

DON BOSCO INSTITUTE OF TECHNOLOGY

Premier Automobiles Road, Kurla (W), Mumbai - 400070



Department of Computer Engineering
(Session 2025-2026 Odd)

MINI PROJECT I PROPOSAL

on

“UniSync: Enhancing Student Experience”

Subject: Mini Project

(Design Thinking Fundamental)

Semester: III (SE Computers)

Subject In-Charge: Ms. Pooja Bansode

Group Members:

Name	Roll No.
Mrunal Sagvekar	36
Rahul Prajapati	28
Elissa Quadros	29
Aditya Sabnis	35

UniSync: Enhancing student experience

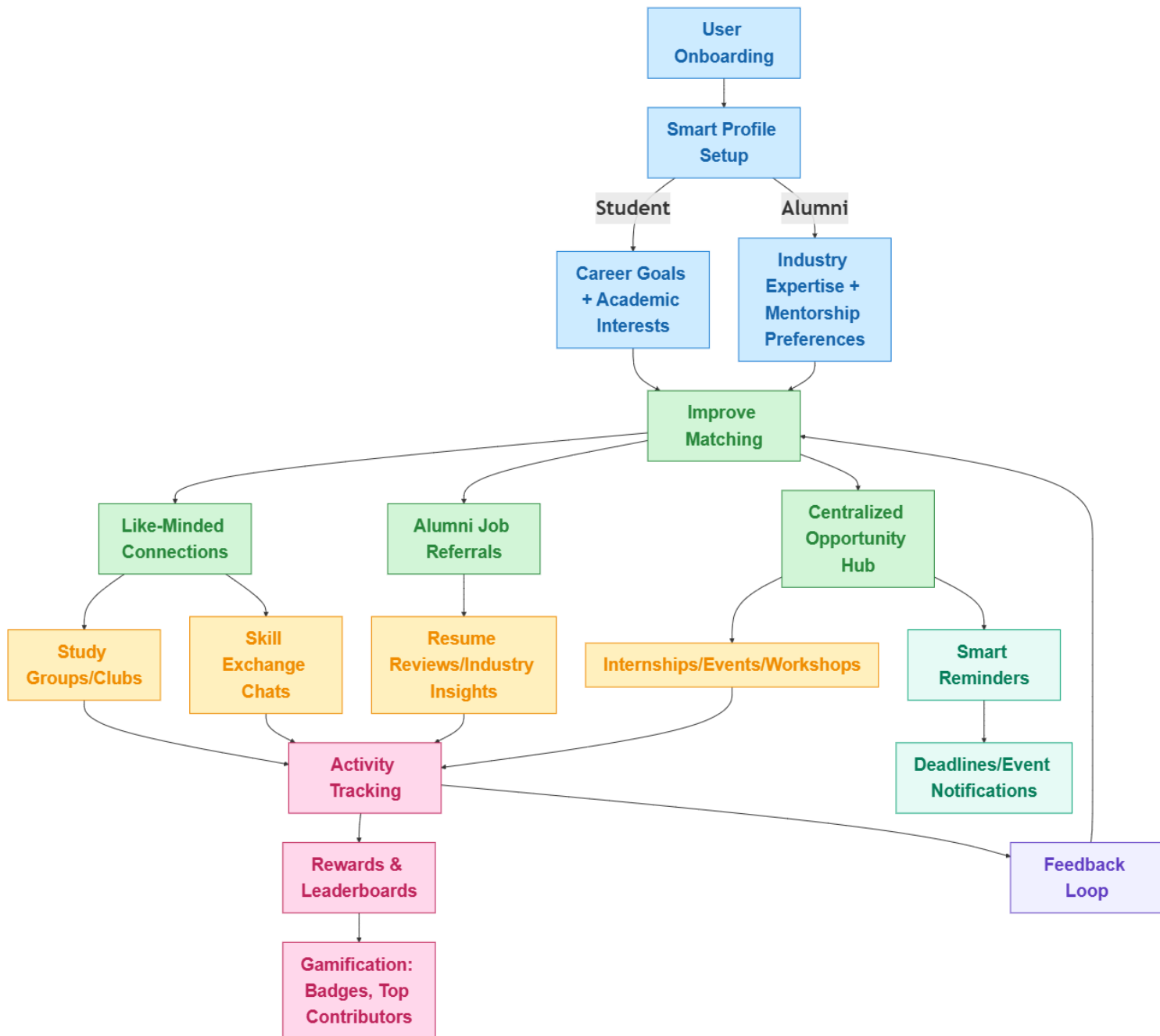
1) Abstract:

In a modern academic environment, students interact with digital platforms not just to access study materials but also to seek guidance, engage in discussions, and receive mentorship. As the volume of this content grows, ranging from academic questions and resource requests to career-oriented discussions, organising it effectively becomes a key challenge.

UniSync is a campus support project designed to foster student-mentor collaboration and academic resource sharing. Within such a platform, users can frequently post queries or share notes that are related to multiple topics simultaneously. The platform's core functionalities are smart routing of questions to mentors, improved content searchability, and intelligent recommendation of notes.

UniSync provides online event hubs for on-campus workshops and hackathons, smart reminders for upcoming events, and helps students connect with groups/communities of like-minded peers. Users can also engage in one-on-one private messaging with seniors, alumni, and mentors. Additionally, the platform provides a gamified experience, rewarding users for participation and featuring leaderboards for top contributors.

2) Design/Workflow Diagram:



3) Algorithms/Methodology Used:

I. Smart Matching Algorithm

Smart matching algorithm refers to a process designed to logically pair or connect entities based on a sophisticated analysis of their characteristics, preferences, and behaviors.

It helps to pair student profiles with mentees based on shared interests, goals and skills. *Example:* Alumni with "Data Science @ Google" nodes link to students studying ML.

- **Data Collection :**

The algorithm gathers comprehensive data about the entities to be matched, including explicit preferences (e.g., survey responses), implicit behaviors (e.g., browsing history, interactions), and relevant attributes (e.g., skills, goals, location).

- **Multi-dimensional Analysis :**

Instead of relying on a single criterion, smart matching considers multiple aspects for matching.

- **Feedback Mechanism :**

The smart matching algorithm accepts feedback from users to learn from successful and unsuccessful matches which helps to refine the matching criteria and improve the quality of future recommendations.

II. Real-time chatting

It allows students to message their peers, seniors or alumni mentors for academic guidance or resource sharing. It includes:

- **Message Creation and Sending:** Users generate messages, which are then transmitted to the server.
- **Server Processing:** The server validates, assigns timestamps and IDs, and stores messages in a database.
- **Message Delivery:** The server identifies active recipient connections and sends the message.
- **Message Storage:** Chat histories are stored in databases for later retrieval.
- **Authentication:** Securely manages user identities and access to conversations.

III. Smart Reminders:

Smart reminder algorithm intelligently adjusts reminder timing based on user behavior and context. Students receive alerts for upcoming events based on their interested domains.

IV. Thread grouping:

Thread grouping involves checking for specific keywords and automatically sorting similar posts/threads into a group. It basically reads and categorizes text.

Example: Post: How to crack internships? Group: #careeradvice or #coding

V. Community matching

It recommends study groups or communities that have the same interest as the user. It works:

Step 1: Tracks what the user discusses or searches about frequently.

Step 2: Uses community detection methods to find groups with similar interests.

Step 3: Recommends those groups or communities to the user.

Example: User chats about "robotics projects" Suggested group: "Robotics Enthusiasts".

4) Possible input and expected outcome:

- **Input:**
"How can I prepare for the GATE exam while managing my semester workload?"
- **Expected Output (Multi-label prediction):**
 - "Exam Preparation" : Connects the user with other students preparing for GATE so that its easier to share resources or receive information related to the exam.
 - "Time Management": The platform sets smart reminders for GATE related workshops / sessions to overcome missed opportunities.
 - "Career Guidance" : Pairs the user profile with a senior mentor which has appeared for the exam in the past so that the user can receive guidance.

The model will analyse the question's textual description and output **multiple labels** relevant to the query.

5) References:

<https://www.goinconnect.com/knowledge-articles/6-ways-students-benefit-from-being-part-of-student-communities>

<https://medium.com/data-science/community-detection-algorithms-9bd8951e7dae>

<https://www.geeksforgeeks.org/javascript/real-time-chat-application-in-javascript/>

<https://www.geeksforgeeks.org/java/java-threadgroup-class/>