1. Title Page

SYSTEM FOR GRADUATE TEACHING MANAGEMENT AT FAU

Group-2

Sai Bharani Lingampalli

Narayana Murthy Kosireddi

Dilip Gopi Chand Tadekoru

Nihar Yogeshbhai Gajera

CEN 5035: Software Engineering

Florida Atlantic University

Dr. Hanqui Zhuang

Table of Contents

2. Summary

Project Overview:

The "TA Hub" is a fully developed, comprehensive software solution for effective management of TAs in the Computer Science Department at North University. It effectively caters to the varied needs of four different types of users: applicants for TAs, department staff, members of the TA committee, and instructors, thus providing an all-encompassing approach toward the whole TA management lifecycle.

TA Applicants: The applicants for TAs may also submit their applications, including their CVs, by using an easy-to-use platform that also allows applicants to mention prior TA experience. According to their qualifications, they will select courses from the list provided by the department.

Department Staff: The administrators of the Department Staff have a lot at their disposal. They can use the TA Hub by entering detailed course prerequisites to match the experience of TA applicants to specific courses. This is more accurate because existing data will create preliminary recommendations.

TA Committee Members: The TA Assignment decisions made through the TA Hub by the TA Committee, would be then passed on to the administrators for action. Detailed records will be kept from these decisions and notifications sent to the TA Applicants.

Instructors: Using the TA Hub, instructors will review the TAs' performance, offering valuable feedback for future assignments and decisions.

3. Introduction:

3.1 Problem Statement Provided By Professor:

Consider the Computer Science Department at Florida Atlantic University, where a software application for managing graduate Teaching Assistants (TAs) is in development. This system involves four distinct user types:

1. **TA Applicants**: These users will utilize the application to submit their applications along with brief CVs. Each applicant must indicate if they have previously served as a TA at North University. If so, they should specify the relevant course(s) and dates. Additionally, applicants will select courses from a department-provided list that they are qualified to assist with.
2. **Administrators**: This user category includes administrators responsible for inputting courses that require TAs. They will match the expertise of TA applicants with these courses and offer preliminary recommendations to the TA committee based on available data.
3. **TA Committee Members**: Members of the TA committee will make decisions regarding TA assignments and forward them to administrators for implementation. After recording decisions, administrators will notify TA applicants of the outcomes. Successful candidates can then accept or decline offers.
4. **Instructors**: Instructors will assess the performance of TAs in their respective classes. The information gathered will be utilized by both administrators and TA committee members for future TA considerations.

3.2 Background:

The FAU Computer Science Department, through the introduction of specially designed software, is actively working on addressing the challenges in graduate TA management. The creative project was put forward to modernize the TA assignment process by shifting the burdensome manual process to an easy-to-use interface. This was expected to provide TA applicants with an easy and smooth way to apply, and for that matter, easy management and processing of each TA application.

procedure, in order to allow informed recommendations from Administrators, and to enable TA Committee Members to make appropriate decisions.

The projected benefits are in line with the university's strong commitment to innovation and are expected to increase productivity, enhance communication, and provide useful data collection for continued improvement. As the project unfolds, it is expected to develop into an integral part of the academic infrastructure for the department. This innovative tool not only will revolutionize the administration of TAs but also is very likely to have a major positive impact on the general prosperity and expansion of the FAU Computer Science department.

3.3 Literature Review and Market Analysis:

A critical review of the literature on educational technology shows how revolutionary it is in bringing improvement to academic assistance and administrative effectiveness. It emphasizes the need for redefining the roles of Teaching Assistants and the dynamics of collaboration in academia. Comprehensive research on various TA management systems at various academic institutions leads and guides the creation of a custom application to best fit the needs within the Computer Science department at FAU. User experience literature will be indispensable in ensuring the application caters specifically to the demands of TA Committee Members, Instructors, Administrators, and Applicants. Thorough market research conveys data on rivals, highlights current offerings, and underlines areas where educational technology can stand out.

Meanwhile, researching institutional needs provides a strategic perspective on how the TA management application fits in with changing academic demands. While user feedback guides the development of the application's specializations in support of operational efficiency and the growth of the FAU Computer Science programme, regulatory compliance and an analysis of technology adoption patterns help to understand market readiness.

4. Requirements and Use Cases

4.1 Requirements

Functional Requirements :

• No registration or login of the accounts of any of the actors in the project should have any kind of authentication errors.

• TA applicants will be allowed to apply and, if served earlier, mention previous experiences.

• The Interface Should Present Eligible Courses to TA Applicants During Course Selection. • The system should be able to associate the requirements with the TA candidates and make recommendations to the TA committee.

• The system must enable the committee members to decide on TA assignments.

• It should be able to facilitate a notification system so it may inform the successful TA applicants about their selection.

• The performance of the TA for particular classes must be visible to instructors.

Usability Requirements:

• The system will be easy for all the actors present in the project to navigate. • The system should not come across any errors while the user is using it. • By synching up with each other, all four users must be able to access the system simultaneously and without any hassle.

• Communications including the acceptance or rejection of a job offer for a TA applicant and decisions regarding choices on TA assignments should be provided at the present time.

• Reports generated on the TA need to be appropriate since the performance TA is accessed by the instructor.

Security and Safety Requirements:

• The data gathered from the actors may contain critical information, so it must be secured by adding additional encryption layers to prevent data loss.

• In the event of any data loss, we must apply redundancy strategies for recovery of the data.

• The actor should recommend setting a strong password for the system when they join up so that the data is secure.

• The system should maintain audit logs of all user activities, including login attempts and changes to data, for the purpose of security monitoring and incident response.

External Nonfunctional Requirements:

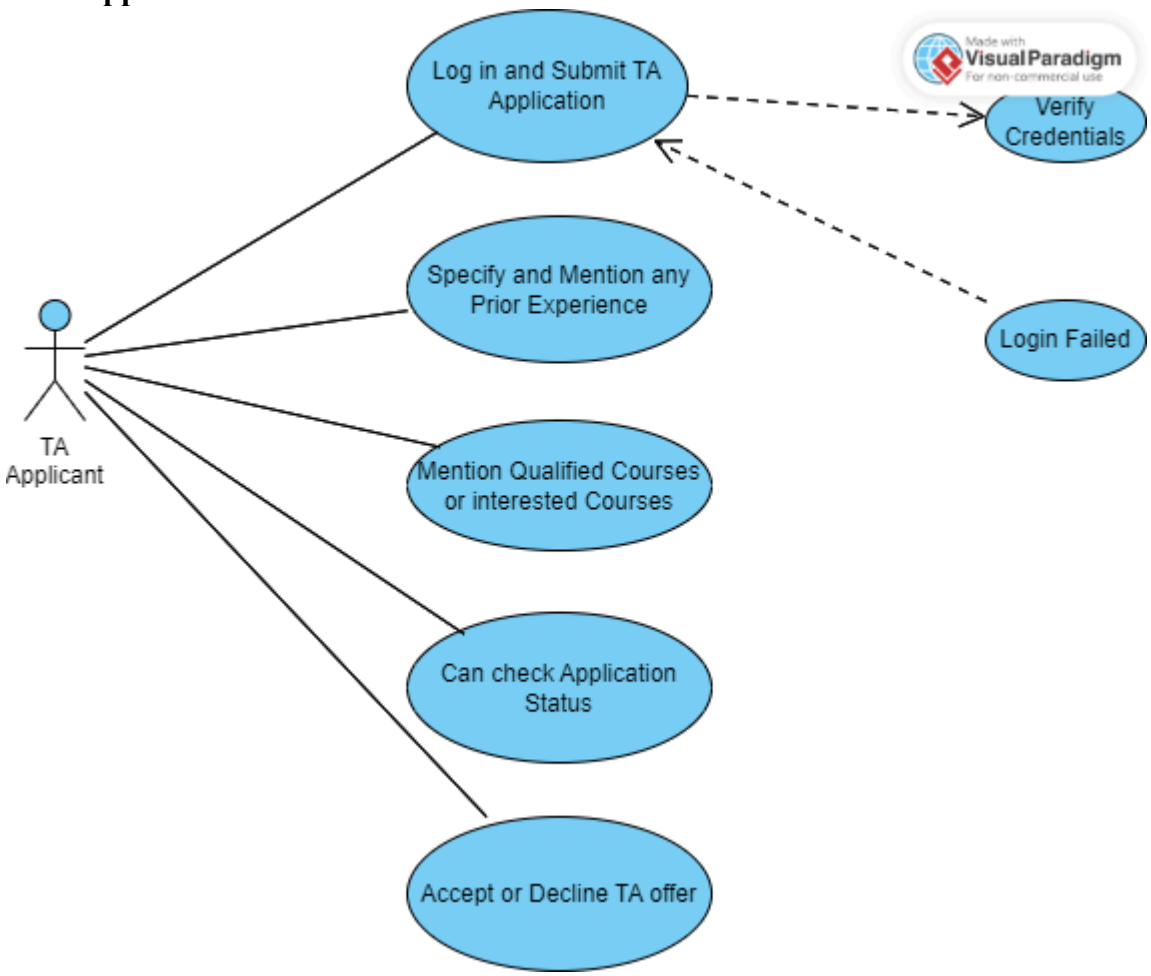
• Application must be compatible with all policies of the university about data protection and integrate smoothly into the university's databases and login systems. • The application should work perfectly across a wide variety of platforms and web browsers for user-accessible and aligned to mobile and web accessibility standards

• In order to ensure user's data security and preparedness of the system in case of any breakdowns/dataloss, provision should be made for solid security: regular back-ups

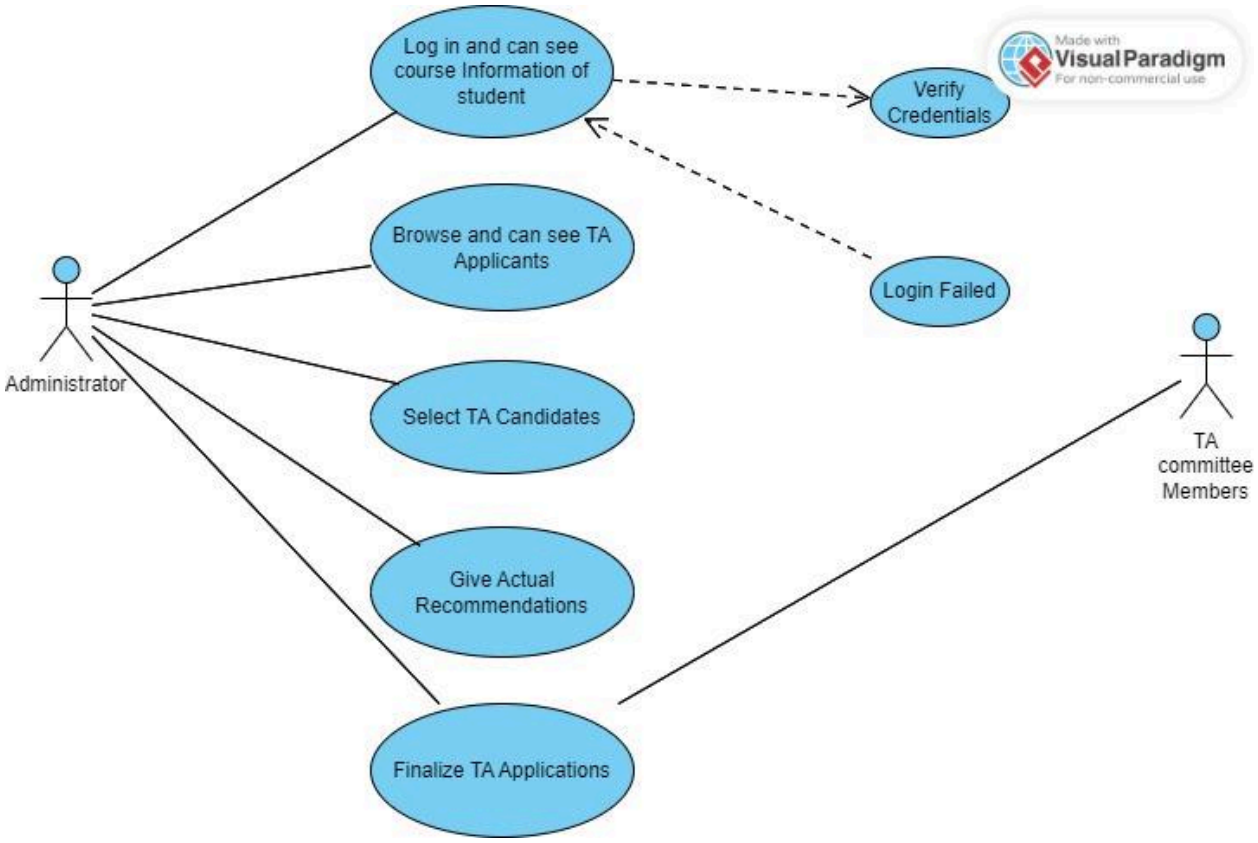
.

4.2 UML Usecase Diagrams:

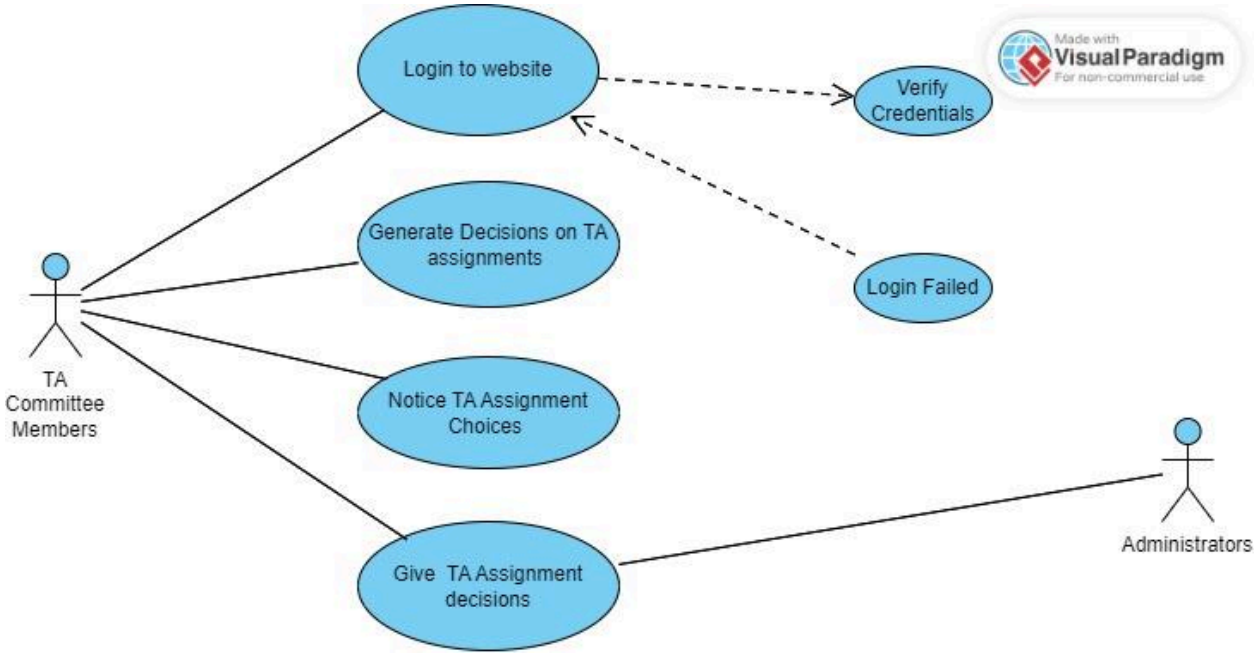
1.TA Applicant:



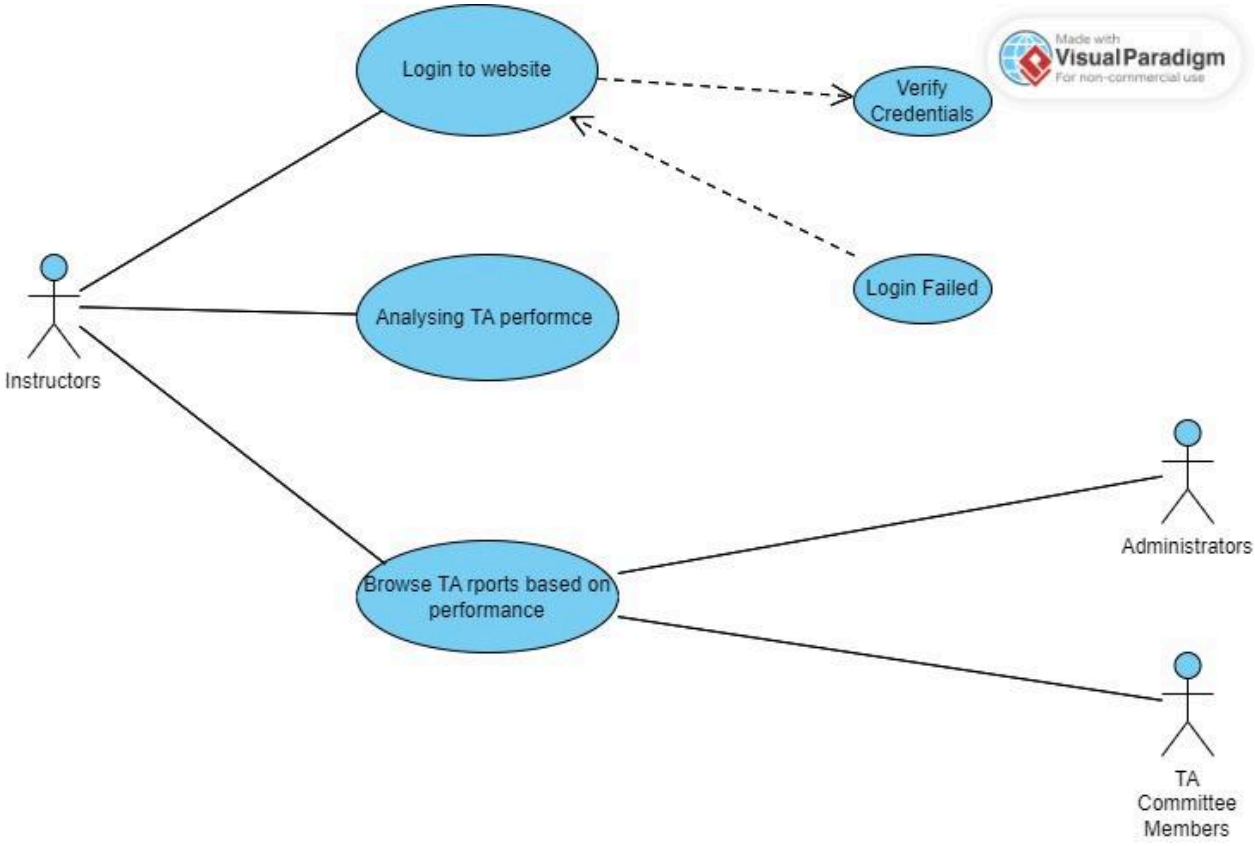
2. Administrators:



3. TA Committee Members:



4. Instructors:



5. System Design

Depending on their responsibilities, administrators, committee members, instructors, and prospective teaching assistants should all be able to sign up, log in, and safely access the system. There shouldn't be any authentication issues when logging in. Candidates for teaching assistant positions ought to be able to choose the classes they can help with, apply, and attach a résumé. Teachers can administer courses that call for a teaching assistant (TA) assignment by logging in. Information on the course and qualifications must also be included. The system ought to be able to connect the requirements to the candidates when endorsing them to the TA committee. When TA applicants select courses, the interface must show them which ones they qualify for. The system will let the committee members choose the TA.

System Design Features According to The participant Type:

The GTA system developed by our team serves four distinct user types: TA applicants, Administrators, TA Committee members, and Instructors. The following components are integral to the system design:

1. TA applicants should easily be able to apply by uploading their CV, filling out an application form, sharing any relevant experience they have, and choosing the courses they are interested in or qualified for.

2. Admins should be able to define what courses are available and those can be selected by the TA applicants according to their preferences and qualifications.

3. Members of the TA Committee shall be empowered to review the TA applications and make decisions regarding the TA assignments and notify selected TAs of their decisions.

4. Instructor must be able to analyze and access performance reports of the TA Applicant.

System Design Elements:

1. Provide a responsive front-end interface, using Next.js, to allow it to work on multiple devices of different screen sizes.

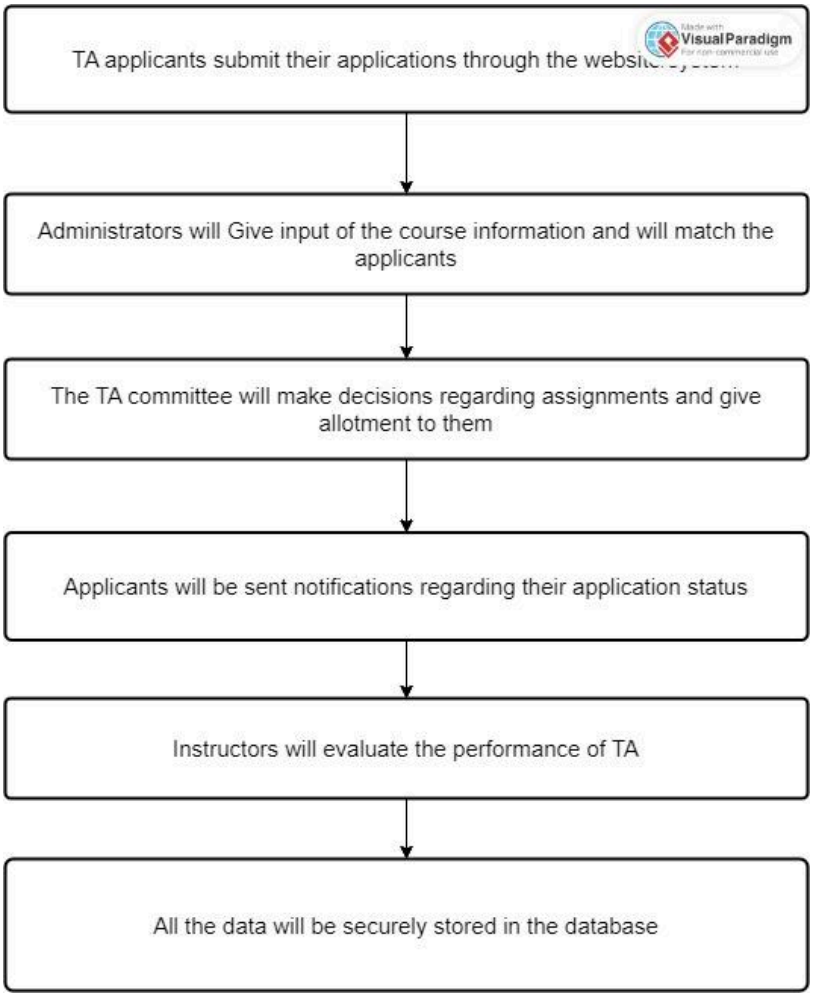
2. Design a strong backend that can handle user logins without problems and also handle data retrieval and posting smoothly without any errors.

3. Use the MySQL database in the design of the system, and guarantee that it stores all information without any problem and safely.

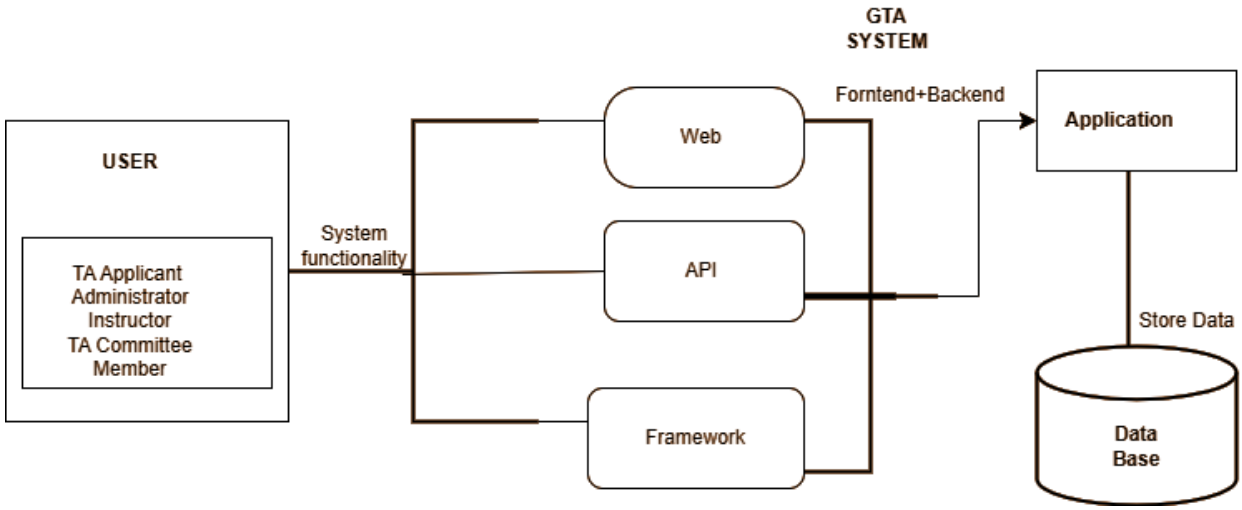
Security in System Design: Upon login, users enter critical data. In order to safeguard this data, encryption technology should be utilized to protect it.

System Maintainability: In order to ensure smooth system operation, we keep releasing new versions with updates and support.

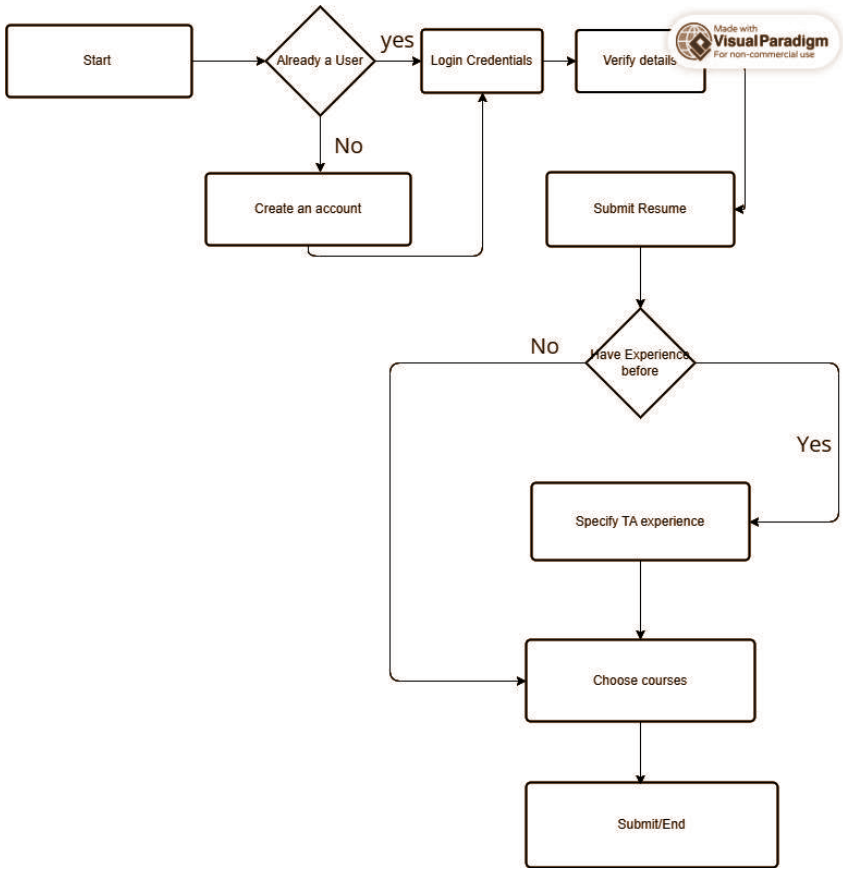
System layout Flow :



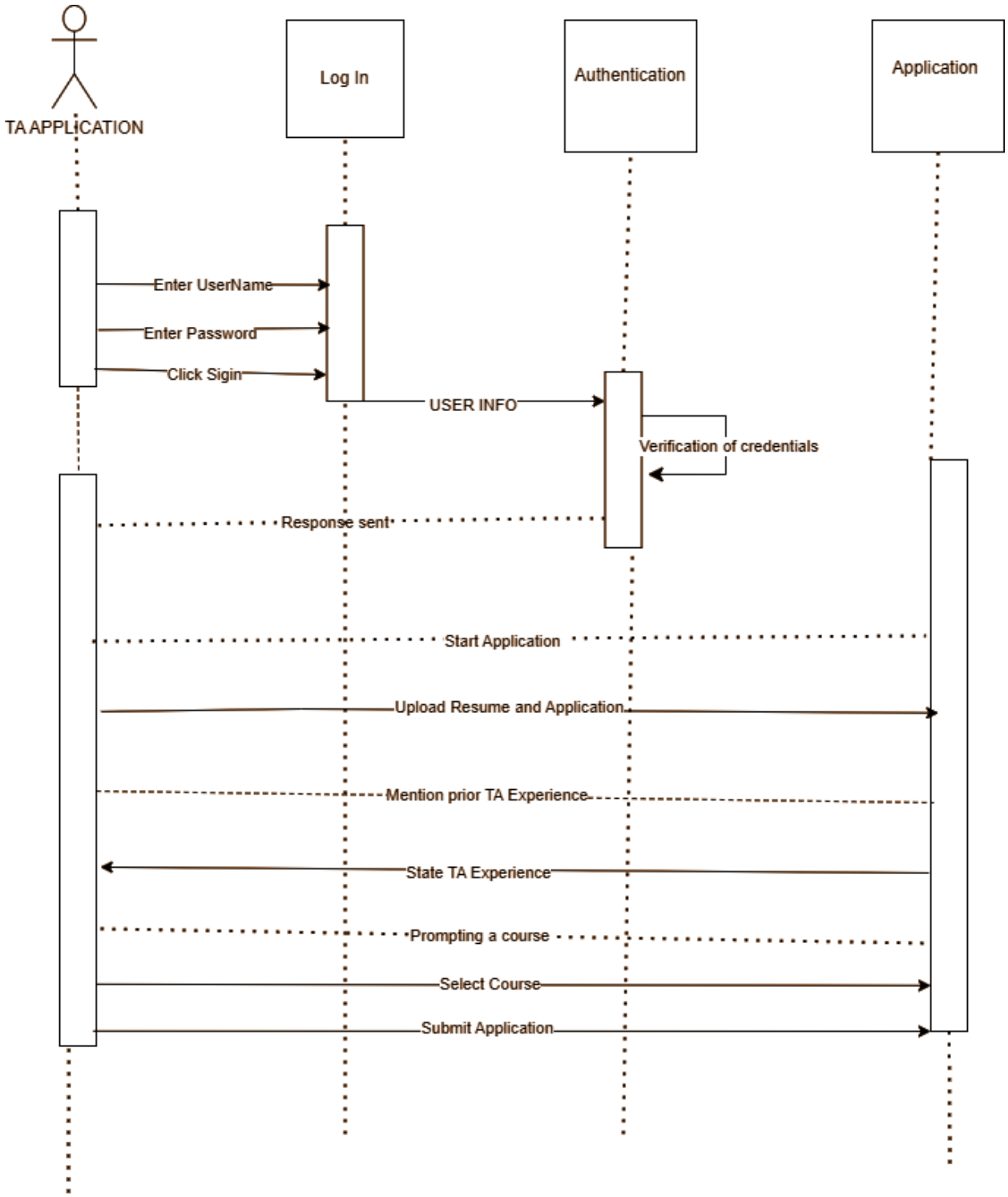
System Design High Level Functional Diagram:



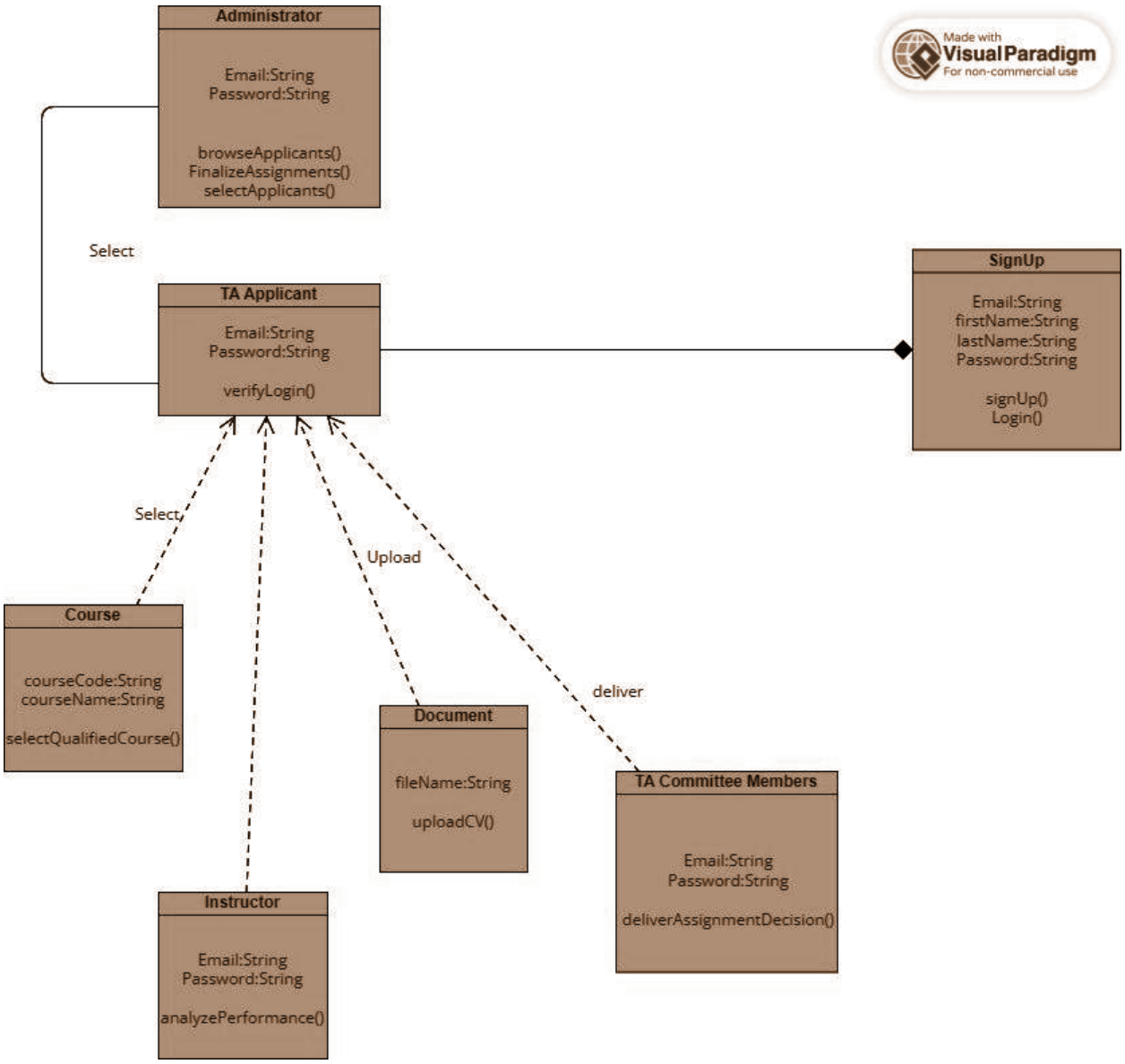
Flow Chart



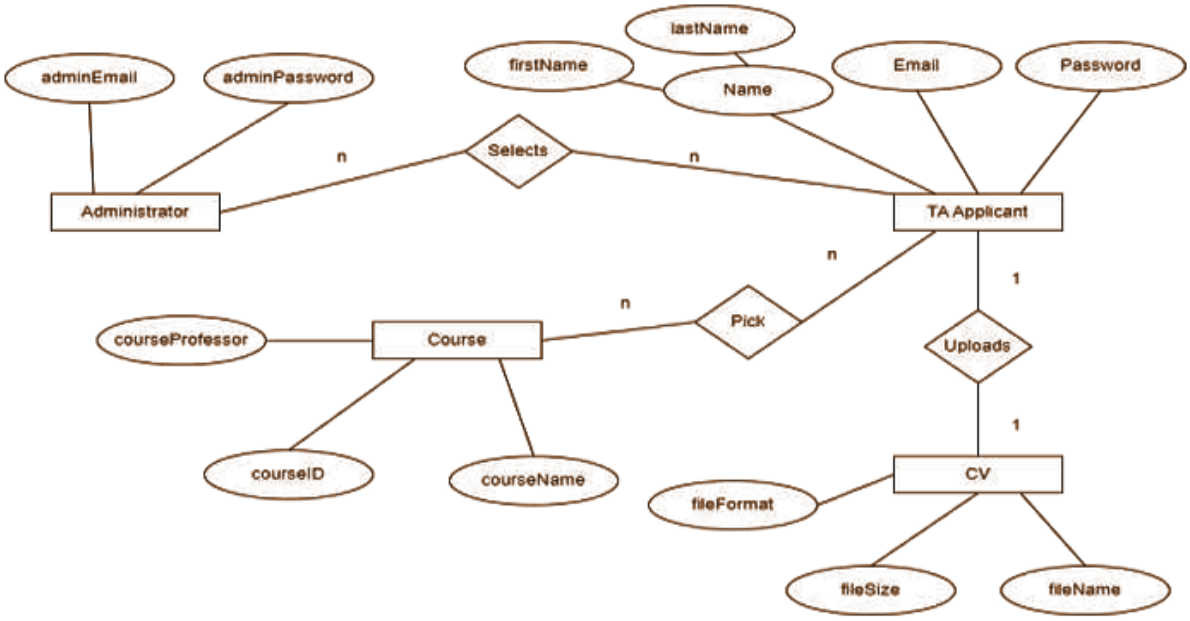
Activity Diagram:



Structural Perspective



ER Diagram: Illustrates the data model



6. Implementation

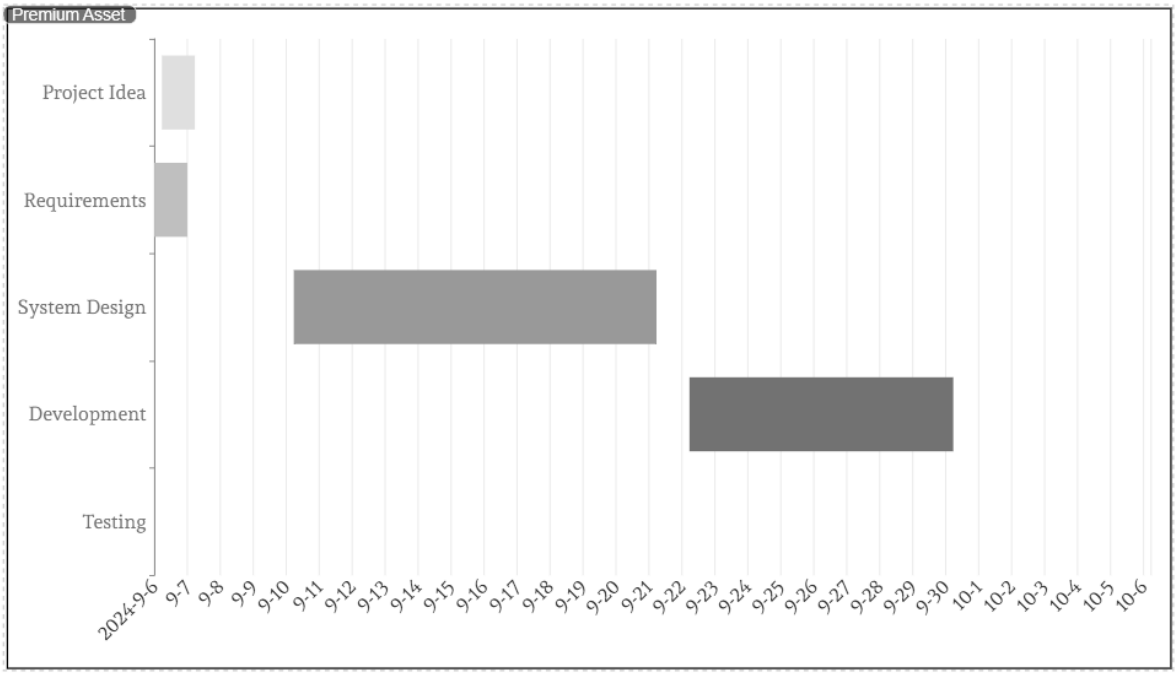
6.1

Sai Bharani Lingampalli - Member A
Narayana Murthy Kosireddi - Member B
Dilip Gopi Chand Tadekoru - Member C
Nihar Yogeshbhai Gajera - Member D

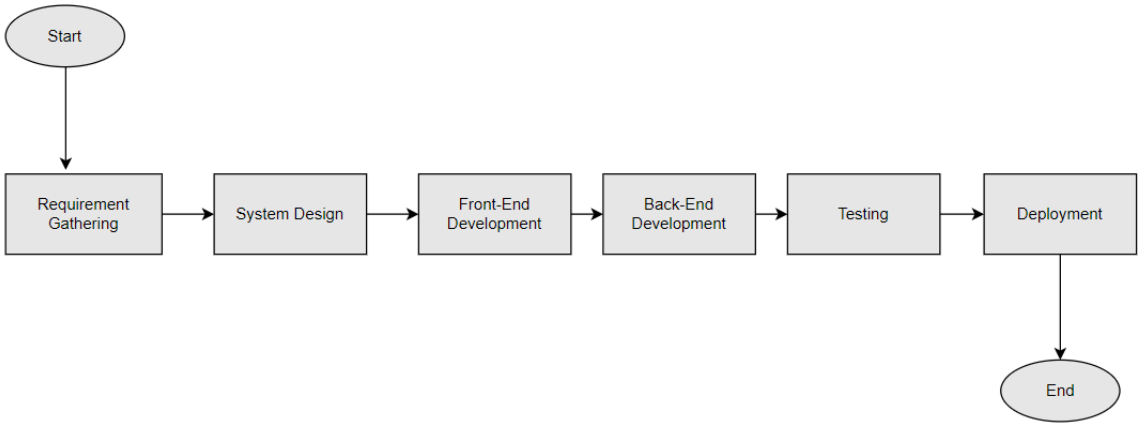
Task	Status	Start Date	End Date	Done by
Project Idea				
1.1 Project Planning	Completed	Sep 6	Sep 6	A
1.2 Determining the solution	Completed	Sep 7	Sep 8	A
Requirements Gathering				
2.1 Functional Requirements	Completed	Sep 6	Sep 7	A
2.2 Non-Functional Requirements	Completed	Sep 6	Sep 7	B
2.3 Use-Case Diagrams	Completed	Sep 7	Sep 7	C
System Design				
3.1 Architectural Design	Completed	Sep 10	Sep 15	D
3.2 User Interface Design	Completed	Sep 11	Sep 15	A &B
3.3 Technical Specifications	Completed	Sep 12	Sep 21	B&C
Development				
4.1 Frontend	Under Review	Sep 22	Sep 27	C&D

Development				
4.2 Backend Development	In Progress	Sep 23	Sep 30	A&B
4.3 Integration	In progress	Sep 23	Sep 30	ALL
Testing				
5.1 Test Planning	In Progress	Oct 1	Oct 5	A &B
5.2 Test Execution	In progress	Oct 2	Oct 6	ALL
Deployment				
6.1 Deploying the app	Incomplete	Oct 14		ALL

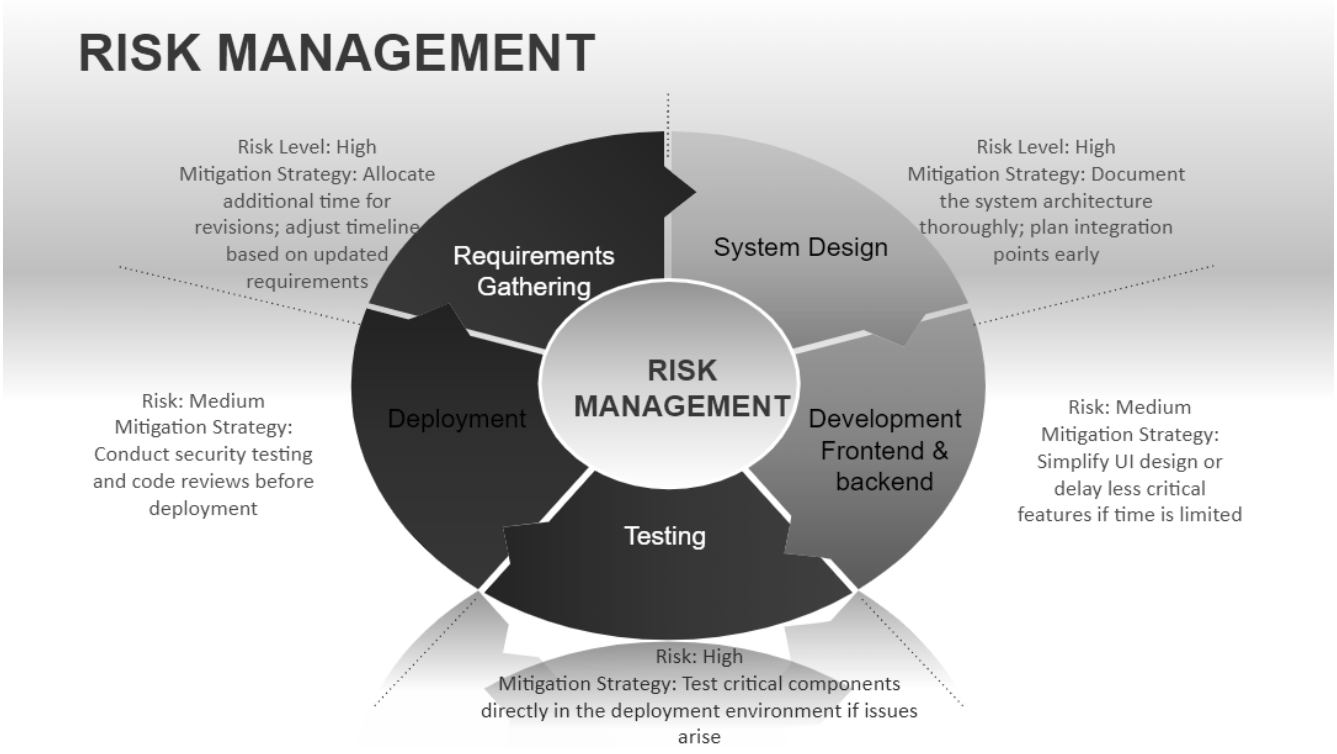
6.2 Gantt chart



PERT Chart (Program Evaluation Review Technique)



Risk Management



Task Breakdown

1. **Task 1: Requirements Gathering and Use Case**
    - **Duration**: 2 days
    - **Responsible Person**: Sai Bharani Lingampalli (Person A)
    - **Due Date**: September 8
    - **Details**: This task involves identifying the project requirements and defining use cases to establish the project scope and functional expectations. It has a short duration because it's an initial, focused effort to outline the key aspects.
2. **Task 2: System Design**
    - **Duration**: Each component takes 1 week (total 2 weeks)
    - **Responsible Person**: Nihar Yogeshbhai Gajera (Person D)
    - **Due Date**: September 21
    - **Details**: This task includes designing the architecture of the system, such as database design, software structure, and technical specifications. It's split into components, each taking one week. Nihar is responsible for ensuring the design aligns with the requirements.

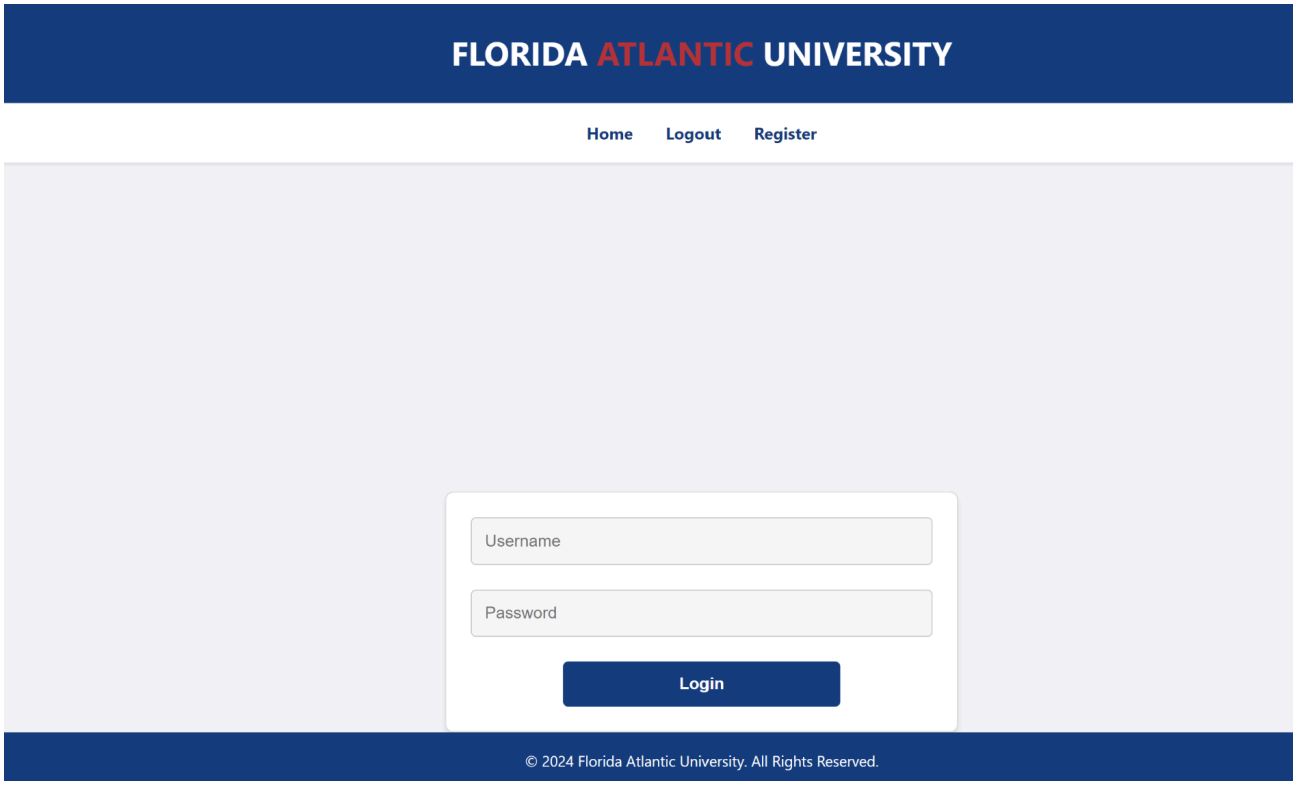
3. **Task 3: Front-End Development**
    - **Duration**: 2 weeks
    - **Responsible Persons**: Sai Bharani Lingampalli (Person A) & Dilip Gopi Chand Tadekoru (Person C)
    - **Deadline**: October 1
    - **Details**: This task focuses on developing the user interface and experience. Two people share the responsibility to complete the task efficiently and on time.
4. **Task 4: Back-End Development**
    - **Duration**: 2 weeks
    - **Responsible Persons**: Sai Bharani Lingampalli (Person A) & Narayana Murthy Kosireddi (Person B)
    - **Deadline**: October 12
    - **Details**: This involves developing the server-side logic, database connections, and API integrations. Again, the workload is shared between two individuals.
5. **Task 5: Testing**
    - **Duration**: 1 week
    - **Responsible Person**: Nihar Yogeshbhai Gajera (Person D)
    - **Deadline**: October 20
    - **Details**: Testing the application for bugs, functionality, and user acceptance is crucial to ensure that the system works as expected before deployment.
6. **Task 6: Documentation**
    - **Duration**: 3 days
    - **Responsible Persons**: All team members
    - **Due Date**: October 25
    - **Details**: Documenting the system (e.g., technical details, and user guides) is a collaborative effort. It ensures that all members are familiar with the system and can provide comprehensive documentation for future use.
7. **Task 7: Deployment**
    - **Duration**: 3 days
    - **Responsible Person**: Nihar Yogeshbhai Gajera (Person D)
    - **Deadline**: October 30
    - **Details**: The final task involves deploying the system, configuring the environment, and ensuring the application runs in a production setting.

Implementated Subsystems Working:

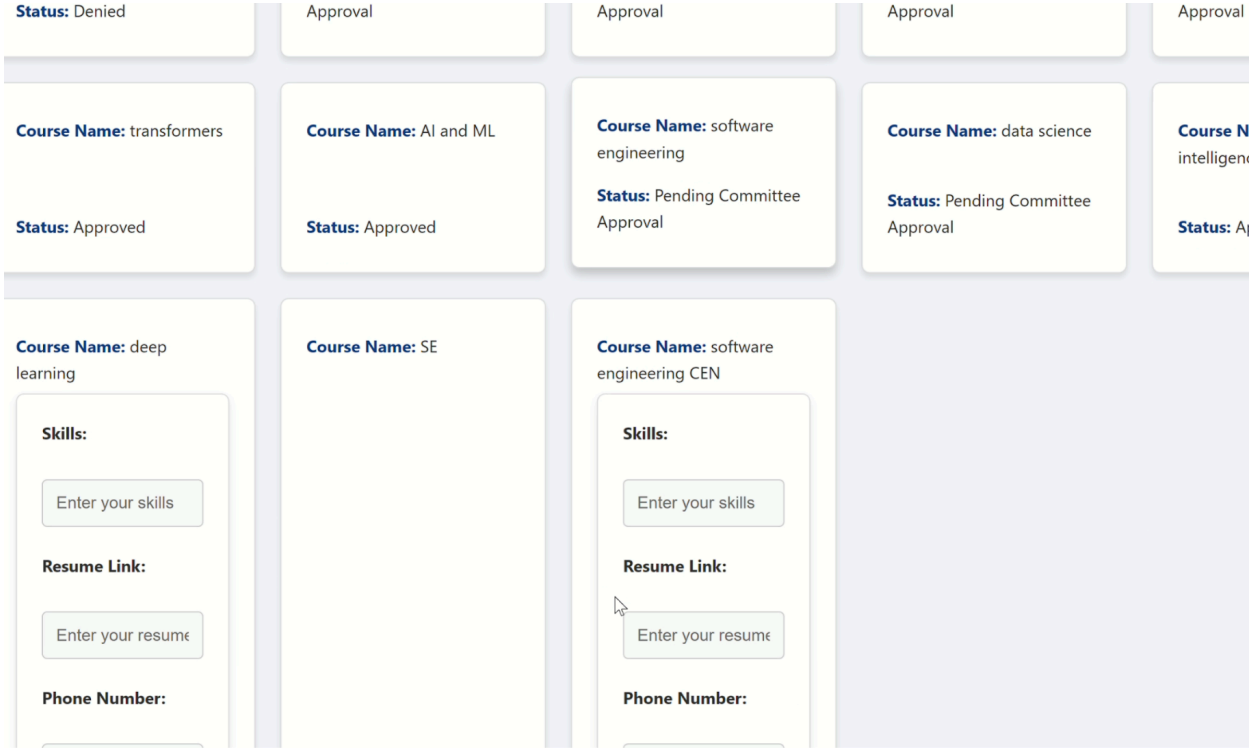
Login and Signup Page:

- *User Authentication:*

1. User enters credentials:



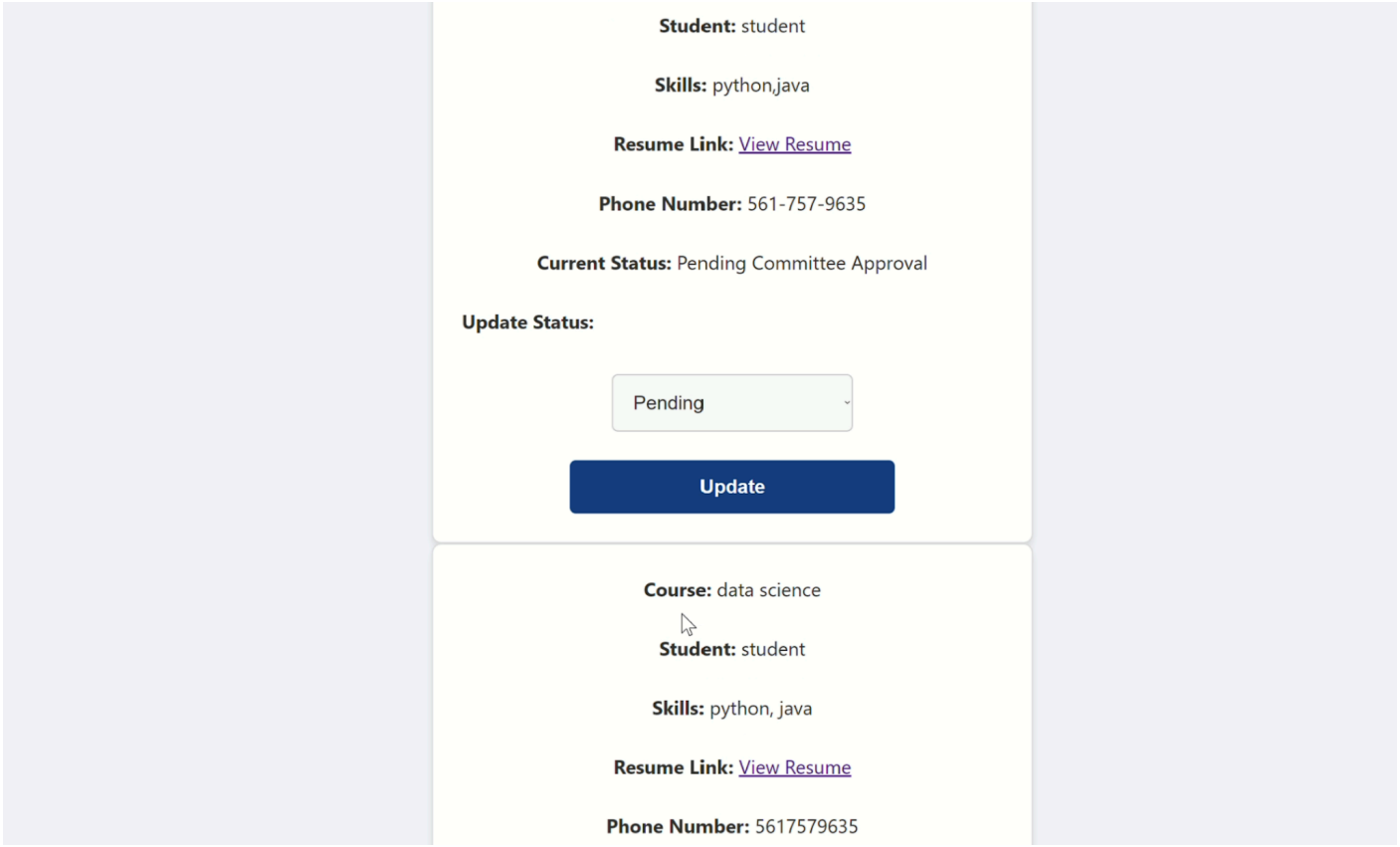
2. Server-Side Validation:



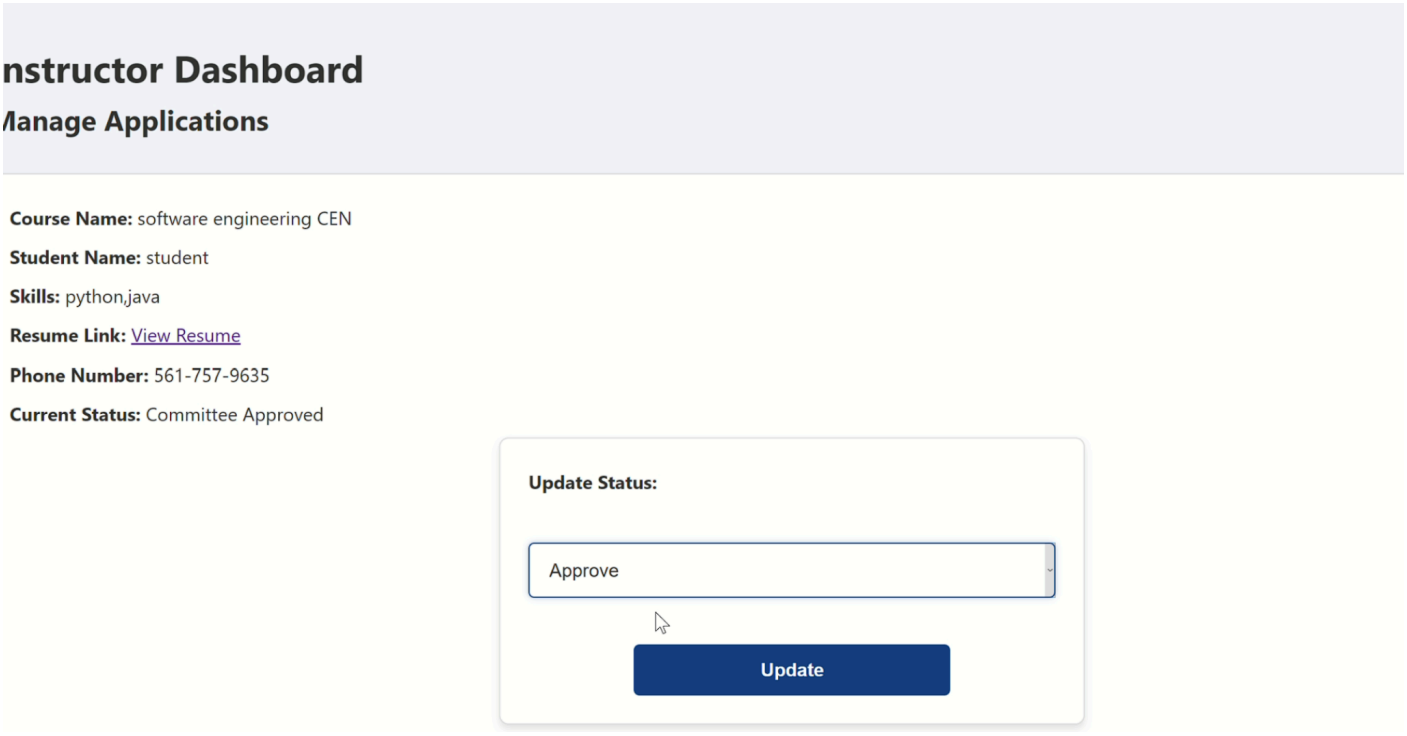
• *Signup Process*

1. Navigation to Registration Page:

Select the "Register" button on the Login Page to be directed to the Sign-Up Page.



Instructor Side



8. Conclusion:

Group 2 has developed the GTA System at FAU to a far higher standard than had been anticipated within the given time frame. Designed for FAU Computer Science Department, the TA Hub is a feature-rich platform with discrete user groups. TA Candidates can easily submit their applications, including all the necessary details about their background and previous teaching experience. Administrators are crucial to the course administration process in matching departmental requirements with TA experience. Members of the TA Committee have decision-making power, ensuring effective TA assignments and open discussion with applicants. Instructors provide insightful feedback through the evaluation of TA performance, feeding back into the cycle of continual growth. Future Plans for the TA Hub:

➢ Optimize decision-making processes for TA assignments using advanced analytics to analyze past data.

➢ Provide better means of communication to promote smooth communication between TA Committee Members, Instructors, Administrators, and TA Applicants.

➢ To maintain consistent information and smooth data flow, explore the options for linking the TA Hub with other academic platforms.

➢ Scope for mechanizing routine procedures in order to avoid laborious manual labour and improve administrative process efficiency.

9. References:

To install Next.js, please follow below reference link:

➢ https://nextjs.org/docs/getting-started/installation

To install antd, please follow below reference link:

➢ https://ant.design/docs/react/introduce

To install Node Js, please follow below reference link:

➢ https://nodejs.org/en/download

To install Express Js, please follow below reference link:

➢ https://expressjs.com/en/starter/installing.html

To install XAMPP, please follow below reference link:

➢ https://www.apachefriends.org/download.html

To create component in react, please follow below reference link:

➢ https://legacy.reactjs.org/docs/components-and-props.html

To create component in react, please follow below reference link:

➢ https://nextjs.org/learn-pages-router/basics/create-nextjs-app/setup