Test Plan Report for Group G

Group H:

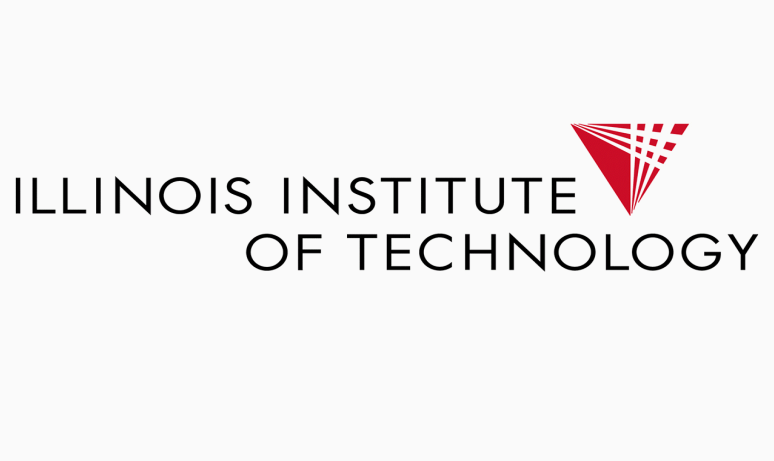
Aditya Rana (A20554487)

Avinash Mantha (A20570450)

Harshil Shah (A20553109)

Ruthwik Dhaipulle (A20548196)

Satyam Rajput (A20537375)



(CS-487)

Software Engineering 1

02/24/2024

Test Plan Report

Test Cases for the key features of Student Assist Application

Goal Setting:

Case 1: Success -
Objective: The User sets specific, measurable, achievable, relevant, time-bound, and academic goals.
Initial State: The user views the goals page/dashboard.
Execution:
– The user clicks on the add goal button. They are then prompted to a goal form. The form entails fields such as goal title, goal description, due date, due time, goal categories, etc. The user enters the title for the goal, they then enter the date by when they intend to achieve the goal (date and time). They then put the goal under categories to decide its relevance.

Expected Final State:
The user successfully enters their goals, which ideally helps them achieve said goals on time.

Case 2: Fail -
Objective: The user successfully sets a specific, measurable time-bound goal.
Initial State: The user views the goals page/dashboard.
Execution:

a) The user clicks on the add goal button, which then prompts them to the “add goal” form. They then attempt to enter the relevant information. However, they forget to add a certain required field (e.g. due date/time, goal title, or fail to categorize the goal).
b) The user clicks on the add goal button, which then prompts them to the “add goal” form. They then attempt to enter the relevant information. However, they accidentally enter incorrect information (e.g. the wrong due date/time, title, or put the goal under the wrong category).

Expected Final State:

a) The user is unable to successfully add a goal. They are prompted with an error message to enter the required information.

b) The user successfully adds the goal, however, due to adding incorrect information, they are unable to achieve the goal on time, or don't achieve the goal accurately.

Personalized Recommendations :

Case 1: **Personalized Academic Resources Success**
Objective: Users receive personalized academic resources based on their preferences and learning styles to assist them in achieving their goals.
Initial State: The user has set up their profile with goals, preferences, and learning styles.
Execution: The user navigates to the recommendations section. The system processes the user's profile data and displays a list of tailored academic resources.
Expected Final State: The user views a list of recommended resources that align with their academic goals and preferences.

Case 1.1: Updating Preferences Success
Objective: Users update their preferences and receive updated recommendations accordingly.
Execution: User modifies their learning style and interests in their profile settings. They then navigate to the recommendations section.
Expected Final State: Recommendations are updated to reflect the user's new preferences, showcasing different resources than before the update.

Case 2: Failed Configured Preferences
Objective: The user tries to get suggestions without configuring any preferences.
Execution: The user navigates to the recommendations section. The system attempts to generate recommendations without sufficient data.
Expected Final State: Before getting tailored suggestions, the user is asked to finish their profile.

Case 2.1: No Recommendations Available Fail
Objective: The user has highly specific or uncommon preferences leading to no available recommendations.
Execution: After setting unique or narrow preferences, the user navigates to the recommendations section.
Expected Final State: The app notifies the user that no recommendations are currently available for their specific preferences and suggests broadening their interests.

Case 2.2: Incorrect Recommendations Fail
Objective: The user has ambiguous preferences that lead to incorrect recommendations that don't align with their goals, learning styles, or preferences.
Execution: While setting up their profile, the user includes preferences that are ambiguous or simply incorrect. They then navigate to the recommendations page.
Expected Final State: The recommendations shown are incorrect and do not align with the user's goals, learning styles, or preferences.

Progress Tracking :

Case 1: Milestone Achievement Success
Objective: The user accomplishes a part of an overall goal.
Initial State: The user has set a goal with several milestones and is actively using the programme.
Execution Steps:

1. Task Completion:
    - The user accomplishes a task that is directly related to reaching a goal milestone.
    - Completing an assignment, completing a task, or tracking the amount of time spent on the goal could all be considered completion actions.
2. Progress Update:
    - The user proceeds to the application's tracking of progress section after marking the task as finished.
    - The user's progress towards the overall goal is recalculated by the system after processing the completion action.

3. Milestone Acknowledgment:
    - The programme evaluates the user's progress and recognises the achievement of an objective.
    - To highlight the milestone accomplishment, the system may send out a notification or show a unique acknowledgement in the progress tracking section.

4. Progress Illustration:
    - An updated progress visual is shown to the user, highlighting the recently reached milestone and how it affects the overall goal advancement.
    - A checklist of milestones, progress bars, pie charts, or other visualizations could be used to show that the most recent milestone has been reached.

Expected Final State:
The application refreshes the total progress towards the main goal and effectively shows the milestone's achievement.
The user receives positive reinforcement through visual acknowledgement and possibly motivational messages, enhancing their motivation to continue working towards the remaining milestones and the ultimate goal.

Case 2: Warning of Stagnant Progress Failed

Objective: After an extended time of inactivity, the application does not alert the user about the stagnant progress.
Initial State: The user has set goals within the application but has not made any progress towards these goals in a considerable amount of time.
Execution Steps:

1. Inactivity Period:
    - The user does not engage with the tasks and activities for over a significant period of time, indicating a lack of progress.
2. Accessing progress tracker:
    - The user chooses to check their progress within the programme after this period of inactivity.
    - To see where they stand right now about certain goals, the user navigates to the progress monitoring section.
3. System Response:
    - The ideal case is that the system should analyze the user's inactivity and lack of progress, triggering a warning or notification about the stagnant progress.
    - In this case of a failure, the system either doesn't recognise the lack of progress or doesn't produce and show the relevant alert and recommendations for getting back on track.

Expected Final State:
The application doesn't notify the user that their progress remains stagnant even though there has been a lot of inactivity.
The possibility to re-engage the user with their goals is lost when there is no warning or motivational suggestion, which may result in long-term inactivity and a worse user experience.
The user doesn't know why they are stagnating and doesn't receive important feedback that could direct them to refocus their efforts on achieving their objectives.

Time Management:

Case 1: Success -
Objective: The User sets frequent reminders and notifications to set a study schedule based on the day's activity.
Initial State: The user views the course page.
Execution:
– The user clicks on the add reminder button.
– The user sets the timing and duration of the study based on availability and then click on save.
Expected Final State:
The user successfully gets prompted via in-app notifications to study at a given time every day for a specific duration to maintain consistency.

Case 1.1: Schedule Conflict Success
Objective: The user sets a reminder to study but gets a schedule conflict warning.
Initial State: The user views the course page.
Execution:
– The user opens up the reminder window and sets the time to study.
– The user tries to save the schedule but gets a warning of a possible conflict of time.

Expected Final State:
The user successfully gets prompted about a possible schedule of conflict and then makes necessary adjustments to clear conflicts and set new reminders.

Case 2: Fail -
Objective: The User sets regular reminders and notifications to set a study schedule based on the day's activity.
Initial State: The user views the course page.
Execution:
– The user clicks on the add reminder button but fails to save the reminder setting.
– The user sets the timing and duration of the study based on availability but incorrectly sets the timing or not entering correct information while prompting data.

Expected Final State:

a) The user is unable to add reminders. They are prompted with an error message to save the required information.

b) The user successfully adds the reminders, however, due to incorrect information, they are unable to set study schedules or don't achieve the goal accurately.

Performance Analysis

Case 1: Successfully Identify the academic Performance Trend
Objective: The aim is to effectively identify performance trends in the student's academic achievements and insights into their academic strengths and weaknesses.
Initial State: User has completed assessments or tasks within the app.
Execution: After using the app for an extended period, the user reviews their performance analysis.

- Data Accumulation: Verify that the user has been utilizing the app regularly for a considerable amount of time, entering grades, finishing assignments, and getting feedback.
- Examine Performance: The user opens the app's performance analysis section.
- Analysis Presentation: The app analyses collected data to identify patterns and show them, emphasizing problems and opportunities for development.

Expected Final State: The user is provided with clear, practical insights regarding the trends in their academic performance, allowing them to make decisions regarding their study strategies.

Case 1.1: Success: Using plenty of data, the app assures the user by pointing out a distinct upward trend in math performance.

Case 1.2: Failure: The app mistakenly identifies a trend based on outlier performances, leading to misleading insights.

Case 2: Insufficient Data to Conduct Analysis
Objective:

- Handle cases where there is not enough information available for a performance evaluation.
- User requests performance analysis even though no assessments have been finished.

Initial State: User has no completed tasks or assessments in the app.

Execution:

- Early Usage: Shortly after opening the app, the user makes an effort to access the performance analysis with little to no data entry.
- Attempt Analysis: The user accesses the section on performance analysis.
- App Response: After evaluating the amount of data available, the app concludes that it is not enough for a meaningful analysis.
- Alerts provided: The user is advised to record more activities, grades, or feedback after being made aware of the insufficient data.

Expected Final State: The user receives guidance on how to improve their input and is informed that additional data is required to facilitate performance analysis.
Case 2.1: Failure: The app is unable to identify incomplete data, leading to unclear or inefficient insights that confuse the user about their performance.

Community:

Case 1: Success -
Objective: The user can connect with students that have similar goals and interests as the user.
Initial State: The user has set up their profile with their goals, interests, and learning styles (e.g. study groups).
Execution:

- The user navigates to the community tab. The system filters through and displays a list of students that share similar goals and interests with the user.
- The user can then send a follow request and a message asking the student to connect.

Expected Final State:

- If the other student(s) accept, they are able to chat, discuss, and send useful academic resources (that do not violate academic integrity).

Case 1.1 Update Goals, Preferences for Community Tab Success -
Objective - The user updates the goals and preferences on their profile to find students that share their goals and interests.
Execution - the user modifies the goals and preferences listed on their profile. They then navigate to the community tab.
Expected Final State - The user discovers recommendations for students they can connect with that align better with their new goals, preferences, and learning styles.

Case 2: No Configured Preferences Fail -

Objective: The user attempts to find students to connect with, without setting their profile up with their goals, interests, or preferred learning methods.

Execution:

- The user navigates to the community tab to find students to connect with.

Expected Final State:

- The community tab does not show any students and prompts the user with an error message informing them to finish setting up their profile

Case 2.1: No Configured Preferences Fail -

Objective: The user attempts to find students to connect with, without setting their profile up with their goals, interests, or preferred learning methods.

Execution:

- The user navigates to the community tab to find students to connect with.

Expected Final State:

- The community tab does not show any students and prompts the user with an error message informing them to finish setting up their profile

Case 2.2: Incorrect Preferences Fail -

Objective: The user navigates to the community tab to find students to connect with. However they fail to accurately set up their goals and preferences on their profile.

Execution:

- The user configures their profile with ambiguous/obscure information, or information that doesn't align with their goals and preferences.
- They then navigate to the community tab to find students to connect with.

Expected Final State:

- The community tab shows a list of students that have different goals and interests from the user

Suggestions:

User Feedback Loop: Including an approach to collect and incorporating user input into subsequent versions of the programme will help in continuously enhancing and optimizing its features.

Goal Setting:
The app helps users set specific, measurable goals... - slightly ambiguous, how exactly do you intend to make goals measurable? Measurable by time or level of importance? My suggestion would be to categorize goals by level of urgency or by their level of difficulty.

Performance analysis feature:
To make the performance analysis feature work even when there isn't a lot of data available, we can either use tools that predict future trends or offer basic insights even with just a little information.This way, users can get started and find value, even before they have large amounts of data.

User Category Clarification: Clearly identify the different user groups that exist within the application, such as advanced and beginner users. To improve personalisation and effectiveness, customize features and recommendations according to the user's goals and level of proficiency.

Community Tab:
On a study platform, it is essential to ensure that it is moderated in some way to avoid violating academic integrity.
Ways to do this:
- not allowing sending links
- communication is moderated by the professor, etc.