume and report.

  3) Added features so that the student can select their committee members and also select a chair for their committee.

  4) Moved the document storage from postgres to S3 as storing the documents in S3 will have performance, the postgres database just has the path for the document stored in S3 bucket.

- **Iteration 3**

  Story Point: 15
  1) Created a dashboard for students which has various options so that the student can navigate to different views such as document submission and edit committee.

  2) Added features to the faculty dashboard, if a student adds a faculty member to their committee then the faculty member can see the response submitted by students.

  3) Faculty members can provide their evaluation, make comments for students and other faculty members, they can see comments made by other faculty members and also nominate the student for reward.

  4) Added features so that the faculty can see the to-do list, completed list of students and also see the response they provided and update if required.

  5) Bug fixes on student document submission logic and also update the UI for student document submission according to client requirements.

- **Iteration 4**

  Story Point: 12
  1) Updated the database model for the student document submission and updated the test cases and faculty dashboard as well.

  2) Admin view was created where the admin can see if the students are ready for review, have the committee members provide feedback, what is the final evaluation of the student and if they are nominated for reward.

  3) Admin can also download the report in CSV format

4) Students can view the comments made by the faculty members.

- **Iteration 5**

  Story Point: 12

  1) Added a feature to update passwords for students and faculty.

  2) Students can view the response they submitted.

  3) Fix timeout issue for the object stored in S3 bucket.

  4) Make the UI consistent on all the pages, adding back button on all views to make the navigation easier

  5) Fix the logic to compute the final evaluation for a student.

  6) Fix bugs in the student document submission page and also make some fields as not mandatory based on the option selected.

  7) We also presented our deployed app to the client and got positive feedback.

5. Customer Meeting Summary

- **Iteration 0 : 16th September**
- In the first client meeting we discussed the requirements for the PhD review system and expectations from this project. We then created the low-fi diagrams to lay down the blueprint of the web application. This was then presented to the client to get approval for the design.

- **Iteration 1 : 30th September and 7th October**
- We implemented the login feature for students, faculty and admin on our web application. We presented the underlying authentication method along with the and frontend for this functionality.

- **Iteration 2 : 21st October**
- We presented the document submission page implementation. The document submission page comprised a mandatory form and resume & report uploading feature. The client suggested some changes in the form fields.
-
- **Iteration 3 : 4th November**
  We presented the changes in the document submission UI as suggested by the client in the last meeting and got positive feedback on them. We also presented the newly implemented edit committee feature and faculty review dashboard.

- **Iteration 4 : 18th November**
- We presented the admin review dashboard that was implemented in this iteration. The client suggested some changes in the student assessment flow where the assessment is done by faculty and committee. We also made some changes in the student document schema as suggested by the client and updated the faculty dashboard coherently.

- **Iteration 5 : 2nd December**
- We presented a new feature that allowed the users to change their password. In the stint of developing this project, we thought of this to be an important requirement which was appreciated by the client as well. We also implemented the functionality to enable students to review their assessments. The client also tried the deployed application with a test user and suggested some cosmetic changes for the UI. We got overall positive feedback for the web application.

6. Individual Contribution

Team Member	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	TOTAL
Neha Joshi	2	2.5	2.5	2	2	11
Shwetima Sakshi	2	2.5	2.5	2	2	11
Prachi Surbhi	2	2	3	2	2	11
Harshvardhan Surolia	3	2	2	2	2	11
Will Harper	2	2	2	2	2	11
Shashank Jagtap	3	2	3	2	2	11

7. BDD/TDD Process

In the initial process of project development our team followed the Behavioral Driven Development (BDD) strategy. During the first meeting with our client, we received a comprehensive overview of the anticipated functionalities and objectives that our software application was expected to achieve. For Iteration 0 our team started by compiling and describing a general list of user stories based on the client's initial meeting with us. Eventually, we made several changes to our user stories as per the client's requirement. We kept track of these changes on a Pivotal Tracker (Github Project).

In the development process, we followed the Test-Driven Development (TDD) strategy to build the code for our software in a structured manner, ensuring that we achieve all the features expected by the client. We employed Rspec and Cucumber frameworks to create test cases before developing a code for a specific functionality. The primary objective of writing test cases in advance was to establish a robust code foundation by systematically passing each of these tests. This approach assisted us in implementing diverse scenarios and making improvements through subsequent iterations.Configuration Management Approach

We used GitHub as our configuration management tool. Each time, team members created a new branch or updated an existing branch while implementing a specific feature. Before each iteration, team members pushed their branches into the main branch after resolving any merge conflicts. The changes were then merged into the main branch after raising a pull request. Over the course of 5 iterations, we had 14 branches and closed 22 pull requests. Git made it easy to manage our code and work on it simultaneously, achieving

Agile Development. Due to this, future teams working on this project as a legacy project will find it easy to understand how the code was developed for each iteration.

8. Issues with tools/deployment

The project initially utilized SQL as its database management system. However, upon deployment on Heroku, various issues surfaced, including problems with Ruby versioning, Heroku stack version mismatches, and Heroku's lack of support for MySQL. To facilitate deployment on Heroku, a migration from SQLite to PostgreSQL became necessary. An additional challenge arose for Windows users who could not install the PostgreSQL gem due to authentication credential requirements. This obstacle was addressed by leveraging the Windows Subsystem for Linux (WSL2) and configuring Ruby on Rails within the Ubuntu subsystem.

After successfully resolving these issues and restructuring the project, deployment on Heroku was achieved, complete with the activation of the automatic deployment feature. The team also encountered minor challenges during Ruby installation, particularly on macOS, where conflicting versions of Ruby were present among team members. This was mitigated by reinstalling Ruby from scratch using the rvm tool and removing outdated and default versions bundled with the operating system.

Furthermore, the project encountered an issue with the AWS S3 bucket used for storing user-uploaded documents. The pre-signed URLs generated for these documents allowed access for only 15 minutes, posing a limitation. To address this, adjustments were made to ensure continuous access for authorized users by utilizing the S3 resource URI.

9. Other Tools

The following additional gems were used in our project:

- AWS Ruby: Official AWS Ruby gem for Amazon Simple Storage Service (Amazon S3). This gem is part of the AWS SDK for Ruby.
- Capybara: Capybara -webkit driver (a gem) is used for true headless browser testing with JavaScript support.
- AWS S3 Bucket: Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance.
- Lucidchart: This is an online tool with which we can create low-fidelity wireframe templates.
- Bcrypt: Bcrypt-ruby is a Ruby binding for the OpenBSD bcrypt() password hashing algorithm, allowing you to easily store a secure hash of your users' passwords.
- Rails ERD: Rails ERD is a plugin for Ruby on Rails that generates diagrams based on your Active Record models.
- csv: We used this gem to read csv files and populate our database.

10. Repository contents and deployment process

Our repository is organized into three primary categories, aligning with Ruby's adherence to the Model-View-Controller Architecture. The model handles application logic, the view is responsible for rendering content, and the controller establishes a link between the model and the user (user requests). Each category corresponds to different features implemented in our project, and we've assigned them informative names based on their functionality, such as faculty, admin, student, student_document, committee, and assessment. Additionally, we've included files for imports, known as GemFiles, following the Ruby architecture. Cucumber testing files, used to test features, are also present. The documentation folder encompasses all six iteration reports. We've streamlined deployment by incorporating all the deployment steps and requirements into our repository, eliminating the need for additional scripts.

11. Future work and enhancement

Currently we have a rake task to populate the database with values, due to this we have to manually add the csv files in the resources folder and push the application on heroku and run the rake task manually. We can create a user interface on the admin dashboard wherein the admin can upload the csv files in the interface and this will populate the database.

12. Important Links

- Project Management (Github): https://github.com/orgs/PhD-Annual-Review-System/projects/2/views/1?filterQuery=

- Github Repository: https://github.com/PhD-Annual-Review-System/PhD-Annual-Review-System

- Heroku: https://phd-review-app-0416787b021c.herokuapp.com/

- Presentation and Demo: https://youtu.be/ifhgklJc8H8