Northeastern University
Brain Buddies

Project Proposal Topic Selection Report

EMGT 5220 - Engineering Project Management
Fall - 2024

Visionary Mavericks

Team Members
Banait, Shreyas
Gupta, Utkarsh
Munagala, Sai Chandra
Shah, Priyank
Shamim, Md Zafer
Shanmuganathan, Akash
Talib, Shiekh

Team Point of Contact
Shanmuganathan, Akash
+1 (857) 693-8403
Shanmuganathan.a@northeastern.edu

TABLE OF CONTENTS

LETTER OF TRANSMITTAL

Date: 12/04/2024

Prof. Sharad Bajracharya

Northeastern Graduate School of Engineering

130 Snell Engineering
360 Huntington Avenue
Boston, MA 02115

Dear Prof. Sharad Bajracharya,

I am pleased to present you with the project proposal for Brain Buddies, a platform designed to enhance academic collaboration among Northeastern students. This initiative addresses the critical challenges students face in finding suitable study partners and forming productive collaborations for academic projects.

Brain Buddies offers a centralized, secure platform that connects students based on courses, skills, and interests. Through features such as real-time communication tools, SSO login for security, and intelligent matching algorithms, the platform facilitates seamless collaboration and ensures a user-friendly experience. The platform aligns with the university's commitment to fostering academic excellence and community building.

This project has been developed with careful consideration of user needs, financial viability, and long-term impact. We are grateful for your guidance and feedback throughout this process, which has been invaluable in shaping the project. We hope you find this proposal compelling and reflective of the skills and knowledge we have gained during the course.

Sincerely,

Banait, Shreyas

Gupta, Utkarsh

Munagala, Sai Chandra

Shah, Priyank

Shamim, Md Zafer

Shanmuganathan, Akash

Talib, Shiekh

EXECUTIVE SUMMARY

Brain Buddies is a platform designed to help Northeastern students easily find academic collaborators. It connects students based on their courses, skills, and interests, making it easier to find study partners or teammates for projects. The goal is to improve teamwork and academic success by creating a more collaborative learning environment.

The platform will be developed in four main steps: planning and research, design and testing, building the platform, and launching it with ongoing updates. Each step is important to ensure the platform is easy to use and meets students' needs.

During the planning phase, surveys and interviews will be used to understand what students want from the platform. This feedback will help decide which features are most important and identify the challenges students face when collaborating. This will ensure the platform meets real needs and improves teamwork.

In the design phase, simple models of the platform will be created and tested with students to make sure it's easy to use. In the development phase, the platform will be built, including features like secure logins, smart matching of students, and tools for real-time communication. Testing will ensure the platform works well and is secure.

Brain Buddies will make it easier for students at Northeastern to connect and collaborate on academic projects. By focusing on students' needs, security, and feedback, the platform will become a valuable resource. With a budget of $136,232 and a solid return on investment, Brain Buddies will offer significant benefits to the Northeastern student community.

..

1. INTRODUCTION

1.1 Problem

Northeastern University students often have difficulty finding study partners and collaborators for academic projects. This issue arises from a lack of a dedicated platform that connects students based on their specific academic needs. Current platforms, like social media and general messaging apps, do not focus on academic collaboration, making it hard for students to find peers with similar interests and schedules. As a result, many students spend too much time searching for study partners instead of focusing on their work, (User Survey Analysis)[1] leading to missed opportunities for collaboration and support.

The main problem is that students struggle to connect with one another in a meaningful way for their studies. Without a tool designed specifically for this purpose, they face challenges in organizing study sessions and group projects. This can create stress and hinder academic success, as collaboration is key to effective learning.

The success of students depends on their ability to learn collaboratively. When students work together, they can share knowledge, provide support, and learn more effectively. However, without a targeted platform to facilitate these connections, students may miss out on valuable teamwork experiences. Addressing this problem is crucial for creating a more supportive and productive academic environment at Northeastern University. Given these challenges, it is essential to develop a dedicated solution that facilitates academic collaboration among students

1.2 Solution

We will create Brain Buddies, an application designed specifically for Northeastern University students. This app will help students find study partners and collaborate on academic projects. Key features include a personalized matching system that connects students based on shared classes, interests, and availability, making it easier for them to find suitable peers.

Brain Buddies will be developed using an Agile approach, involving short cycles called sprints. Each sprint will focus on:

- Secure Login: Students will log in with their Northeastern credentials for security.
- Matching System: A personalized matching tool will suggest study partners based on common courses and availability.
- Group Management: Students can form study groups easily, with tools for scheduling and sharing resources.

After each sprint, we will gather user feedback to refine the platform.

Brain Buddies targets the unique academic needs of Northeastern University students. Unlike general platforms, our tool provides tailored connections based on courses and interests. This focused design will enhance collaboration, leading to improved academic performance and a stronger campus community. By addressing the challenges students face in finding study partners, we believe Brain Buddies will significantly enhance their learning experience.

[1] *User Survey Analysis.* (n.d.). https://northeasternmy.sharepoint.com/:w:/g/personal/banait_s_northeastern_edu/EbqPZ4kf7YZBgaDDVrnvUYBe_8cAtqJkT9iIBzYlpyDTA?e=4acscK

2. PURPOSE & SOLUTION

2.1 Purpose

Students at Northeastern University face difficulties when trying to collaborate or learn from each other, especially when seeking peers with specific expertise or when trying to join academic projects. This is due to the lack of a streamlined platform that helps them quickly find and connect with others who have the skills they need. Student at Northeastern University has been facing a lot of problems when they try to collaborate with other student/alumni but are unable to because the lack of good centralize system to connect students having specific skills/interests. If some student wants to collaborate with other students for a group project or needs academic support, he must either search everyone on LinkedIn or social media which get tough and is highly inefficient. Tools like LinkedIn and NU Source are too focused-on jobs and much less on academic collaborations finding classmates and seniors. The idea behind Brain Buddies is to provide a platform where students can contact their peers/alumni if they need any help regarding academics

2.2 Objective:

The project is about helping Northeastern University students to connect and work together based on similar academic interests and projects. Students can use the platform to find other students, join groups, and easily find students with similar interests. The user mockup flow will be minimalistic and secure, with features like login, finding matching students, getting real-time messages, and searching for specific things.

Major Deliverables:

- **Secure Authentication:** Implement Northeastern University's Single Sign-On (SSO) for secure login and user verification, ensuring that only verified students can access the platform.
- **Data Encryption**: All the data which will be transferred between the user and platform will be encrypted using SSL/TLS protocols ensuring sensitive data like password, personal details are protected.
- **User Consent and Data Control**: User will have control over their data that will them to manage what data can be shared with other users. There will also be an option to delete data if they no longer wish to use Brain Buddies.
- **Matching Algorithm:** The matching algorithm connects students by considering shared courses, interests, study habits, and schedules through a weighted scoring system. It takes key inputs such as course enrollment to match students in the same classes, interest tags to pair those with similar hobbies or academic preferences, and study habits to align individuals with compatible study styles, such as group or individual study. Additionally, the algorithm factors in schedule alignment to identify students with overlapping availability. To ensure better matches over time, the algorithm incorporates user feedback, allowing for continuous improvement.
- **Study Group Management:** Brain Buddies will use tools like task boards (Trello) and calendars and scheduling tools (Microsoft Calendar) to create, manage, and organize study groups or academic projects, allowing students to track group progress and coordinate tasks efficiently. Further to work on a project in real time Brain Buddies has Microsoft OneDrive integration.
- **Real-Time Collaboration Tools:** Build in-platform communication features such as messaging via MS Teams, notifications, and alerts to help students stay updated on study sessions, deadlines, and group activities.

- **User-Friendly Interface:** Design an intuitive, responsive interface which will adapt to different devices, screen sizes. That makes it easy for students to get familiar with the UI irrespective of the device they are on which in turn will make it easy to find and join study groups, manage academic projects, and customize their collaboration preferences.

Boundaries:

In-Scope:

- The platform will only be for Northeastern University students.
- Students will log in using their Northeastern credentials.
- It will focus on helping students connect for studying and collaborating on projects.

Out-of-Scope:

- It won't be available to students from other schools or universities.
- There won't be any integration with outside academic tools or platforms.
- Advanced features like AI tutoring won't be part of the first version but might come in future updates.

3. TECHNICAL OVERVIEW

1. **Overview**

The project will begin by gathering user feedback through surveys to identify pain points in academic collaboration at Northeastern University. A cross-functional team will be established, including technical developers, UI/UX designers, a project manager, and stakeholders, ensuring their regular feedback is incorporated.

Brain Buddies is addressing a common problem faced by Northeastern University students in finding a partner in academics by creating a platform dedicated to students and alumni based on academic interest, courses. The goal of the platform is to provide a streamlined platform to collaborate as well as tools to improve group study sessions, project planning, and sharing resources.

2. **Scope**
    - Mobile App: With backend support develop a mobile app which is compatible with all platforms.
    - Tools and Features: Providing communications via instant messaging, scheduling via calendars and online cloud file sharing
3. **Functional Requirements**

3.3.1 User Authentication

- Authentication Protocol: Single Sign On (SSO) through Northeastern University identity provider using OAuth 2.0.
- Multi-Factor Authentication: MFA via DUO as an added security layer.

3.3.2 Protocols

- Encryption[2]: Using HTTPS with SSL/TLS for all data transmission. For sensitive data transmission (like password, personal information) AES-256 encryption. And using TLS 1.2 or higher for securing all data transmissions.
- Data Collections Policy: Collection of only minimum necessary data for platform functionality also setting transparent data usage policy through clear, concise Privacy Policy and Terms of Service agreements presented at account creations.
- Control: User has control over their data visibility, consent to share their data and choose what data whey want to share as well as deleting their profile. User will retain full data ownership and will be able to export their data in machine readable format.
- Data Retention: Set policies focused on data retention and automatic purging of inactive accounts after a set amount of time.
- Data Sharing: No user data will ever be shared with any third party.

[2] Encryption Standards:
National Institute of Standards and Technology (NIST) guidelines for AES encryption.
https://csrc.nist.gov/projects/aes

- Data Residency: Closely following FERPA guidelines [3]to host user data within United States using AWS region compliance[4].
- Annual Compliance Certification: All stakeholders will undergo annual training on FERPA compliance.
- Role-Based Access Control[5]
  Assign role-based access permission like
  - Students: View and modify data, manage visibility settings, and delete accounts.
  - Admins: Limited to anonymized analytics data and cannot access individual user profiles.
  - Developers: Only access test environments; no access to live user data.
  - Support Staff: Restricted access to user-initiated support tickets only.

3.3.3 Audits

- Access logging and Monitoring: Keeping a log of all data access events like timestamps, access data types and user ID. Implementing AWS CloudWatch [6]for real time monitoring.
- Security Audits [7]: Performing quarterly penetration testing to identity and address any vulnerability. As well as employing third party auditors to verify compliance with FERPA standards.
- Incident Response Plan: Implementing a response plan in case of data breaches including notifying users, following GDPR standards[8].
- Data Review: Conducting a semi-annual review of collected data to ensure all data is relevant and adherent to privacy policy.

3.3.4 Matching Algorithms and Preferences [9]

Scoring System for matching users: Users will be matched based on a score which will consider factors such as:

- Academic courses and Interests (40%): Matches are based on common interest and common academic courses, and it will weigh 40% of total score.

---

[3] FERPA (Family Educational Rights and Privacy Act):
U.S. Department of Education, Family Policy Compliance Office
FERPA Overview

[4] AWS Security Practices:
AWS provides extensive documentation on compliance, encryption, and data residency.
https://aws.amazon.com/compliance/

[5] Role-Based Access Control (RBAC):
NIST RBAC Model and Principles
https://csrc.nist.gov/publications/detail/sp/800-162/final

[6] AWS CloudWatch:
Amazon documentation on logging and monitoring with AWS CloudWatch.
https://aws.amazon.com/cloudwatch/

[7] Best Practices for Data Retention and Security Audits:
International Association of Privacy Professionals (IAPP).
https://iapp.org/

[8] GDPR (General Data Protection Regulation): European Union Official Journal
https://gdpr-info.eu/

[9] General Matching Algorithm Design
Burke, R. (2007). *Hybrid web recommender systems*. The adaptive web, 377-408.
[Matching Algorithms in Practice](https://medium.com/swlh/building-a-recommendation-engine-for-matching-partners-360a1b40c8f5)

- Availability (20%): Users will be matched if they have overlapping free time. For the next best match, the user should have at least 50% overlap for study sessions.
- Study habits (15%): User will only be matched if they have same study habit's like studying in quiet environments, short/long study sessions, etc.
- Shared tags (15%): Tags like Project Management, Economics, Python will be matched using cosine similarity matrix.
- Past Collaborations (10%): User past collaborations will be considered. Positive feedback will help to improve score by 10%.

Improvements: Based on feedback from users, algorithm accuracy will be tracked and used for incremental upgrades in algorithm using following mechanism:

- Feedback Loop: User will rate the study sessions in the range 1 to 5. Lower rating will be used to analyze mismatched attributes and will be considered for future matching scores.
- Incremental updates: Scoring thresholds are periodically reviewed to optimize quality. Monthly model training incorporates new feedback.
- Error Analysis: Mismatched pairs will be examined by root cause analysis. Patterns from mismatched pairs will be examined and used to improve matching algorithm.
- A/B testing: Testing multiple algorithms on subset of user to measure user satisfactions and paring rate while incorporating wining algorithm into main deployment.

Machine Learning [10]Integrations: Machine learning will be used for scoring system to get higher accuracy and adaptability.

- Collaborative Filtering: [11]By using historical data from successful matches from past user, compatibility for new users will be predicted. Matrix factorization will be implemented to uncover latent features that will define successful matches.
- Natural language Processing: using pre trained model like BERT, to identify semantics relationship between user inputs and terms. Applies Natural Language Processing [12]to analyze user profiles and course description.
- Dynamic Weighing System: Using a reinforcement learning [13]approach will be adjusting the final score based on the user setting/feedback. If a user wants study habit is more crucial, the final score will be adjusted accordingly.

---

[10] Machine Learning Libraries
[Scikit-learn Documentation] (https://scikit-learn.org/stable/)
[TensorFlow Documentation] (https://www.tensorflow.org/)
[PyTorch Documentation] (https://pytorch.org/)

[11]Collaborative Filtering
Koren, Y., Bell, R., & Volinsky, C. (2009). *Matrix factorization techniques for recommender systems*. IEEE Computer, 42(8), 30-37

[12] Natural Language Processing
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019).
BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT.
[Word2Vec Explained] (https://towardsdatascience.com/understanding-word2vec-and-its-implementation-c041ae931bdf)in

[13] Reinforcement Learning
Sutton, R. S., & Barto, A. G. (2018).

Backend Implementation

- Technology: Python will be the backbone of logic and machine learning model also using python's libraries like Scikit-learn, TensorFlow, and PyTorch for algorithm implementation. Redis for frequent matches and improve response times.
- Performance Optimization: Redis [14]will manage frequent re-matching and will compute scoring matrices to minimize processing time.

3.3.5 Group Management

- Kanban: Integrating Trello APIs for task management within the app.
- Calendar: Integrating Microsoft Calendar API for scheduling.

3.3.6 Project Management

- Cloud Storage: Integrating Microsoft OneDrive for document storing and sharing as well as collaborating.
- Permissions: Providing access control like ability to edit or view only to each file user want to share to protect sensitive content.

3.3.7 Communication

- Messaging: Integrating Microsoft Teams SDK for real time collaboration seamless connectivity.
- Notifications: Providing updates and keep user informed about group activities via in app notification and optional SMS and email notifications.

3.3.8 Collaboration

- Reminders: Adding automatic remainders about upcoming study sessions.
- File support: Adding multi-media support all file types

3.3.9 UI and UX

- UI: User friendly design implementation using React Native for multi-platform support also adding easy navigation with limited steps for gathering all relevant information.
- Accessibility: Implementing WCAG 2.1 standards for easy accessibility to people with disability.

4. Non-Functional Requirements

3.4.1 Performance

- Latency: Keeping response times under 200 ms for important operations.
- Scalability: Integrating backend system architecture using AWS auto scaling and elastic load balancing to support up to 10000 users.

3.4.2 Reliability

- Uptime: Using AWS EC2 and RDS to target 99.9% availability and database management.
- Recovery: Implementing daily backup with retention period of 30 days while configuring RDS for automated backup.

Reinforcement learning: An introduction. MIT Press.
[Q-learning Algorithm Overview]
(https://towardsdatascience.com/q-learning-algorithm-with-python6a5fb4c6e3ce)

[14] AWS and Redis Performance Optimization
- [AWS Auto Scaling Documentation] (https://aws.amazon.com/autoscaling/)
- [Redis Caching for Performance] (https://redis.io/docs/about/)

3.4.3 Compliance: Strictly following FERPA guidelines to handle students' data and keep user privacy.

5. System Architecture

3.5.1 Frontend: For framework using React Native for compatibility between iOS and Android.

3.5.2 Backend

- Core Framework: Using Node JS and REST API for seamless integration between frontend and backend. For real time messaging and notification implementing WebSocket support.
- Database: Implementation of PostgreSQL hosted on AWS RDS for data like profile, group, matches. Redis for data like recent matching. Cloud hosting AWS EC2 for data like user uploads.

3.5.3 Integration

- Single Sign On and DUO.
- Messaging Teams SDK.
- Notification – optional Outlook for email and Twilio for SMS.
- Calendar – Microsoft calendar API.
- Task Management – Trello API.

6. **Development Phases**

3.6.1 Project Initiation: Setting up development environment initializing git repository and establishing code architecture also doing user surveys.

3.6.2 Design and prototype: Wireframing onboarding, matching and group creation pages. Designing mock UI for feedback surveys.

3.6.3 Core Development:

- Backend: Building data models authentication and API endpoints.
- Frontend: Developing match, login interface and integrating Trello and Microsoft calendar.[15]

3.6.4 Testing and feedback: Conducting user testing and releasing beta version for testing and feedback.

3.6.5 Deployment: Full deployment with AWS CloudWatch, providing onboarding material, FAQ and user support channels.

4. IMPLEMENTATION PLAN

1. Part I: Project Initiation

The project will begin by gathering user feedback through surveys to identify pain points in academic collaboration at Northeastern University. A cross-functional team will be established, including technical developers, UI/UX designers, a project manager, and stakeholders, ensuring their regular feedback is incorporated.

Team Structure:

- Project Manager: Responsible for overall project coordination
- Developers (2-3): Frontend and backend developers responsible for coding the app.
- UI/UX Designers (1-2): Design the user interface and user experience for the app.
- QA Engineers (1-2): Responsible for testing, bug tracking, and ensuring the app meets quality standards.
- Cloud Administrator: Handles cloud setup and API integrations.

Key tools:

- Survey tools (Google Forms, SurveyMonkey) for gathering feedback
- Slack/Teams for communication and collaboration
- Jira/Asana for task tracking and project management

2. Part II: Project Planning

The project will adopt an Agile methodology with 2–4-week sprints. Sprint planning, reviews, and retrospectives will be conducted regularly to ensure the project is adaptable to changes and feedback from stakeholders.

High-Level Product Backlog:

- User survey setup and data collection.
- Secure login functionality (OAuth 2.0).
- User profile and preference management.
- Matching algorithm.
- UI/UX design and front-end development for collaboration interface.
- Group management and scheduling feature development.
- Beta testing and user feedback integration.

Key tools:

- Jira or Trello for managing sprint boards
- GitHub/GitLab for version control and continuous integration (CI/CD)
- Figma for designing UI
- Backend will be hosted on AWS EC2 with AWS RDS for database management

3. Part III: Project Execution

During execution, tasks will be tracked through Kanban boards in Jira, and sprints will follow Agile ceremonies (planning, daily standups, sprint reviews, retrospectives).

Metrics for monitoring:

- User engagement (time spent on platform, number of logins)
- Group creation and activity levels

- Academic collaboration frequency (tracked through in-platform interactions)

Key tools:

- Jira for sprint and task tracking
- Google Analytics for engagement tracking

4. **Part IV: Project Closure**

At the end of the project, a detailed final report will summarize the outcomes, including key accomplishments, challenges, and metrics such as platform adoption and user satisfaction. A post-project retrospective will capture lessons learned for future iterations or projects.

Metrics will be used to assess project success, such as:

- Improvement in student collaboration (based on pre- and post-launch surveys)
- Reduction in academic collaboration bottlenecks

Key tools:

- Confluence for documentation
- AWS CloudWatch for performance monitoring

5. **Project Schedule:**

- Months 1-3: Initial development and internal testing.
- Months 4-6: Beta launch with select Northeastern students for feedback.
- Months 7-9: Full launch with improvements based on the beta phase.
- Months 10-12: Post-lau nch updates and scaling.

4.1 Brain Buddies Mobile App Development Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) outlines the steps necessary to develop and implement the Brain Buddies mobile application successfully. In this project, flexibility, collaboration, and iterative development will be emphasized using Agile methodology. This approach ensures regular stakeholder engagement and continuous feedback through defined ceremonies, including Sprint Planning**,** Daily Standups**,** Sprint Reviews, and Sprint Retrospectives**.**

To deliver incremental features of the app, the project will be broken down into manageable sprints. The team structure consists of cross-functional teams with roles such as Product Manager, Scrum Master, Developers (frontend and backend), UI/UX Designers, and Quality Assurance Engineers. Product Managers prioritize the product backlog, and Scrum Masters make sure Agile practices are followed. To ensure that the app aligns with the needs and expectations of users, regular touchpoints will be conducted with stakeholders, including Northeastern University students and faculty members. Please refer to Appendix A for the detailed WBS.

4.2 Schedule

The "Brain Buddies" development process is planned and tracked using the project schedule, which is an essential tool. The project schedule specifies the beginning and ending dates of every task as well as the sequence in which they must be finished.

The four main stages of the schedule are planning, design, development, and deployment. Beta testing, API integration, core functionality development, user research, and app interface design are important tasks. Resource allocation, task dependencies, and timelines will all be managed using Microsoft Project.

By monitoring progress against the original plan using a Gantt chart, the project manager will be able to spot any delays and take appropriate corrective action. The project will remain within budget and on schedule with regular updates and ongoing oversight. Please refer to Appendix B for the detailed schedule.

4.3 Responsibility Chart

The responsibility matrix helps assign tasks to team members clearly. It indicates who should be informed, who must approve the task, who will contribute, and who oversees each task. This guarantees that everyone is aware of their responsibilities and prevents one individual from being overworked.

Team members are listed on one side of the matrix, and tasks are listed on the other. It displays the roles that each person plays for each activity. Confusion, overlaps, or responsibilities being left unfilled are avoided in this way. The matrix can help the team work more efficiently and prevent miscommunications. For a comprehensive responsibility matrix, see to Appendix C.

4.4 Resource Allocation

The resource allocation plan details the budget for the project's expenses, including labor, software tools, cloud services, and other operational costs required for the development of "Brain Buddies." This plan ensures that all necessary resources are available to keep the project on track.

This plan helps us allocate resources effectively, ensuring the project stays within budget while meeting its objectives. For a detailed breakdown of resource allocation.

Resource	Type	Role	Effort (hrs)	Rate/Hour	% Allocation
Utkarsh	Labor	Full Stack Developer	160	65 $	18.865
Talib	Labor	project manager	304	60 $	35.84
sai	Labor	Data Engineer	168	65 $	19.81
Zafer	Labor	Product Manager	296	60 $	34.9
priyank	Labor	UI/UX Engineer	144	55 $	16.93
Aakash	Labor	Full Stack Developer	160	65 $	18.865
Shreyas	Labor	Quality Assurance Engineer	128	50 $	15.09

4.5 Stakeholders

The following is a brief overview of the stakeholders involved in the project, as mentioned throughout this report.

Primary Stakeholders

Project Team

- Full Stack Developer - Gupta, Utkarsh
- Project Manager - Talib, Shiekh
- Data Engineer - Munagala, Sai Chandra
- Product Manager - Shamim, Md Zafer
- UI/UX Engineer - Shah, Priyank
- Full Stack Developer - Shanmuganathan, Akash
- Quality Assurance Engineer - Banait, Shreyas

Secondary Stakeholders

- University IT Department
- Northeastern University Administration
- University Library and Learning Resource Centers
- Alumni Relations or Mentorship Programs
- University Legal and Compliance Teams
- University Marketing and Communications Team

Customer

- Northeastern University students
- Northeastern University alumni
- University Faculty and Staff

Vendors

- Amazon AWS
- Development Tools
- Microsoft
- DigiCert - SSL/TLS Certificate Provider
- OAuth Provider for Northeastern University's SSO Full Stack Developers
- Trello
- Google Analytics
- GitHub
- SurveyMonkey and Google Forms
- TrustArc - Data Privacy Compliance Solutions

Development Team UI/UX Designers

- QA Engineers
- DevOps Engineers
- Database Administrators
- Security Analyst
- Cloud Operations Specialist

Technical Support Team

- Technical Support Specialist
- IT Administrators
- IT Manager
- Application Support Specialist
- Support Operations

Sponsors

- Northeastern University Administration

5. EXECUTION PLAN

5.1 Project Monitoring

Keeping track of the project is important to ensure "Brain Buddies" succeeds. The monitoring plan will include regular reviews of progress, costs, schedules, staff performance, and quality.

Monitoring Strategies:

- Progress Tracking:
  - Weekly Meetings: Regular meetings with the team to check progress.
  - Milestone Checks: Reviews at key points in the project
- Metrics to Measure Success:
  - Tasks completed from the work breakdown structure (WBS).
  - Customer satisfaction scores after launch.
  - User activity on the platform (e.g., logins, active users).
- Reviews and Testing:
  - Stakeholder meetings to confirm the project is on the right path.
  - Beta testing with students to gather feedback.
- Documentation:
  - Create detailed reports on progress, issues, risks, and solutions.
  - Regularly update the risk register to reflect current challenges.

5.2 Project Control

- Managing Costs:
  - Track expenses like labor, software, and cloud services.
  - Regularly check if the project is providing good value for its cost.
  - Avoid unnecessary spending and stick to the $112,440 budget.
- Keeping the Schedule:
  - Use tools like Jira or Microsoft Project to track timelines.
  - Update schedules and fix any delays quickly.
  - Have backup plans for unexpected problems.
- Managing the Team:
  - Clearly assign tasks using a RACI matrix.
  - Regularly review team performance and provide feedback.
  - Offer training if needed to address skill gaps.
- Ensuring Quality:
  - Set clear quality standards for all deliverables.
  - Use testing tools and audits to maintain quality.
  - Apply feedback from beta testing to improve the final product.

5.3 Risk and Issue Management

- Handling Risks:
  - Keep a list of possible risks like delays or privacy issues.
  - Plan ways to reduce the impact of these risks, such as using backup systems.
- Fixing Issues:
  - Use a system like Jira to track and resolve problems.
  - Assign tasks to the right team members for quick solutions.

5.4 Project Auditing

Regular audits will check if the project is on track, following standards, and using resources well. Audits will include:

- Technical Review: Check system design, APIs, and data security.
- Status Review: Compare current progress with the plan, budget, and scope.
- Final Review: Evaluate success, note lessons learned, and give recommendations.

5.5 Project Termination

Closure Steps:

- Check all deliverables to ensure goals are met.
- Get approval from stakeholders to close the project.
- Store all project documents for future reference.

Lessons Learned:

- Record what worked, what didn't, and areas to improve.
- Discuss lessons with the team for future projects.

After the Project:

- Create a plan for maintaining and updating "Brain Buddies."
- Train a support team to handle ongoing operations.

6. RISK ASSESSMENT MANAGEMENT PLAN

6.1 Identification and Analysis of Risks

Effective risk management is essential for the success of the Brain Buddies project. Below is a structured approach to identifying, analyzing, and addressing potential risks:

User Engagement Risks

- Low User Adoption: Students may not find the platform engaging or relevant.
- Insufficient User Feedback: Difficulty in collecting meaningful feedback to improve features.

System Functionality Risks

- Matching Algorithm Failure: Inaccurate or irrelevant matches may lead to dissatisfaction.
- Integration Issues: Difficulty integrating third-party APIs for features like SSO or calendar syncing.

Technical Risks

- Server Downtime: Outages can disrupt user access.
- Security Vulnerabilities: Risk of data breaches or unauthorized access to user information.

Team-Related Risks

- Resource Shortages: Insufficient skilled team members or delays in hiring.
- Communication Gaps: Misunderstandings or lack of coordination among team members.

Financial Risks

- Budget Overruns: Unexpected expenses during development or deployment.
- Underfunding: Limited budget for ongoing maintenance or updates.

Compliance Risks

- Data Privacy Violations: Non-compliance with FERPA or other data protection regulations.
- Legal Issues: Challenges with intellectual property or licensing of third-party tools.

Usability Risks

- Complex User Interface: Difficulty in navigating the app may discourage users.
- Accessibility Issues: Failure to accommodate users with disabilities.

6.2 Risk Register

Risk Description	Likelihood	Impact	Rank	Action Plan
Low user adoption	Medium	High	4	Conduct user surveys, promote the app through university channels, and engage students in beta testing.
Matching algorithm failure	Medium	High	4	Test the algorithm with real-world data and refine it based on feedback.
Server downtime	Low	High	1	Use cloud platforms with uptime guarantees and implement a disaster recovery plan.
Security vulnerabilities	Medium	High	2	Implement encryption, MFA, and regular security audits.
Budget overruns	Medium	Medium	3	Maintain a detailed budget and allocate contingency funds.
Resource shortages	Low	Medium	1	Crosstrain team members and maintain a pool of backup resources.
Accessibility issues	Medium	Medium	3	Follow WCAG 2.1 guidelines and conduct usability testing with diverse user groups.
Integration Issues	Medium	Medium	3	Schedule buffer time for integration testing and ensure API compatibility during development.
Compliance violations	Low	High	3	Work with legal experts to ensure FERPA compliance and other regulations.
Insufficient user feedback	Medium	Medium	3	Use in-app surveys and analytics tools to gather user feedback regularly.

6.3 Risk Mitigation and Monitoring

Risks will be regularly monitored throughout the project lifecycle to ensure timely identification and response. A risk management framework will be implemented with regular updates to the risk register. The project team will hold weekly risk reviews to assess new risks and track the mitigation progress for identified ones.

By proactively addressing risks, the Brain Buddies project aims to minimize disruptions and ensure successful delivery.

7. FINANCIAL PLAN WITH BUDGET

6.1 High-Level Details

The "Brain Buddies" project budget is planned to cover all necessary stages, guaranteeing effective resource utilization. Key stages including planning, design, development (frontend and backend), quality assurance, deployment, and continuing maintenance are all included in the $109,720.00 labor cost.

We have set aside $17,600.00 for software subscription expenses in addition to manpower, which include necessary tools and platforms for communication, development, and project administration.

Phase	**Total Phase Cost**
Planning	$32,480.00
Design	$12,640.00
Backend	$21,200.00
Frontend	$13,960.00
Quality Assurance	$18,160.00
Deployment and Launch	$10,560.00
Ongoing Maintenance and Support	$18,320.00
Total Labor Cost	$109,720.00
Software Subscription Cost	$17,600.00
Contingency Cost	$8,912.00

Refer to Appendix D for a detailed breakdown of costs.

6.2 Budget Justification

The budget for the "Brain Buddies" project has been carefully designed to balance costs with expected benefits. A Cost-Benefit Analysis (CBA) was conducted to evaluate the project's value. The study compared the anticipated costs of the project phases with the expected benefits, including improved student collaboration, time savings, and academic performance enhancements.

Total Cost:

- The total project cost is $136,232.00, broken down as follows:
  - Software subscriptions: $17,600.00
  - Labor: $109,720.00

Expected Benefits:

- The projected benefits include:
  - Increased efficiency in student collaboration, reducing time spent finding study partners.
  - Improved academic performance by streamlining group projects.

Benefit-Cost Ratio (BCR) Calculation:

The Benefit-Cost Ratio (BCR) is calculated as:
BCR=Total Anticipated Benefits / Total Project Costs

For this project:

- Total Anticipated Benefits = $120,000 (estimated value of time savings, improved collaboration, and academic outcomes).
- Total Project Costs = $136,232.
- BCR=120,000/136,232=0.88

A BCR of less than 1 (0.88) indicates that the immediate monetary benefits do not fully cover the costs, suggesting that the project may appear as a loss in its initial phase.

Strategic Value and Long-Term Potential:

While the BCR indicates a potential short-term loss, "Brain Buddies" has significant strategic value:

- Educational Impact: The app is designed for student connectivity and educational purposes, aiming to foster collaboration and improve academic outcomes.
- Monetization Opportunities: In the future, the app could generate revenue through:
  - Advertisements targeting students and educational services.
  - Premium features for enhanced functionality.

Conclusion:

The short-term financial metrics do not fully reflect the long-term potential of "Brain Buddies." By addressing an essential need in student connectivity and academic collaboration, this project lays the groundwork for future monetization opportunities and broader societal benefits. Therefore, the investment is justified despite the current BCR.

7 TEAM CREDENTIALS

Shreyas Shashikant Banait

Shreyas is a graduate student pursuing a master's in engineering management at Northeastern University. He contributed significantly to financial budget planning and creating the technical overview. Additionally, Shreyas played a role in other parts of the project, supporting team efforts as needed.

Utkarsh Gupta

Utkarsh is a Master's student at Northeastern University with three years of experience as an RPA and AI developer. His major contributions included assisting in schedule development. Utkarsh also helped in other areas of the project, ensuring its successful completion.

Sai Chandra Munagala

Sai Chandra is a graduate student pursuing a Master's in Energy Systems at Northeastern University. He played a key role in helping with the schedule and Work Breakdown Structure (WBS). Sai Chandra supported other parts of the project, contributing to its overall success.

Priyank Shah

Priyank is a master's student in Engineering Management at Northeastern University, focusing on Quality Assurance and Quality Control. He contributed to the schedule and documentation while providing support for other parts of the project to ensure a smooth execution.

Md Zafer Shamim

Zafer is pursuing a master's in engineering management at Northeastern University, with an interest in supply chain management. He took the lead in creating the financial budget, RACI chart, Work Breakdown Structure (WBS), and the project presentation. Zafer also assisted in other areas of the project as needed.

Akash Shanmuganathan

Akash is a graduate student in Engineering Management at Northeastern University with over two years of experience as a Quality Assurance and Software Engineer. His contributions included developing the Work Breakdown Structure (WBS), execution plan, implementation plan, documentation, and presentation. Akash also supported other aspects of the project as required.

Shiekh Talib

Talib is a graduate student in Engineering Management at Northeastern University with a background in Civil Engineering. He was instrumental in risk management, schedule creation, and using Microsoft Project for planning. Talib also contributed to other parts of the project, ensuring alignment with team objectives.

APPENDICES

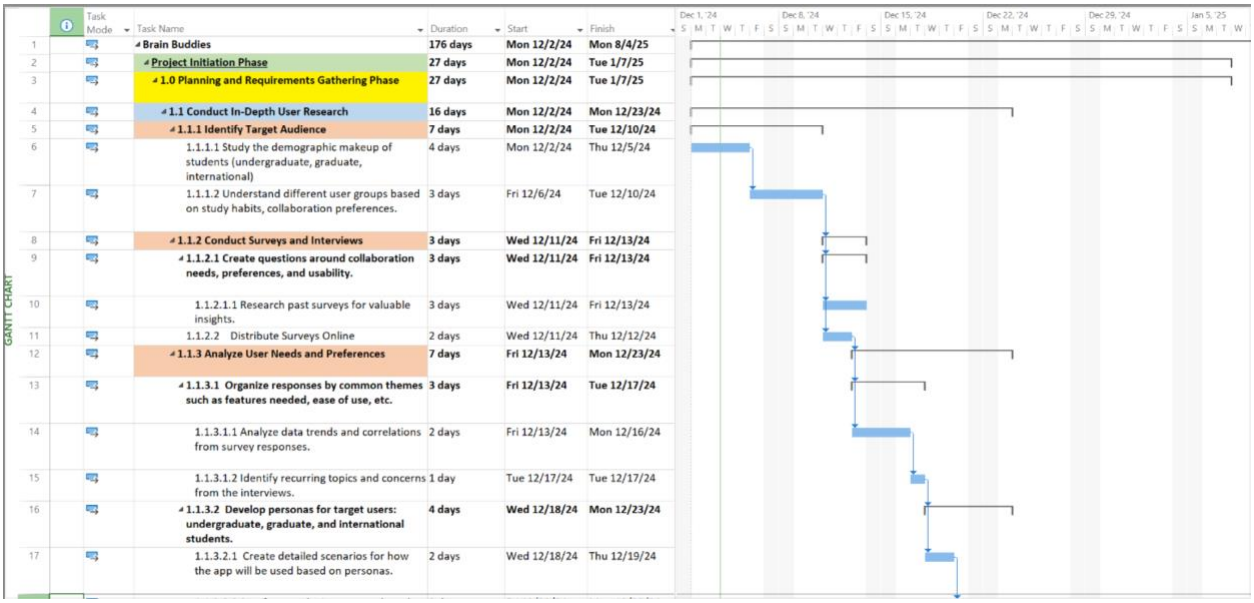
Appendix A: Work Break Down Structure (WBS)

ID	Task Name
1	**Brain Buddies**
2	**Project Initiation Phase**
3	**1.0 Planning and Requirements Gathering Phase**
4	**1.1 Conduct In-Depth User Research**
5	1.1.1 Identify Target Audience
6	1.1.2 Conduct Surveys and Interviews
7	1.1.3 Analyze User Needs and Preferences
8	**1.2 Define App Objectives**
9	1.2.1 Set Smart Goals for App
10	1.2.2 Identify Core Features
11	1.2.3 Determine Key Performance Indicator
12	**1.3 Create Detailed Requirement Document**
13	1.3.1 Define Functional Requirements
14	1.3.2 Define Non-Functional Requirements
15	1.3.3 Develop Technical Specifications
16	**Project Planning Phase**
17	**2.0 Design and Prototyping Phase**
18	**2.1 Develop Wireframes**
19	2.1.1 Sketch Basic Layouts
20	2.1.2 Create Initial Wireframe Drafts
21	2.1.3 Refine Wireframes
22	**2.2 Create Interactive Prototype**
23	2.2.1 Develop Clickable Prototypes
24	2.2.2 Conduct Prototype Usability Testing
25	2.2.3 Iterate Based on Testing Feedback
26	**2.3 Refine Design Based on Feedback**
27	2.3.1 Gather Feedback from Stakeholders
28	2.3.2 Adjust Visual and UX Design
29	2.3.3 Finalize Design
30	**Project Execution Phase**
31	**3.0 Backend Development Phase**
32	**3.1 Choose a Cloud Platform**
33	3.1.1 Evaluate Cloud Providers
34	3.1.2 Select Cloud Storage and Database Options
35	**3.2 Implement User Authentication**
36	3.2.1 Design User Authentication System
37	3.2.2 Integrate Auth or SSO

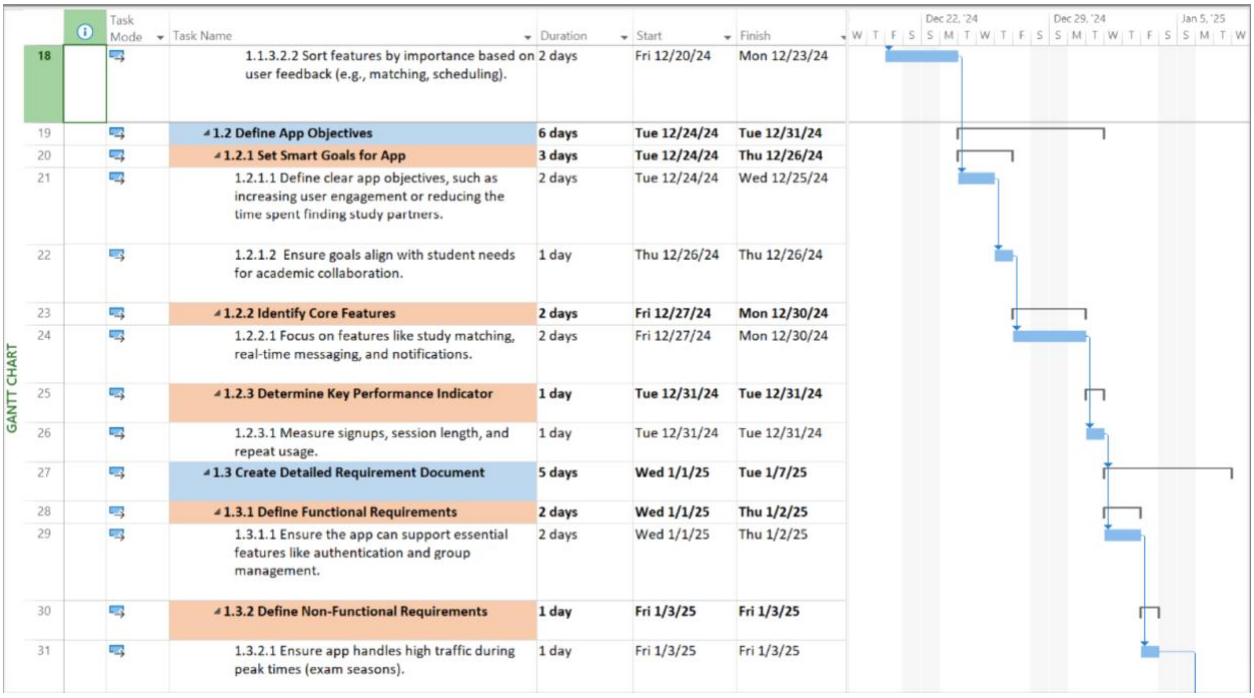
38	**3.3 Develop Core Features**
39	3.3.1 Implement Key Backend Functionalities
40	3.3.2 Ensure Data Security and Privacy
41	3.3.3 Integrate Backend Services
42	**4.0 Frontend Development Phase**
43	**4.1 Develop the User Interface**
44	4.1.1 Create UI Components
45	4.1.2 Ensure Consistency Across Screens
46	**4.2 Integrate Third-Party Services**
47	4.2.1 Identify Required Third-Party APIs
48	4.2.2 Integrate and Test APIs
49	**4.3 Implement Push Notifications**
50	4.3.1 Design Notification Strategy
51	4.3.2 Integrate Notification System
52	**5.0 API Integration and Testing Phase**
53	**5.1 Connect Frontend and Backend Functionality**
54	5.1.1 Test API Endpoints
55	5.1.2 Ensure Data Flow Consistency
56	**5.2 Conduct Unit Testing**
57	5.2.1 Test Individual Components
58	5.2.2 Verify Each Component's Functionality
59	**5.3 Conduct Integration Testing**
60	5.3.1 Ensure End-to-End System Functionality
61	**5.4 Test Third-Party Integrations**
62	5.4.1 Validate External Services Integration
63	5.4.2 Ensure Error Handling
64	5.4.3 Conduct Load Testing for API Integration
65	**Project Monitoring and Control Phase**
66	**6.0 Quality Assurance and Beta Testing Phase**
67	**6.1 Conduct Thorough Testing**
68	6.1.1 Perform System Testing
69	6.1.2 Conduct Stress and Load Testing
70	**6.2 Conduct Beta Testing**
71	6.2.1 Select Beta Testers
72	6.2.2 Collect Beta Testers' Feedback
73	**6.3 Incorporate Beta Testing Feedback**
74	6.3.1 Identify and Implement Changes
75	**Project Closeout**
76	**7.0 Deployment and Launch Phase**
77	**7.1 Deploy App to App Stores**

78	7.1.1 Prepare App Store Listings
79	7.1.2 Submit App for Approval
80	**7.2 Monitor User Interactions**
81	7.2.1 Collect Usage Data
82	7.2.2 Analyze User Engagement
83	**7.3 Provide Post-Launch Support**
84	7.3.1 Address Immediate Bug Reports
85	**8.0 Ongoing Maintenance and Support Phase**
86	**8.1 Provide Continuous Support to User**
87	8.1.1 Offer In-App Support Channels
88	8.1.2 Maintain Responsive Helpdesk
89	**8.2 Deliver Regular Updates**
90	8.2.1 Schedule Feature Updates
91	8.2.2 Perform Bug Fixes and Patches
92	**8.3 Monitor Performance**
93	8.3.1 Track App Stability Metrics
94	**8.4 Gather and Analyze User Feedback**
95	8.4.1 Conduct User Surveys
96	8.4.2 Implement Feedback Analysis
97	**7.2 Monitor User Interactions**
98	7.2.1 Collect Usage Data
99	7.2.2 Analyze User Engagement
100	**7.3 Provide Post-Launch Support**
101	7.3.1 Address Immediate Bug Reports
102	**8.0 Ongoing Maintenance and Support Phase**
103	**8.1 Provide Continuous Support to User**
104	8.1.1 Offer In-App Support Channels
105	8.1.2 Maintain Responsive Helpdesk
106	**8.2 Deliver Regular Updates**
107	8.2.1 Schedule Feature Updates
108	8.2.2 Perform Bug Fixes and Patches
109	**8.3 Monitor Performance**
110	8.3.1 Track App Stability Metrics
111	**8.4 Gather and Analyze User Feedback**
112	8.4.1 Conduct User Surveys
113	8.4.2 Implement Feedback Analysis

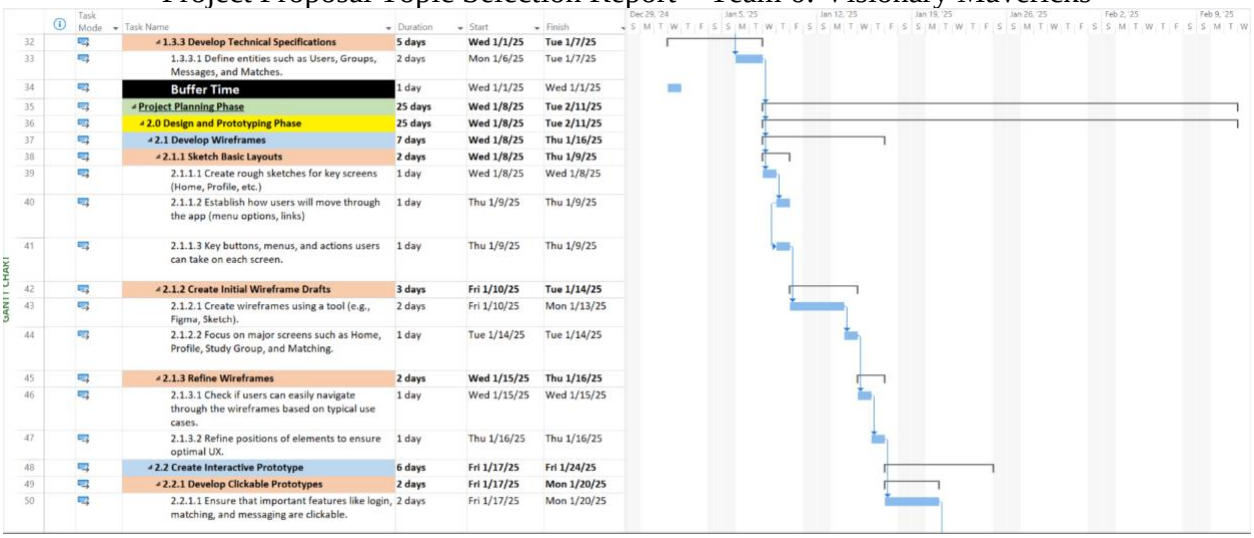
Appendix B: Schedule



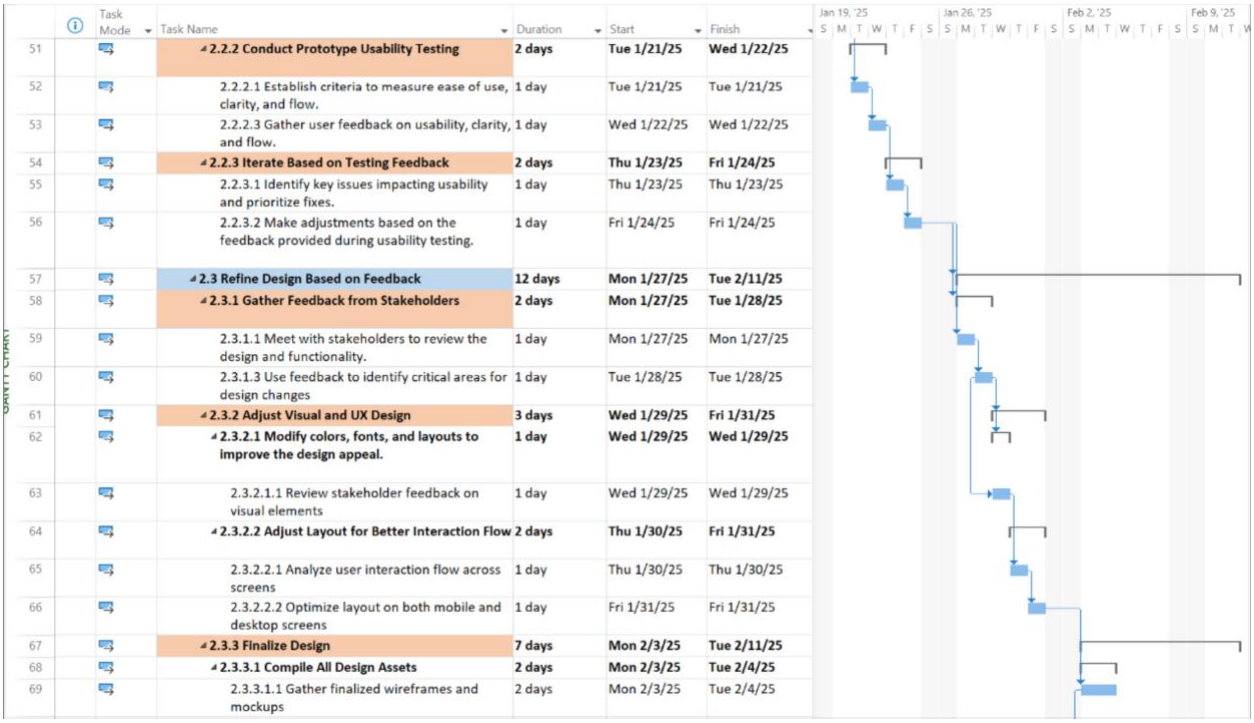
	Task Mode	Task Name	Duration	Start	Finish
1		**Brain Buddies**	**176 days**	**Mon 12/2/24**	**Mon 8/4/25**
2		**Project Initiation Phase**	**27 days**	**Mon 12/2/24**	**Tue 1/7/25**
3		**1.0 Planning and Requirements Gathering Phase**	**27 days**	**Mon 12/2/24**	**Tue 1/7/25**
4		**1.1 Conduct In-Depth User Research**	**16 days**	**Mon 12/2/24**	**Mon 12/23/24**
5		**1.1.1 Identify Target Audience**	**7 days**	**Mon 12/2/24**	**Tue 12/10/24**
6		1.1.1.1 Study the demographic makeup of students (undergraduate, graduate, international)	4 days	Mon 12/2/24	Thu 12/5/24
7		1.1.1.2 Understand different user groups based on study habits, collaboration preferences.	3 days	Fri 12/6/24	Tue 12/10/24
8		**1.1.2 Conduct Surveys and Interviews**	**3 days**	**Wed 12/11/24**	**Fri 12/13/24**
9		**1.1.2.1 Create questions around collaboration needs, preferences, and usability.**	**3 days**	**Wed 12/11/24**	**Fri 12/13/24**
10		1.1.2.1.1 Research past surveys for valuable insights.	3 days	Wed 12/11/24	Fri 12/13/24
11		1.1.2.2 Distribute Surveys Online	2 days	Wed 12/11/24	Thu 12/12/24
12		**1.1.3 Analyze User Needs and Preferences**	**7 days**	**Fri 12/13/24**	**Mon 12/23/24**
13		**1.1.3.1 Organize responses by common themes such as features needed, ease of use, etc.**	**3 days**	**Fri 12/13/24**	**Tue 12/17/24**
14		1.1.3.1.1 Analyze data trends and correlations from survey responses.	2 days	Fri 12/13/24	Mon 12/16/24
15		1.1.3.1.2 Identify recurring topics and concerns from the interviews.	1 day	Tue 12/17/24	Tue 12/17/24
16		**1.1.3.2 Develop personas for target users: undergraduate, graduate, and international students.**	**4 days**	**Wed 12/18/24**	**Mon 12/23/24**
17		1.1.3.2.1 Create detailed scenarios for how the app will be used based on personas.	2 days	Wed 12/18/24	Thu 12/19/24



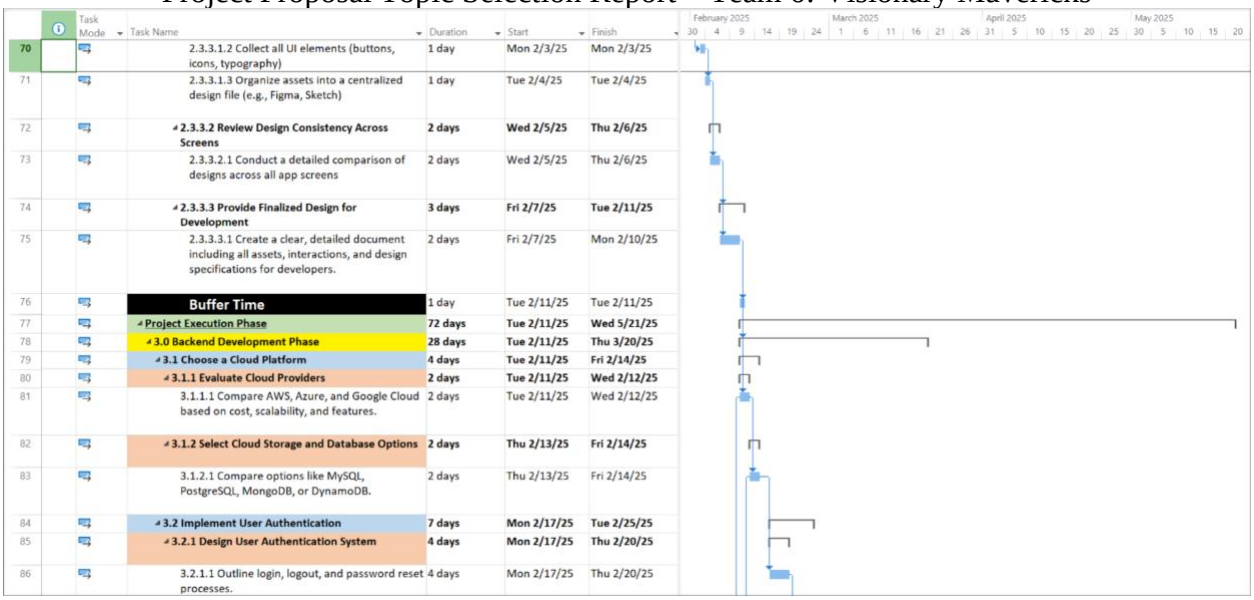
	Task Mode	Task Name	Duration	Start	Finish
18		1.1.3.2.2 Sort features by importance based on user feedback (e.g., matching, scheduling).	2 days	Fri 12/20/24	Mon 12/23/24
19		**1.2 Define App Objectives**	**6 days**	**Tue 12/24/24**	**Tue 12/31/24**
20		**1.2.1 Set Smart Goals for App**	**3 days**	**Tue 12/24/24**	**Thu 12/26/24**
21		1.2.1.1 Define clear app objectives, such as increasing user engagement or reducing the time spent finding study partners.	2 days	Tue 12/24/24	Wed 12/25/24
22		1.2.1.2 Ensure goals align with student needs for academic collaboration.	1 day	Thu 12/26/24	Thu 12/26/24
23		**1.2.2 Identify Core Features**	**2 days**	**Fri 12/27/24**	**Mon 12/30/24**
24		1.2.2.1 Focus on features like study matching, real-time messaging, and notifications.	2 days	Fri 12/27/24	Mon 12/30/24
25		**1.2.3 Determine Key Performance Indicator**	**1 day**	**Tue 12/31/24**	**Tue 12/31/24**
26		1.2.3.1 Measure signups, session length, and repeat usage.	1 day	Tue 12/31/24	Tue 12/31/24
27		**1.3 Create Detailed Requirement Document**	**5 days**	**Wed 1/1/25**	**Tue 1/7/25**
28		**1.3.1 Define Functional Requirements**	**2 days**	**Wed 1/1/25**	**Thu 1/2/25**
29		1.3.1.1 Ensure the app can support essential features like authentication and group management.	2 days	Wed 1/1/25	Thu 1/2/25
30		**1.3.2 Define Non-Functional Requirements**	**1 day**	**Fri 1/3/25**	**Fri 1/3/25**
31		1.3.2.1 Ensure app handles high traffic during peak times (exam seasons).	1 day	Fri 1/3/25	Fri 1/3/25



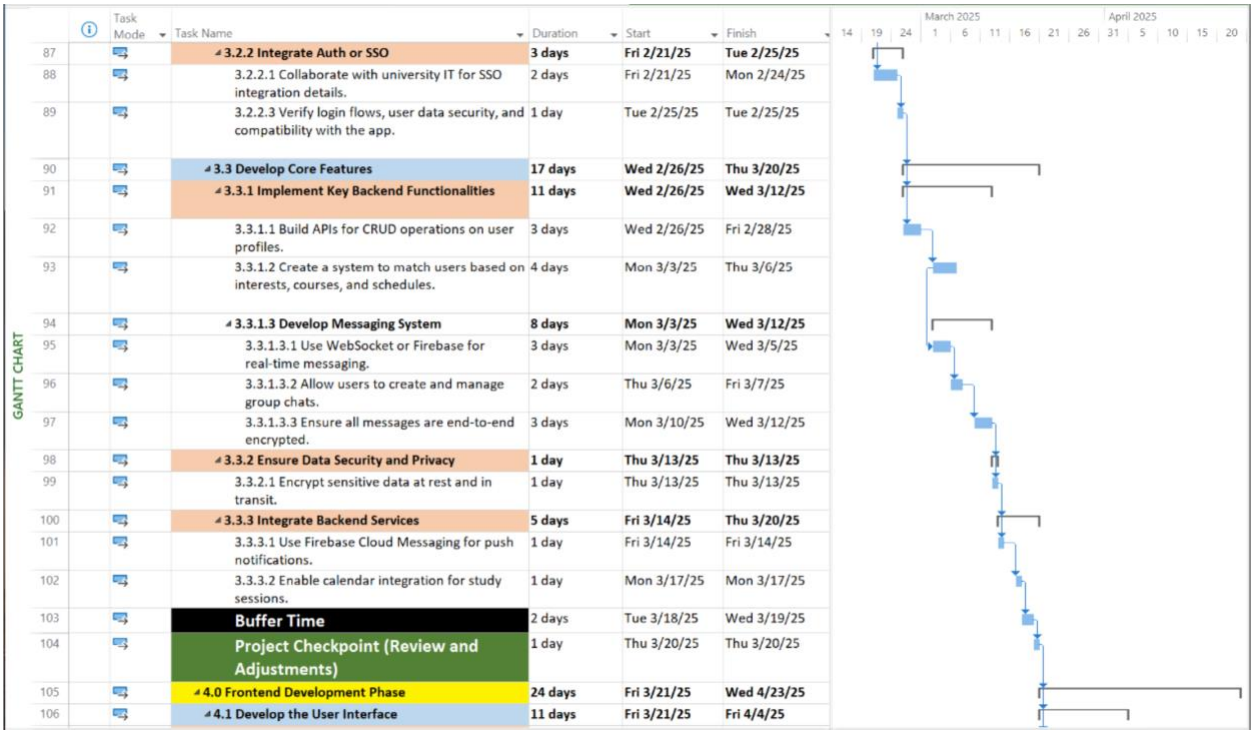
	Task Mode	Task Name	Duration	Start	Finish
32		**1.3.3 Develop Technical Specifications**	**5 days**	**Wed 1/1/25**	**Tue 1/7/25**
33		1.3.3.1 Define entities such as Users, Groups, Messages, and Matches.	2 days	Mon 1/6/25	Tue 1/7/25
34		**Buffer Time**	1 day	Wed 1/1/25	Wed 1/1/25
35		**Project Planning Phase**	**25 days**	**Wed 1/8/25**	**Tue 2/11/25**
36		**2.0 Design and Prototyping Phase**	**25 days**	**Wed 1/8/25**	**Tue 2/11/25**
37		**2.1 Develop Wireframes**	**7 days**	**Wed 1/8/25**	**Thu 1/16/25**
38		**2.1.1 Sketch Basic Layouts**	**2 days**	**Wed 1/8/25**	**Thu 1/9/25**
39		2.1.1.1 Create rough sketches for key screens (Home, Profile, etc.)	1 day	Wed 1/8/25	Wed 1/8/25
40		2.1.1.2 Establish how users will move through the app (menu options, links)	1 day	Thu 1/9/25	Thu 1/9/25
41		2.1.1.3 Key buttons, menus, and actions users can take on each screen.	1 day	Thu 1/9/25	Thu 1/9/25
42		**2.1.2 Create Initial Wireframe Drafts**	**3 days**	**Fri 1/10/25**	**Tue 1/14/25**
43		2.1.2.1 Create wireframes using a tool (e.g., Figma, Sketch).	2 days	Fri 1/10/25	Mon 1/13/25
44		2.1.2.2 Focus on major screens such as Home, Profile, Study Group, and Matching.	1 day	Tue 1/14/25	Tue 1/14/25
45		**2.1.3 Refine Wireframes**	**2 days**	**Wed 1/15/25**	**Thu 1/16/25**
46		2.1.3.1 Check if users can easily navigate through the wireframes based on typical use cases.	1 day	Wed 1/15/25	Wed 1/15/25
47		2.1.3.2 Refine positions of elements to ensure optimal UX.	1 day	Thu 1/16/25	Thu 1/16/25
48		**2.2 Create Interactive Prototype**	**6 days**	**Fri 1/17/25**	**Fri 1/24/25**
49		**2.2.1 Develop Clickable Prototypes**	**2 days**	**Fri 1/17/25**	**Mon 1/20/25**
50		2.2.1.1 Ensure that important features like login, matching, and messaging are clickable.	2 days	Fri 1/17/25	Mon 1/20/25



	Task Mode	Task Name	Duration	Start	Finish
51		**2.2.2 Conduct Prototype Usability Testing**	**2 days**	**Tue 1/21/25**	**Wed 1/22/25**
52		2.2.2.1 Establish criteria to measure ease of use, clarity, and flow.	1 day	Tue 1/21/25	Tue 1/21/25
53		2.2.2.3 Gather user feedback on usability, clarity, and flow.	1 day	Wed 1/22/25	Wed 1/22/25
54		**2.2.3 Iterate Based on Testing Feedback**	**2 days**	**Thu 1/23/25**	**Fri 1/24/25**
55		2.2.3.1 Identify key issues impacting usability and prioritize fixes.	1 day	Thu 1/23/25	Thu 1/23/25
56		2.2.3.2 Make adjustments based on the feedback provided during usability testing.	1 day	Fri 1/24/25	Fri 1/24/25
57		**2.3 Refine Design Based on Feedback**	**12 days**	**Mon 1/27/25**	**Tue 2/11/25**
58		**2.3.1 Gather Feedback from Stakeholders**	**2 days**	**Mon 1/27/25**	**Tue 1/28/25**
59		2.3.1.1 Meet with stakeholders to review the design and functionality.	1 day	Mon 1/27/25	Mon 1/27/25
60		2.3.1.3 Use feedback to identify critical areas for design changes	1 day	Tue 1/28/25	Tue 1/28/25
61		**2.3.2 Adjust Visual and UX Design**	**3 days**	**Wed 1/29/25**	**Fri 1/31/25**
62		**2.3.2.1 Modify colors, fonts, and layouts to improve the design appeal.**	**1 day**	**Wed 1/29/25**	**Wed 1/29/25**
63		2.3.2.1.1 Review stakeholder feedback on visual elements	1 day	Wed 1/29/25	Wed 1/29/25
64		**2.3.2.2 Adjust Layout for Better Interaction Flow**	**2 days**	**Thu 1/30/25**	**Fri 1/31/25**
65		2.3.2.2.1 Analyze user interaction flow across screens	1 day	Thu 1/30/25	Thu 1/30/25
66		2.3.2.2.2 Optimize layout on both mobile and desktop screens	1 day	Fri 1/31/25	Fri 1/31/25
67		**2.3.3 Finalize Design**	**7 days**	**Mon 2/3/25**	**Tue 2/11/25**
68		**2.3.3.1 Compile All Design Assets**	**2 days**	**Mon 2/3/25**	**Tue 2/4/25**
69		2.3.3.1.1 Gather finalized wireframes and mockups	2 days	Mon 2/3/25	Tue 2/4/25

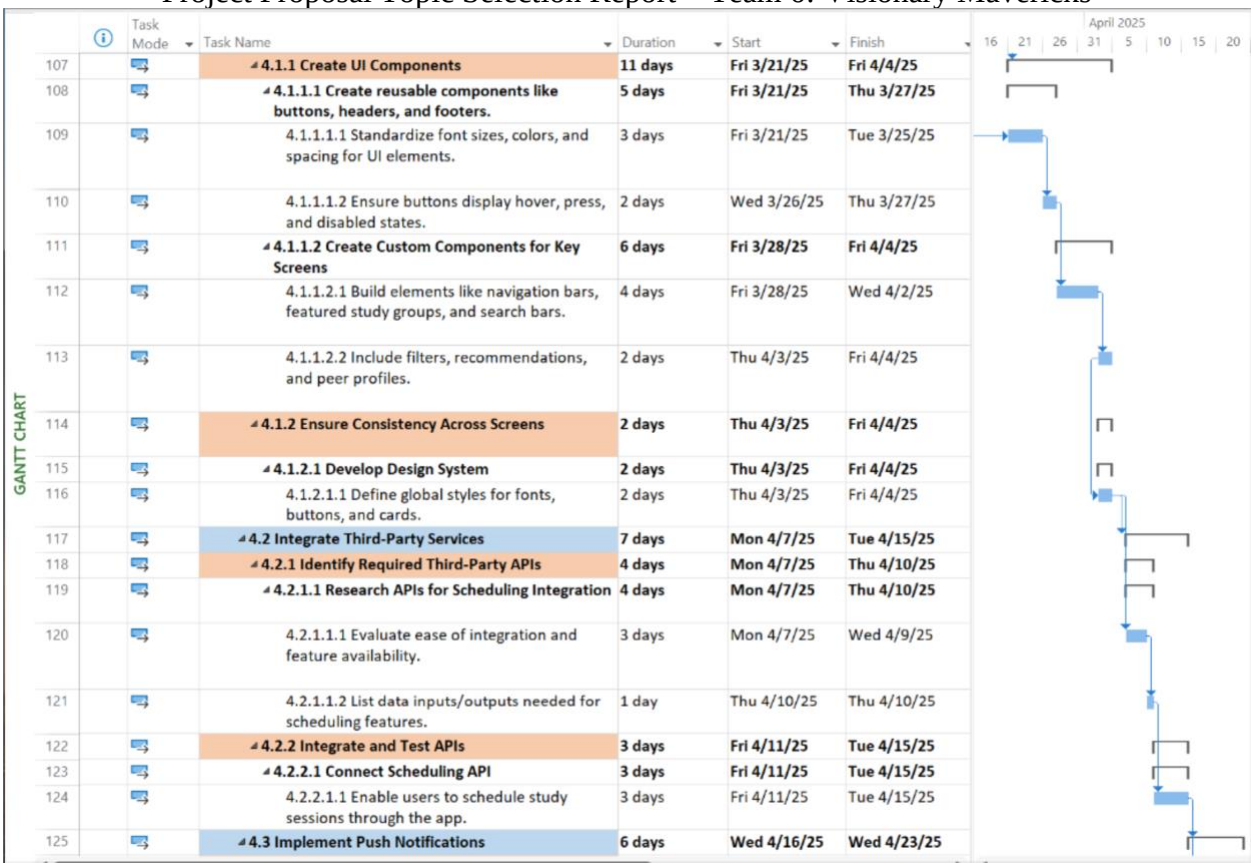


	Task Mode	Task Name	Duration	Start	Finish
70		2.3.3.1.2 Collect all UI elements (buttons, icons, typography)	1 day	Mon 2/3/25	Mon 2/3/25
71		2.3.3.1.3 Organize assets into a centralized design file (e.g., Figma, Sketch)	1 day	Tue 2/4/25	Tue 2/4/25
72		**2.3.3.2 Review Design Consistency Across Screens**	**2 days**	**Wed 2/5/25**	**Thu 2/6/25**
73		2.3.3.2.1 Conduct a detailed comparison of designs across all app screens	2 days	Wed 2/5/25	Thu 2/6/25
74		**2.3.3.3 Provide Finalized Design for Development**	**3 days**	**Fri 2/7/25**	**Tue 2/11/25**
75		2.3.3.3.1 Create a clear, detailed document including all assets, interactions, and design specifications for developers.	2 days	Fri 2/7/25	Mon 2/10/25
76		**Buffer Time**	1 day	Tue 2/11/25	Tue 2/11/25
77		**Project Execution Phase**	**72 days**	**Tue 2/11/25**	**Wed 5/21/25**
78		**3.0 Backend Development Phase**	**28 days**	**Tue 2/11/25**	**Thu 3/20/25**
79		**3.1 Choose a Cloud Platform**	**4 days**	**Tue 2/11/25**	**Fri 2/14/25**
80		**3.1.1 Evaluate Cloud Providers**	**2 days**	**Tue 2/11/25**	**Wed 2/12/25**
81		3.1.1.1 Compare AWS, Azure, and Google Cloud based on cost, scalability, and features.	2 days	Tue 2/11/25	Wed 2/12/25
82		**3.1.2 Select Cloud Storage and Database Options**	**2 days**	**Thu 2/13/25**	**Fri 2/14/25**
83		3.1.2.1 Compare options like MySQL, PostgreSQL, MongoDB, or DynamoDB.	2 days	Thu 2/13/25	Fri 2/14/25
84		**3.2 Implement User Authentication**	**7 days**	**Mon 2/17/25**	**Tue 2/25/25**
85		**3.2.1 Design User Authentication System**	**4 days**	**Mon 2/17/25**	**Thu 2/20/25**
86		3.2.1.1 Outline login, logout, and password reset processes.	4 days	Mon 2/17/25	Thu 2/20/25



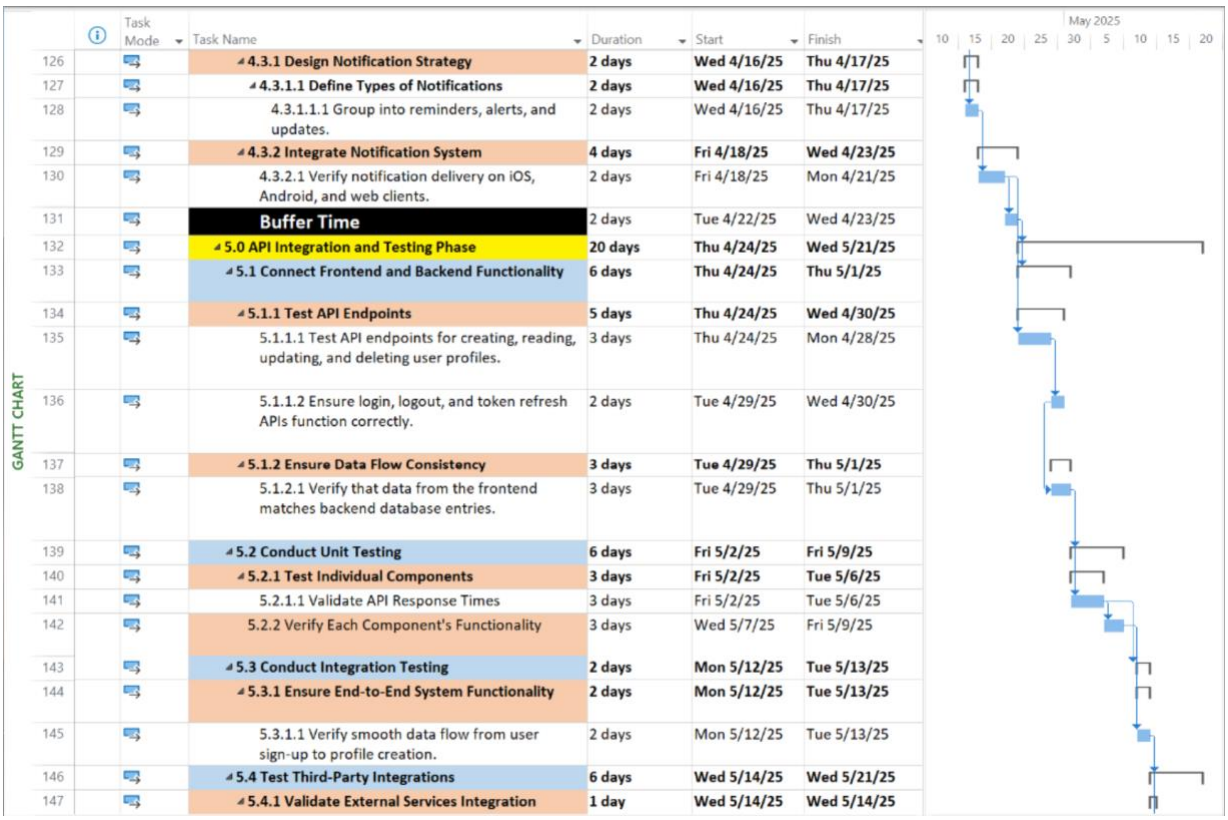
GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
87		**3.2.2 Integrate Auth or SSO**	**3 days**	**Fri 2/21/25**	**Tue 2/25/25**
88		3.2.2.1 Collaborate with university IT for SSO integration details.	2 days	Fri 2/21/25	Mon 2/24/25
89		3.2.2.3 Verify login flows, user data security, and compatibility with the app.	1 day	Tue 2/25/25	Tue 2/25/25
90		**3.3 Develop Core Features**	**17 days**	**Wed 2/26/25**	**Thu 3/20/25**
91		**3.3.1 Implement Key Backend Functionalities**	**11 days**	**Wed 2/26/25**	**Wed 3/12/25**
92		3.3.1.1 Build APIs for CRUD operations on user profiles.	3 days	Wed 2/26/25	Fri 2/28/25
93		3.3.1.2 Create a system to match users based on interests, courses, and schedules.	4 days	Mon 3/3/25	Thu 3/6/25
94		**3.3.1.3 Develop Messaging System**	**8 days**	**Mon 3/3/25**	**Wed 3/12/25**
95		3.3.1.3.1 Use WebSocket or Firebase for real-time messaging.	3 days	Mon 3/3/25	Wed 3/5/25
96		3.3.1.3.2 Allow users to create and manage group chats.	2 days	Thu 3/6/25	Fri 3/7/25
97		3.3.1.3.3 Ensure all messages are end-to-end encrypted.	3 days	Mon 3/10/25	Wed 3/12/25
98		**3.3.2 Ensure Data Security and Privacy**	**1 day**	**Thu 3/13/25**	**Thu 3/13/25**
99		3.3.2.1 Encrypt sensitive data at rest and in transit.	1 day	Thu 3/13/25	Thu 3/13/25
100		**3.3.3 Integrate Backend Services**	**5 days**	**Fri 3/14/25**	**Thu 3/20/25**
101		3.3.3.1 Use Firebase Cloud Messaging for push notifications.	1 day	Fri 3/14/25	Fri 3/14/25
102		3.3.3.2 Enable calendar integration for study sessions.	1 day	Mon 3/17/25	Mon 3/17/25
103		**Buffer Time**	2 days	Tue 3/18/25	Wed 3/19/25
104		**Project Checkpoint (Review and Adjustments)**	1 day	Thu 3/20/25	Thu 3/20/25
105		**4.0 Frontend Development Phase**	**24 days**	**Fri 3/21/25**	**Wed 4/23/25**
106		**4.1 Develop the User Interface**	**11 days**	**Fri 3/21/25**	**Fri 4/4/25**



GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
107		**4.1.1 Create UI Components**	**11 days**	**Fri 3/21/25**	**Fri 4/4/25**
108		**4.1.1.1 Create reusable components like buttons, headers, and footers.**	**5 days**	**Fri 3/21/25**	**Thu 3/27/25**
109		4.1.1.1.1 Standardize font sizes, colors, and spacing for UI elements.	3 days	Fri 3/21/25	Tue 3/25/25
110		4.1.1.1.2 Ensure buttons display hover, press, and disabled states.	2 days	Wed 3/26/25	Thu 3/27/25
111		**4.1.1.2 Create Custom Components for Key Screens**	**6 days**	**Fri 3/28/25**	**Fri 4/4/25**
112		4.1.1.2.1 Build elements like navigation bars, featured study groups, and search bars.	4 days	Fri 3/28/25	Wed 4/2/25
113		4.1.1.2.2 Include filters, recommendations, and peer profiles.	2 days	Thu 4/3/25	Fri 4/4/25
114		**4.1.2 Ensure Consistency Across Screens**	**2 days**	**Thu 4/3/25**	**Fri 4/4/25**
115		**4.1.2.1 Develop Design System**	**2 days**	**Thu 4/3/25**	**Fri 4/4/25**
116		4.1.2.1.1 Define global styles for fonts, buttons, and cards.	2 days	Thu 4/3/25	Fri 4/4/25
117		**4.2 Integrate Third-Party Services**	**7 days**	**Mon 4/7/25**	**Tue 4/15/25**
118		**4.2.1 Identify Required Third-Party APIs**	**4 days**	**Mon 4/7/25**	**Thu 4/10/25**
119		**4.2.1.1 Research APIs for Scheduling Integration**	**4 days**	**Mon 4/7/25**	**Thu 4/10/25**
120		4.2.1.1.1 Evaluate ease of integration and feature availability.	3 days	Mon 4/7/25	Wed 4/9/25
121		4.2.1.1.2 List data inputs/outputs needed for scheduling features.	1 day	Thu 4/10/25	Thu 4/10/25
122		**4.2.2 Integrate and Test APIs**	**3 days**	**Fri 4/11/25**	**Tue 4/15/25**
123		**4.2.2.1 Connect Scheduling API**	**3 days**	**Fri 4/11/25**	**Tue 4/15/25**
124		4.2.2.1.1 Enable users to schedule study sessions through the app.	3 days	Fri 4/11/25	Tue 4/15/25
125		**4.3 Implement Push Notifications**	**6 days**	**Wed 4/16/25**	**Wed 4/23/25**

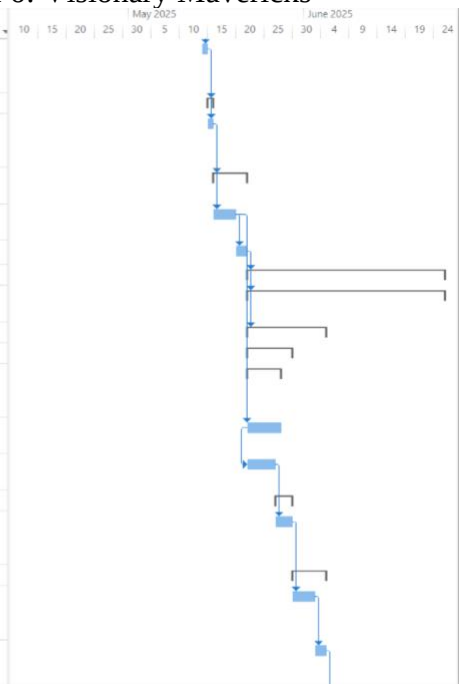


GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
126		**4.3.1 Design Notification Strategy**	**2 days**	**Wed 4/16/25**	**Thu 4/17/25**
127		**4.3.1.1 Define Types of Notifications**	**2 days**	**Wed 4/16/25**	**Thu 4/17/25**
128		4.3.1.1.1 Group into reminders, alerts, and updates.	2 days	Wed 4/16/25	Thu 4/17/25
129		**4.3.2 Integrate Notification System**	**4 days**	**Fri 4/18/25**	**Wed 4/23/25**
130		4.3.2.1 Verify notification delivery on iOS, Android, and web clients.	2 days	Fri 4/18/25	Mon 4/21/25
131		**Buffer Time**	2 days	Tue 4/22/25	Wed 4/23/25
132		**5.0 API Integration and Testing Phase**	**20 days**	**Thu 4/24/25**	**Wed 5/21/25**
133		**5.1 Connect Frontend and Backend Functionality**	**6 days**	**Thu 4/24/25**	**Thu 5/1/25**
134		**5.1.1 Test API Endpoints**	**5 days**	**Thu 4/24/25**	**Wed 4/30/25**
135		5.1.1.1 Test API endpoints for creating, reading, updating, and deleting user profiles.	3 days	Thu 4/24/25	Mon 4/28/25
136		5.1.1.2 Ensure login, logout, and token refresh APIs function correctly.	2 days	Tue 4/29/25	Wed 4/30/25
137		**5.1.2 Ensure Data Flow Consistency**	**3 days**	**Tue 4/29/25**	**Thu 5/1/25**
138		5.1.2.1 Verify that data from the frontend matches backend database entries.	3 days	Tue 4/29/25	Thu 5/1/25
139		**5.2 Conduct Unit Testing**	**6 days**	**Fri 5/2/25**	**Fri 5/9/25**
140		**5.2.1 Test Individual Components**	**3 days**	**Fri 5/2/25**	**Tue 5/6/25**
141		5.2.1.1 Validate API Response Times	3 days	Fri 5/2/25	Tue 5/6/25
142		5.2.2 Verify Each Component's Functionality	3 days	Wed 5/7/25	Fri 5/9/25
143		**5.3 Conduct Integration Testing**	**2 days**	**Mon 5/12/25**	**Tue 5/13/25**
144		**5.3.1 Ensure End-to-End System Functionality**	**2 days**	**Mon 5/12/25**	**Tue 5/13/25**
145		5.3.1.1 Verify smooth data flow from user sign-up to profile creation.	2 days	Mon 5/12/25	Tue 5/13/25
146		**5.4 Test Third-Party Integrations**	**6 days**	**Wed 5/14/25**	**Wed 5/21/25**
147		**5.4.1 Validate External Services Integration**	**1 day**	**Wed 5/14/25**	**Wed 5/14/25**

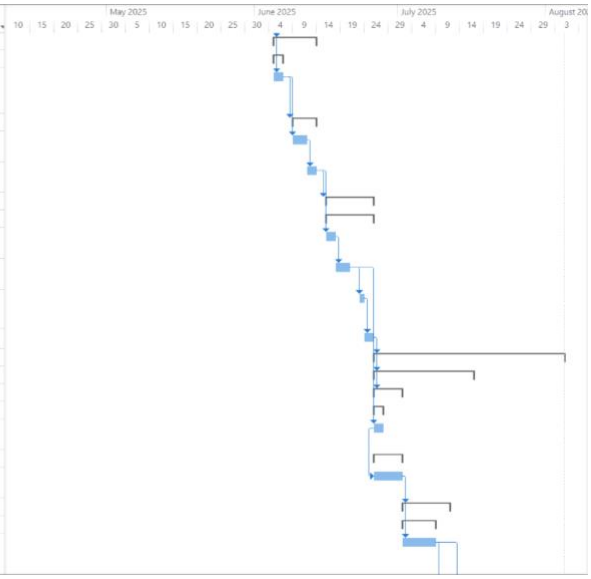
GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
148		5.4.1.1 Ensure accurate event creation and updates with Google Calendar or Outlook.	1 day	Wed 5/14/25	Wed 5/14/25
149		**5.4.2 Ensure Error Handling**	**1 day**	**Thu 5/15/25**	**Thu 5/15/25**
150		5.4.2.1 Simulate failures in external services and check for fallback handling.	1 day	Thu 5/15/25	Thu 5/15/25
151		**5.4.3 Conduct Load Testing for API Integration**	**4 days**	**Fri 5/16/25**	**Wed 5/21/25**
152		5.3.3.1 Test how the integrated system performs under heavy traffic.	2 days	Fri 5/16/25	Mon 5/19/25
153		**Buffer Time**	2 days	Tue 5/20/25	Wed 5/21/25
154		**Project Monitoring and Control Phase**	**25 days**	**Thu 5/22/25**	**Wed 6/25/25**
155		**6.0 Quality Assurance and Beta Testing Phase**	**25 days**	**Thu 5/22/25**	**Wed 6/25/25**
156		**6.1 Conduct Thorough Testing**	**10 days**	**Thu 5/22/25**	**Wed 6/4/25**
157		**6.1.1 Perform System Testing**	**6 days**	**Thu 5/22/25**	**Thu 5/29/25**
158		**6.1.1.1 Verify essential features like user login, matching system, and group creation.**	**4 days**	**Thu 5/22/25**	**Tue 5/27/25**
159		6.1.1.1.1 Ensure seamless sign-in and sign-out processes.	4 days	Thu 5/22/25	Tue 5/27/25
160		6.1.1.1.2 Verify accuracy and speed of matching suggestions.	3 days	Thu 5/22/25	Mon 5/26/25
161		**6.1.1.2 Verify Data Handling Security**	**3 days**	**Tue 5/27/25**	**Thu 5/29/25**
162		6.1.1.2.1 Ensure user data is securely encrypted during storage and transmission.	3 days	Tue 5/27/25	Thu 5/29/25
163		**6.1.2 Conduct Stress and Load Testing**	**4 days**	**Fri 5/30/25**	**Wed 6/4/25**
164		6.1.2.1 Test system performance under heavy user load (e.g., 10,000 concurrent users).	2 days	Fri 5/30/25	Mon 6/2/25
165		6.1.2.2 Ensure app remains stable during extended periods of high activity.	2 days	Tue 6/3/25	Wed 6/4/25



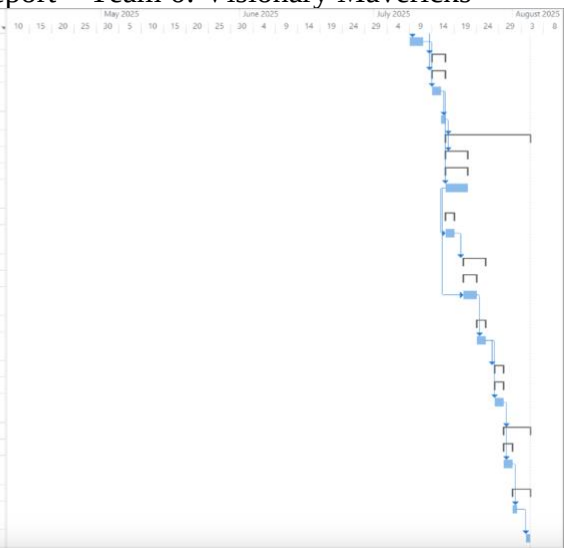
GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
166		**6.2 Conduct Beta Testing**	**7 days**	**Thu 6/5/25**	**Fri 6/13/25**
167		**6.2.1 Select Beta Testers**	**2 days**	**Thu 6/5/25**	**Fri 6/6/25**
168		6.2.1.1 Choose testers from various academic backgrounds and usage levels.	2 days	Thu 6/5/25	Fri 6/6/25
169		**6.2.2 Collect Beta Testers' Feedback**	**5 days**	**Mon 6/9/25**	**Fri 6/13/25**
170		6.2.2.1 Analyze interactions, feature usage, and navigation patterns.	3 days	Mon 6/9/25	Wed 6/11/25
171		6.2.2.2 Collect qualitative feedback on app usability and performance.	2 days	Thu 6/12/25	Fri 6/13/25
172		**6.3 Incorporate Beta Testing Feedback**	**8 days**	**Mon 6/16/25**	**Wed 6/25/25**
173		**6.3.1 Identify and Implement Changes**	**8 days**	**Mon 6/16/25**	**Wed 6/25/25**
174		6.3.1.1 Categorize issues into critical, high, medium, and low priority.	2 days	Mon 6/16/25	Tue 6/17/25
175		6.3.1.2 Fix issues that block core functionalities or cause crashes.	3 days	Wed 6/18/25	Fri 6/20/25
176		**Project Checkpoint 2 (Review and Adjustments)**	1 day	Mon 6/23/25	Mon 6/23/25
177		**Buffer Time**	2 days	Tue 6/24/25	Wed 6/25/25
178		**Project Closeout**	**28 days**	**Thu 6/26/25**	**Mon 8/4/25**
179		**7.0 Deployment and Launch Phase**	**15 days**	**Thu 6/26/25**	**Wed 7/16/25**
180		**7.1 Deploy App to App Stores**	**4 days**	**Thu 6/26/25**	**Tue 7/1/25**
181		**7.1.1 Prepare App Store Listings**	**2 days**	**Thu 6/26/25**	**Fri 6/27/25**
182		7.1.1.1 Write a concise and engaging app summary for stores.	2 days	Thu 6/26/25	Fri 6/27/25
183		**7.1.2 Submit App for Approval**	**4 days**	**Thu 6/26/25**	**Tue 7/1/25**
184		7.1.2.1 Ensure code meets store standards and is free of bugs.	4 days	Thu 6/26/25	Tue 7/1/25
185		**7.2 Monitor User Interactions**	**8 days**	**Wed 7/2/25**	**Fri 7/11/25**
186		**7.2.1 Collect Usage Data**	**5 days**	**Wed 7/2/25**	**Tue 7/8/25**
187		7.2.1.1 Integrate tools like Google Analytics or Firebase for real-time tracking.	5 days	Wed 7/2/25	Tue 7/8/25



GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
188		7.2.2 Analyze User Engagement	3 days	Wed 7/9/25	Fri 7/11/25
189		**7.3 Provide Post-Launch Support**	**3 days**	**Mon 7/14/25**	**Wed 7/16/25**
190		**7.3.1 Address Immediate Bug Reports**	**3 days**	**Mon 7/14/25**	**Wed 7/16/25**
191		7.3.1.1 Enable email, chat, or in-app support for bug reporting.	2 days	Mon 7/14/25	Tue 7/15/25
192		**Buffer Time**	1 day	Wed 7/16/25	Wed 7/16/25
193		**8.0 Ongoing Maintenance and Support Phase**	**13 days**	**Thu 7/17/25**	**Mon 8/4/25**
194		**8.1 Provide Continuous Support to User**	**3 days**	**Thu 7/17/25**	**Mon 7/21/25**
195		**8.1.1 Offer In-App Support Channels**	**3 days**	**Thu 7/17/25**	**Mon 7/21/25**
196		8.1.1.1 Integrate live chat functionality for real-time help.	3 days	Thu 7/17/25	Mon 7/21/25
197		**8.1.2 Maintain Responsive Helpdesk**	**2 days**	**Thu 7/17/25**	**Fri 7/18/25**
198		8.1.2.1 Assign a team to manage user queries and issues.	2 days	Thu 7/17/25	Fri 7/18/25
199		**8.2 Deliver Regular Updates**	**5 days**	**Mon 7/21/25**	**Fri 7/25/25**
200		**8.2.1 Schedule Feature Updates**	**3 days**	**Mon 7/21/25**	**Wed 7/23/25**
201		8.2.1.1 Identify and prioritize new features for updates.	3 days	Mon 7/21/25	Wed 7/23/25
202		**8.2.2 Perform Bug Fixes and Patches**	**2 days**	**Thu 7/24/25**	**Fri 7/25/25**
203		8.2.2.1 Track and analyze error patterns.	2 days	Thu 7/24/25	Fri 7/25/25
204		**8.3 Monitor Performance**	**2 days**	**Mon 7/28/25**	**Tue 7/29/25**
205		**8.3.1 Track App Stability Metrics**	**2 days**	**Mon 7/28/25**	**Tue 7/29/25**
206		8.3.1.1 Identify root causes for app crashes and resolve them.	2 days	Mon 7/28/25	Tue 7/29/25
207		**8.4 Gather and Analyze User Feedback**	**4 days**	**Wed 7/30/25**	**Mon 8/4/25**
208		**8.4.1 Conduct User Surveys**	**2 days**	**Wed 7/30/25**	**Thu 7/31/25**
209		8.4.1.1 Identify areas needing improvement based on user feedback.	2 days	Wed 7/30/25	Thu 7/31/25
210		**8.4.2 Implement Feedback Analysis**	**2 days**	**Fri 8/1/25**	**Mon 8/4/25**
211		8.4.2.1 Align future updates with user demands and expectations.	1 day	Fri 8/1/25	Fri 8/1/25
212		**Buffer Time**	1 day	Mon 8/4/25	Mon 8/4/25



Appendix C: RACI Matrix

ID	Task	Project Manager	Product Manager	Full Stack Developer	Full Stack Developer	UI/UX Designer	QA Engineer	Data Engineer
1	**0.0 Project Initiation**							
2	0.1 Define Project Scope	C	R,A					
3	0.2 Identify Key Stakeholders	C	R,A					
4	0.3 Define Success Metrics	C	R,A					
5	0.4 Create Initial Project Timeline	R			C			C
6	0.5 Develop Risk Management Plan	R,A	C		C			C
8	**1.0 Planning and Requirements Gathering Phase**							
9	1.1 Conduct In-Depth User Research		R,A					
10	1.2 Define App Objectives		R,A			C		
11	1.3 Create Detailed Requirements Document		R,A					
12	1.4 Stakeholder Identification and Mapping		R,A					
13	1.5 Stakeholder Communication Plan	R	R,A					
14	1.6 Stakeholder Feedback Collection		R,A					
15	**2.0 Sprint Planning and Setup**							
16	2.1 Sprint 0 (Define product backlog, team onboarding)	A	R	I	I	I	I	I
17	2.2 Sprint Planning	R	A	I	I	I	I	I
19	**3.0 Design and Prototyping Phase**							
20	3.1 Develop Wireframes		R			C		
21	3.2 Create Interactive Prototypes		A			R		
22	3.3 Refine Design Based on Feedback		A	C	C	R		
24	**4.0 Backend Development Phase**							
25	4.1 AWS Cloud Integration				R			R
26	4.2 Implement User Authentication			R	R			
27	4.3 Develop Core Features		I	R	R	C		
28	4.4 Integrate with External APIs			C	C			R
29	4.5 Stakeholder Technical Review	I	R,A	I	I	I	I	I
30	**5.0 Frontend Development Phase**							
31	5.1 Develop the User Interface		C	R	R	R		
32	5.2 Integrate Third-Party Services			R	R			C
33	5.3 Implement Push Notifications		A	I	R	R		I
34	5.4 Stakeholder Demo	I	R,A	C	C	C		
35	**6.0 API Integration and Testing Phase**							
36	6.1 Connect Frontend and Backend Functionality			R,A	R,A			
37	6.2 Conduct Unit Testing			A	A		R	
38	6.3 Conduct Integration Testing						R	A
39	6.4 Test Third-Party Integrations			C	C		R,A	C
41	**7.0 Quality Assurance and Beta Testing Phase**							
42	7.1 Conduct Thorough Testing			I	I		R,A	I
43	7.2 Conduct Beta Testing						R,A	
44	7.3 Incorporate Beta Testing Feedback			I	I		R,A	I
45	**8.0 Deployment and Launch Phase**							
46	8.1 Deploy App to App Stores		R					
47	8.2 Monitor User Interactions		R					
48	8.3 Provide Post-Launch Support	R	R					
49	8.4 Stakeholder Post-Launch Evaluation	R	R					

Appendix D: Budget Plan

Phase	Task	Duration (days)	Hours per Task	Number of People	Wages/hr	Labor Cost ($)
Planning	Stakeholder Communication Plan	7	56	1	60	3360
Planning	Feedback Collection	4	32	1	60	1920
Planning	Audience Analysis	7	56	1	60	3360
Planning	SMART Goals	3	24	1	60	1440
Planning	Core Features Identification	3	24	2	60	2880
Planning	Functional Specs	1	8	2	60	960
Planning	Non-Functional Specs	1	8	1	60	480
Planning	Software Procurement	0	0	0	0	17600
Planning	Misc Costs (Planning)	1	8	1	60	480
					Total Phase Cost	32480
Design Phase						
Design	Sketch Layouts	2	16	1	55	880
Design	Initial Drafts	3	24	1	60	1440
Design	Wireframe Refinement	3	24	1	55	1320
Design	Usability Testing	4	32	1	50	1600
Design	Gather Feedback	2	16	1	60	960
Design	Visual Adjustments	7	56	1	55	3080
Design	Final Design	3	24	2	60	2880
Design	Misc Costs (Design)	1	8	1	60	480
					Total Phase Cost	12640
Backend Development Phase						
Project Management	Sprint Planning	3	24	1	60	1440
Backend	Data Security Setup	5	40	2	65	5200
Backend	Database Integration	5	40	1	65	2600
Backend	Backend APIs	8	64	2	65	8320
Backend	Server Configuration	5	40	1	65	2600
Backend	Misc Costs (Backend)	2	16	1	65	1040
					Total Phase Cost	21200
Frontend Development Phase						
Frontend	Component Consistency	4	32	1	65	2080
Frontend	User Testing	6	48	1	50	2400
Frontend	Third-Party Service Integration	3	24	1	65	1560
Frontend	UI Adjustments	5	40	2	60	4800
Frontend	Accessibility Testing	4	32	1	65	2080
Frontend	Misc Costs (Frontend)	2	16	1	65	1040
					Total Phase Cost	13960
Quality Assurance and Beta Testing Phase						
Quality Assurance	System Testing	10	80	2	55	8800
Quality Assurance	Integration Testing	7	56	2	55	6160
Quality Assurance	User Acceptance Testing	8	64	1	50	3200
					Total Phase Cost	18160
Deployment and Launch Phase						
Deployment and Launch	Prepare App Store Listings	3	24	1	60	1440
Deployment and Launch	Submit App for Approval	4	32	1	60	1920
Deployment and Launch	Technical Documentation	2	16	1	60	960
Deployment and Launch	Collect Usage Data	3	24	1	60	1440
Deployment and Launch	Analyze User Engagement	5	40	2	60	4800
					Total Phase Cost	10560
Ongoing Maintainance and Support Phase						
Ongoing Maintainance and Support	Offer in-App Support	4	32	1	60	1920
Ongoing Maintainance and Support	Maintain Responsive Helpdesk	5	40	2	65	5200
Ongoing Maintainance and Support	Schedule Feature Updates	3	24	5	60	7200
Ongoing Maintainance and Support	Launch Patches	2	16	2	65	2080
Ongoing Maintainance and Support	Conduct User Surveys	2	16	1	60	960
Ongoing Maintainance and Support	Implement Feedback	2	16	1	60	960
					Total Phase Cost	18320
					Total Labor Cost	**109720 $**
					Software Subscription Cost	**17600 $**
					Contingency Cost	**8912 $**
					Total Projected Cost	**136232 $**