Logo

Description automatically generated

**Guide Genie**

**Kunal Sahjwani**

**Shruti Varade**

**Navkar Jain**

**Samyak gangwal**

**Date: 2-18-2024**

**Contents**

1. Project Description……………………………………………………………… 01

2. Business Case…………………………………………………………………….03

5. Requirements…………………………………………………………………….04

6. Work Breakdown Structure………………………………………………………05

7. Context Diagram………………………………………………………………….08

8. Data Flow Diagram……………………………………………………………….10

9. Use Case ………………………………………………………………………… 11

10. State chart Diagram……………………………………………………………….12

11. Privacy Component……………………………………………………………….13

12. Methodology………………………………………………………………………15

13. Conclusion…………………………………………………………………………16

14. References…………………………………………………………………………17

***MOTIVATION***

*Finding the right research advisor is crucial for students to excel in their academic and research pursuits. However, the process can be time-consuming and daunting due to the vast pool of potential advisors and their diverse areas of expertise. A recommendation system can streamline this process by matching students with suitable professors based on their research interests and preferences.*

**PROJECT DESCRIPTION**

The recommendation system aims to assist students in identifying and connecting with professors who closely align with their research interests. By leveraging data analytics and machine learning algorithms, the system analyzes student input regarding their research interests, academic background, and preferences. It then generates a list of top professors along with their contact information, enabling students to reach out for potential research opportunities.

**BUSINESS CASE FOR THE PROJECT**

***Current State***

* Currently, students often rely on manual searches, recommendations from peers, or departmental advisors to find research advisors. This process can be inefficient and may not always result in the best matches**.**

***Impact***

* Implementing a recommendation system can significantly reduce the time and effort required for students to find suitable research advisors. It can also improve advisor-student matches, leading to more productive research collaborations and academic outcomes.

***Desired State***

* **User Friendly**

The proposed application is similar to existing apps available. Hence it is very user-friendly and can ease the process of recommendation for the students.

* **Transparency within the system**

Students would be able to find all the information required to read more about the research papers of the professors recommended, also would be able to access the information needed to contact the professor. The desired state is to create a user-friendly platform that seamlessly connects students with compatible research advisors, enhancing the overall academic experience and research productivity.

**REQUIREMENTS**

1. ***Functional requirements –***

* User registration and profile creation.
* Input interface for student research interests and preferences.
* Profiling of professors based on research areas, publications, and expertise.
* Matching algorithm to generate advisor recommendations.
* Contact information display for recommended professors.
* Feedback mechanism for students to rate advisor matches.

1. ***Technical Requirements:***

* Database management system for storing student and professor profiles.
* Machine learning algorithms for recommendation generation.
* User interface development for seamless interaction.
* Security protocols to protect user data and privacy.

1. ***Business Policies***

* Privacy Policy: Ensuring the confidentiality of user data and compliance with data protection regulations.
* Feedback Policy: Encouraging students to provide feedback on advisor matches to continually improve the system.
* Usage Policy: Clarifying the intended use of the recommendation system for academic purposes only.

**WORK BREAKDOWN STRUCTURE**

**Phase 1: Planning and Requirements Gathering (2 months)**

Week 1-2: Form project team, designate roles, and responsibilities.

Week 3-4: Conduct stakeholder meetings with students, faculty, and departmental advisors to gather requirements and feedback.

Week 5-6: Analyze gathered requirements, define project scope, objectives, and success criteria.

Deliverable: Requirements documentation, project plan, and scope definition.

**Phase 2: System Design and Development (6 months)**

Week 1-2: Design database schema and system architecture.

Week 3-4: Develop user interface mockups and wireframes.

Week 5-8: Begin front-end development, focusing on user registration, profile creation, and input interface.

Week 9-12: Implement back-end functionalities, including data storage, algorithm development, and matching logic.

Week 13-16: Integrate front-end and back-end components, conduct initial testing and debugging.

Week 17-20: Enhance system features based on user feedback and usability testing.

Deliverable: System architecture documentation, user interface prototype, initial system implementation.

**Phase 3: Testing and Evaluation (2 months)**

Week 1-4: Conduct comprehensive testing, including unit testing, integration testing, and user acceptance testing.

Week 5-6: Gather feedback from pilot users and stakeholders, identify and address any issues or bugs.

Week 7-8: Perform performance testing and optimization to ensure system scalability and efficiency.

Deliverable: Test reports, bug fixes, optimized system performance.

**Phase 4: Deployment and User Training (1 month)**

Week 1-2: Prepare for system deployment, including server setup, data migration, and security configurations.

Week 3-4: Roll out the system to users, conduct training sessions for students and faculty on system usage and best practices.

Deliverable: Deployed system, user documentation, training materials.

**Phase 5: Maintenance and Updates (Ongoing)**

Establish regular maintenance routines to address any system issues, bugs, or security vulnerabilities.

Monitor user feedback and usage patterns to identify areas for improvement and future updates.

Implement regular software updates and enhancements based on evolving user needs and technological advancements.

**CONTEXT DIAGRAM**

**Prototype context diagram**

A diagram of a guide

Description automatically generated

Our current prototype does not have access to student information therefore we do not have labels to our data. With the implementation of the prototype, we can start collecting student information securely and get labels for our recommendations to provide a better user experience.

**TO BE CONTEXT DIAGRAM**

A diagram of a diagram

Description automatically generated

**DATA FLOW DIAGRAM**

A diagram of a student process

Description automatically generated

The data flow diagram for level 0 was built on the basis of 'to be context diagram'. The processes were broken down to show the flow of data between the processes. The boxes with a title are the major processes of the system as referred from the context diagram. The open boxes are called datasets where the data is saved. The boxes without titles are the actors from the use case who would interact with each other using the system and have an exchange of data using these processes.

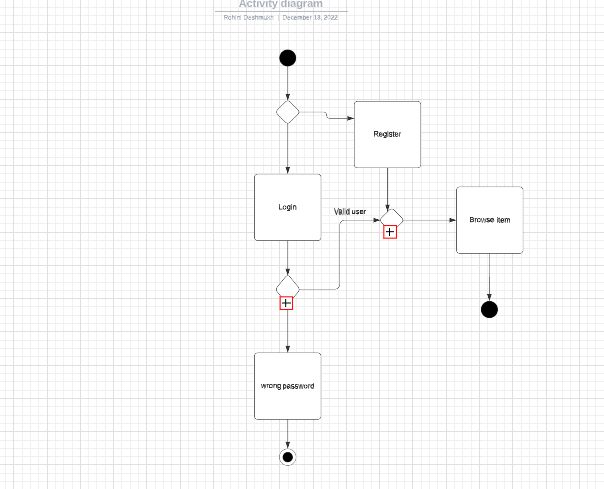
**USE CASE DIAGRAM**

A diagram of a person

Description automatically generated

In the Use case diagram, there are two actors, the professors and the students who can access the system and get recommendations according to their area of interest and can explore more about the faculty/student.

**STATE CHART DIAGRAM for LOGGING**

****



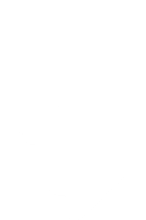
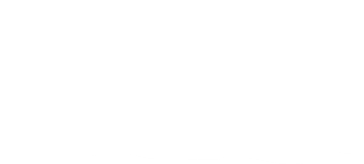
In the above state chart diagram is for the login of the student to the system. If he/she is a valid student with the correct credentials they have access to the recommendation system

If a student is a new to system they have to register first, a valid college student can access the application. And if the credentials are incorrect the message gets popped up as “Wrong Password” and they are send back to the home screen and thus the end of the process.

**PRIVACY COMPONENT**

Diagram

Description automatically generated



**PRIVACY COMPONENT**

This is one of the important components of our system which will protect our student's personal information. The Student and professor are the main actors in our privacy component diagram. All the communication and action between the actors are protected with some security tools.

First, while login in or Signing in to the system all the details are secured using security like SSL (Secure Sockets Layer)which is an encryption-based security protocol which will protect all email communication and alerts. And Multi-Factor Authentication (MFA) will help the student to log in safely to the app with their UMASS student email and password. All these details while logging in are secured in our app database.

Secondly, the password of the student and staff is secured using hashing technique. The password they entered is encrypted into short numbers or letters using an algorithm.

This above security will help to protect all communication and avoid Spam mail to the students and only authorized, registered students can have benefited. There is a privacy notice on the data that is stored or used for data management for the university to keep track of the registered student.

**METHODOLOGY**

While Waterfall helps with the completion of a single project, Agile helps with the completion of several smaller projects. Even if creating an application is a single project, it contains several smaller projects or intricate capabilities like alert generation, safe student data storage. Waterfall focuses on effective project delivery, while Agile provides a product mindset with a focus on customer satisfaction .Agile projects frequently cost less money and are finished more quickly. Since many project components are ambiguous and vague, they offer greater flexibility but less predictable outcomes. Waterfall projects frequently cost more money and take longer to finish. Agile greatly decreases total project risk while maximizing value throughout the development process. The Agile model makes it possible to easily adopt changes while maintaining openness with donors, staff, and students. As soon as the first design is successful, products are created and integrated while adhering to customer requirements and generating cash through various self-funding initiatives. Data-driven decision-making techniques will help the university business  discover untapped markets and internal possibilities.

**Conclusion**

* With the implementation of the recommendation system, the manual effort to find professors would reduce which would encourage research work.
* The successful implementation would give the people in academia a space to network and get to know each other.
* The project would also help professors select best students for their research work.
* This would eventually help student interest in research work and incline students to innovate and develop new things, creating work opportunities.
* This prototype already utilizes real data for the recommendation therefore it is easily scalable to a larger scale to include more universities and create a space for people in academia.

**REFERENCES**

* Scholarly from pypi

[scholarly · PyPI](https://pypi.org/project/scholarly/)

* Gemini Word Embeddings

[Gemini - Qdrant](https://qdrant.tech/documentation/embeddings/gemini/)

* Google scholar

[Google Scholar](https://scholar.google.com/)

* Umass faculty directory

[Directory - UMass Boston (umb.edu)](https://www.umb.edu/directory/)