Software Design and Analysis (SE-2002)

Assignment 1(Initiation)

Software: Anki

Group#10

Sana Idrees(23i-2039), Manahil(23i-3000),

Maryam Fatima(23i-3007) and Ayesha Khan(23i-3037)

BS-Software Engineering(4B)

Submitted to: Dr. Behjat Zuhaira

Client: Group# 1

Table of Contents

Brainstormed Solution:

To enhance the learning insights **gamification, collaboration** and **AI-driven insights** will be integrated in the system. The key features are as follow:

- ✓ **Battle mode** for real-time competition
- ✓ **Daily Challenges** tailoring user progress
- ✓ **Streaks & Achievements** for increasing consistency and motivation
- ✓ **AI Buddy** for personalized feedback
- ✓ **Leaderboards** for social learning and **community engagement**

Functional Requirements:

1. Battle mode for Flashcards

1.1 **Real Time Battles:** The system shall allow users to challenge their friends in a battle mode by confronting them in flashcard battle in real time.

1.2 **Game Statistics**: The system shall record the statistics of the game such as the number of cards answered correctly/incorrectly, average time taken to answer each card, winner and loser of the game.

1.3 **Post Game Leadership board**: The system shall display a leadership board after each game, displaying the users based on their winning stats.

1.4 **Badge Rewards:** The system shall reward winner with an increased ranking badge such as a Master of Memory, Brain Brawler, Ultimate quiz King/Queen.

2. Daily Challenges

2.1 **Personalized Challenges:** The system shall generate flashcards challenges on daily basis based on their last studied subject on the application.

2.2 **Progress Syncing:** The system shall keep track of the user's progress and synchronize it with the user's credentials.

2.3 **Streak Points**: The system shall rewards users with streak points for keeping a better track of user's progress.

3. Streaks and Achievements

3.1 **Streaks Tracking**: The system shall keep track of the user's consecutive study days and consequently assign streak tags.

3.2 **Streak Break or Reset:** The system should break the streak if user is inactive for more than 24 hours' time period.

3.3 **Notifications:** The system shall tag the streak on user's profile, based on the time duration passed such as if the streak was for 7 days, then Consistency Champ should be tagged and if the streak was for 21 days then Unstoppable Learner should be tagged.

3.4 **Notifications for warning when streak is about to end:** The system shall notify the user if the streak is about to end such as "Oh no! Champ you're about to lose your streak of 6-days".

4. AI Study Buddy

4.1 **Weakness Analysis**: The system shall give analysis of the weak study areas after each study session.

4.2 **Motivational Feedback**: The system shall use the AI tool to give motivational and informative side notes to the user.

4.3 **Recap Session:** The system shall allow the user to interact with the AI tool to do a quick recap of their study session.

5. Leaderboard and Community Engagement

5.1 **Group Leaderboard:** The system shall display a leaderboard ranking based on the user's streaks, badges and wins on the community page.

5.2 **Sharing Progress**: The system shall allow the user's friend circle to see their progress.

5.3 **Sharing Stats in Community Wall**: The system shall allow the users to share their statistics and achievements in community wall.

5.4 **Group Features**:

- ✓ The system should allow shared flashcard among the group
- ✓ The system should allow collaborative challenges (e.g., "Group Quiz Night")

Identifying Actors

Primary Actors of the System:

1. Learner

Learner is any individual who interacts with the system for **personal study** and learning sessions.

- **"Why student is a primary member?"**

✓ The main user of the system

✓ Directly interacts with all the study tools and system features.

- **Role and Responsibilities:**

✓ Creates and studies flashcards.

✓ Engages with gamification features to enhance learning process.

✓ Uses AI powered recommendations for better understanding.

✓ Tracks personal progress and leaderboard position.

✓ An active participant of challenges and quizzes.

✓ Streak and badge holder based on their consistency and progress.

Goals:

✓ They can master the subjects and topics by reviewing their flashcards.
✓ They can **record and display the statistics** like correct or incorrect answers, winner details etc
✓ The learner **maintains study streaks**
✓ They **feel engaged** while using the application.
✓ The learners **earn** different **ranking badges** based on their study consistency and victories.

- ✓ They **receive the feedback** on the basis of their interactions with the application which helps them to stay motivated like **earning achievements titles** as “consistency Champ” or “unstoppable learner” for maintaining streaks.
- ✓ They can **track their progress** in reviewing notes and learning overtime.
- ✓ They can **customize the application** according to their preferences.
- ✓ They can **optimize the amount of time** they spend on making and reviewing flashcards.
- ✓ The learners are provided with **daily flashcards challenges** about the last studied subjects.
- ✓ They interact with the AI tool for a **quick recap** of their study sessions.
- ✓ The learner receives **AI recommendations** on how to improve their study performance.
- ✓ They can **enhance memory retention** through benefit from spaced repetition techniques.
- ✓ They **monitor progress on leaderboard**

2. Group Study Member

This represents a **student who works in a collaborative environment**.

- **“Why group study member is a separate primary member?”**

✓ Not all students study in groups and vice versa.

✓ Group members need extra system interactions such as chat, shared decks, group challenges.

- **Roles and Responsibilities:**

✓ Joins or creates study groups

✓ Competes in group challenges

✓ Views and compares progress with their friends on leaderboard.

✓ Engages in community discussions.

✓ Shares flashcard/decks with group peers in real time.

l Goals:

- ✓ They can share the decks in real time for collaborative learning.
- ✓ They can **monitor** each other **progress on the leaderboard**.
- ✓ Can view each **group member with ranking badges** based on their team performance.
- ✓ They may create a collaborative blog/vlog documenting their learning process.
- ✓ They can **create group learning strategies** for exam preparation.
- ✓ Engage in **community chats** with members where they can discuss about their study plans.
- ✓ Create and **share in real-time the quizzes**, study challenges with the group members.

- ✓ They can **create comprehensive study summaries** for the entire group.
- ✓ they can **create a collaborative study calendar** by scheduling sessions, deadlines, and share feedback.
- ✓ They **can create study workbooks or templates** for study use
- ✓ They can **share statistics with peers**

Actors Goals and Use Cases

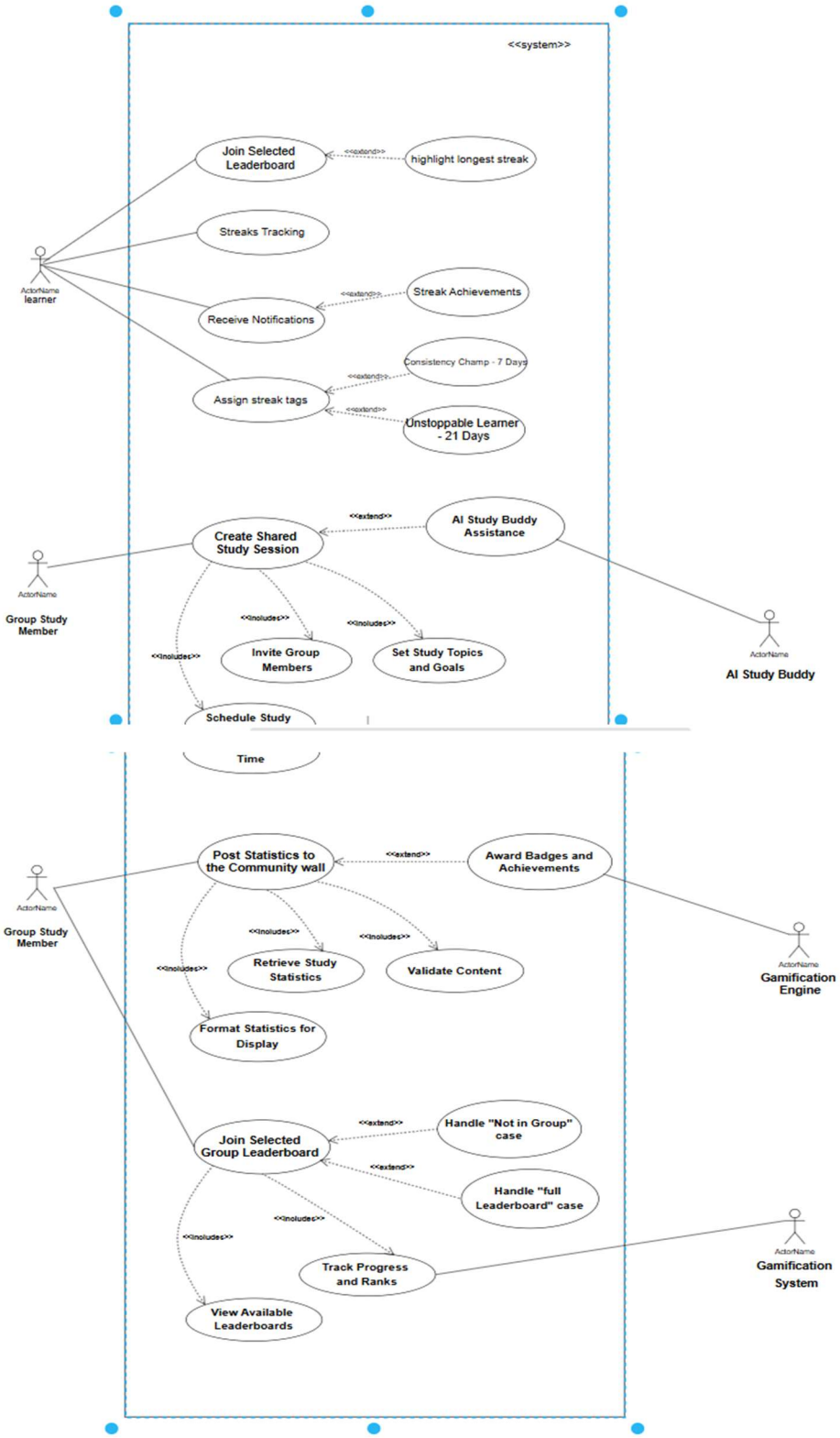
Actor	Goal	User Cases	Associated FRs
Learner	Track study consistently	View Streak Progress	3.1,3.3
	Engage in challenges	Complete daily challenge	2.1,2.3
	Compete with friends	Initiate Flashcards Battle	1.1,1.2,1.4
	Optimize Study Plans	Review AI driven recommendations	4.1,4.2,4.3
	Maintain consistency and motivation	Receive Streak warning notifications	3.2
Group Study Member	Collaborate on shared deck	Create shared study sessions	5.4
	Compete in group challenges	Join Group Leaderboard	5.4
	Compare progress with peers	View Group Statistics	5.1,5.3
	Share achievement	Post Statistics to Community Wall	5.2,5.3

Use Case Testing(New FR's):

Use Case	Boss Test	EBP Test	Size test	Comments
View Streak Progress	F	P	F	• Boss test passed because this use case aligns with the goal of tracking study consistency and this directly supports user engagement and motivation. • EBP test passed because it helps to track the study performance which is a measure able feature, progress tracking is visible through UI which is an observable feature then it involves retrieving and displaying the streaks data which is a single task.
Complete daily challenge	P	P	P	• EBP test approved as the system tracks challenge completion, making it observable and valuable, and is a

				self-contained task where a user selects, attempts, and completes a challenge in one flow without requiring multiple independent processes.
Initiate Flashcards Battle	P	P	P	User reports winning or either losing a battle
Review AI driven recommendations	P	P	P	Actionable Feedback Session
Receive Streak warning notifications	F	F	F	It is System triggered task rather that user initiated
Create shared study sessions	P	P	P	Collaborative activity between group
Join Group Leaderboard	P	P	P	Competitive engagement
View Group Statistics	F	F	F	Just passive viewing
Post Statistics to Community Wall	P	P	P	Active sharing of achievements
Key: Pass=P, Fail=F				

Use cases Diagram (New FR's):


<<system>>
Join Selected Leaderboard
<<extend>>
highlight longest streak
Streaks Tracking
ActorName
learner
Streak Achievements
Receive Notifications
Consistency Champ - 7 Days
Assign streak tags
Unstoppable Learner - 21 Days
AI Study Buddy Assistance
Create Shared Study Session
ActorName
Group Study Member
<<Inoludes>>
Invite Group Members
Set Study Topics and Goals
Schedule Study Time
ActorName
AI Study Buddy
Post Statistics to the Community wall
Award Badges and Achievements
ActorName
Group Study Member
Retrieve Study Statistics
Validate Content
Format Statistics for Display
ActorName
Gamification Engine
Handle "Not in Group" case
Join Selected Group Leaderboard
Handle "full Leaderboard" case
Track Progress and Ranks
View Available Leaderboards
ActorName
Gamification System

Fully Dressed Format of Use cases:

1. Title: AI-Driven Recommendations

Scope:

Ankidroid Enhanced Learning System

Level:

User Goal

Primary Actor:

Learner/Student

Stakeholders and Interests:

Student: Wants personalized study recommendations to improve learning efficiency.

System Administrator: Ensures AI recommendations function correctly and securely.

Institution (if applicable): May want insights on student learning trends (if consent is given).

Preconditions:

The student must have an active account and be logged in.

The system must have sufficient historical study data for personalized recommendations.

The AI recommendation engine must be functional and accessible.

Success Guarantee (Postconditions):

The student receives personalized study recommendations based on learning history and preferences.

Recommendations are accurate, relevant, and actionable (e.g., suggested flashcards, study sessions, group study partners).

Main Success Scenario:

The student accesses the AI Recommendations section.

The system analyses past study behaviour, progress, and weak areas.

AI generates a list of recommended study materials, topics, or study sessions.

The student reviews the recommendations.

The student selects a recommendation (e.g., a suggested flashcard deck or study session).

The system provides access to the chosen content.

The system updates learning progress based on the student's engagement.

Extensions (Alternate Success/Failure Scenarios):

The system lacks enough data to generate personalized recommendations.

The system prompts the user to complete a learning preferences survey to improve recommendations.

The system offers general trending decks as an alternative.

The AI recommendations are irrelevant or not useful to the student.

The student provides feedback (e.g., "not relevant"), and AI adjusts future suggestions accordingly.

The system allows students to refine recommendations by selecting specific topics of interest.

The student does not engage with the recommendations.

The system sends a follow-up notification after a set period, reminding the user to check recommendations.

The student encounters technical issues (e.g., recommendations fail to load).

The system displays an error message and suggests trying again later.

The system logs the error for administrators to investigate.

Special Requirements (Non-Functional Requirements):

Performance: Recommendations should be generated in under 2 seconds to ensure smooth user experience.

Scalability: The AI system should handle thousands of concurrent users without performance degradation.

Accuracy: Recommendations should be at least 80% relevant based on prior user interactions.

Security: The system must protect student data privacy and comply with GDPR and FERPA.

Technology & Data Variations List:

AI recommendations may be generated using Machine Learning (ML) models trained on student interaction data.

The system may use Natural Language Processing (NLP) to suggest topics based on notes and previous flashcard usage.

Recommendations could be displayed in different UI formats, such as a list view, visual graph, or chatbot interface.

Frequency of Occurrence:

Investigation Influence: The frequency of AI usage will be monitored to determine effectiveness and improvements.

Testing Considerations:

AI recommendation accuracy should be tested with at least 1000 study sessions before deployment.

Load testing must ensure AI handles high traffic scenarios efficiently.

Implementation Timing:

The AI recommendation feature can be introduced in phases, starting with basic recommendations before expanding to adaptive learning models.

Continuous updates may be released based on feedback loops from student interactions.

Miscellaneous (Open Issues):

Bias in AI Recommendations:

How to ensure that AI does not favor certain topics or learning styles disproportionately?

How to prevent AI from reinforcing ineffective study habits?

Privacy Concerns:

How will student data be handled securely while maintaining recommendation accuracy?

Should AI recommendations be opt-in by default?

Recommendation Feedback Loop:

How will the system continuously improve its recommendations based on user feedback?

Should there be an explainability feature to show why a recommendation was made?

Edge Cases Handling:

How should the system behave for new users with no study history?

How will AI adapt when a student drastically changes their study habits?

2. Title: Create Shared Study Sessions

Scope:

Anki-based collaborative learning software

Level:

User goal

Primary Actor:

Student/Learner (Initiating the shared study session)

Secondary Actor(s):

- Group Study Member(s) (Participants joining the session)
- AI Study Buddy (If enabled, provides insights, quizzes, and suggestions)

Stakeholders and Interests:

Students: Want to collaborate and study effectively in real-time.

Group Members: Need seamless access to shared resources.

System Admin: Ensures session security and server efficiency.

Preconditions:

The student must be logged into the system.

The student must have at least one deck of flashcards available.

Internet connectivity is required for real-time collaboration.

Success Guarantee (Postconditions):

A shared study session is successfully created.

Invited participants can access and interact with shared study materials.

Main Success Scenario:

- The student selects the "Create Shared Study Session" option.
- The system prompts the student to select a deck for the session.
- The student configures session settings (e.g., duration, permissions, interactive mode).
- The system generates a unique session link/code.
- The student invites group members via the generated link or by selecting contacts.
- The invited members join the session.
- The system synchronizes flashcards and study progress in real-time.
- Participants engage in collaborative learning (e.g., quizzes, discussions).
- The session ends, and a summary of progress is displayed.
- Extensions (Alternative Scenarios):

4a. Student cancels session creation → System discards the session.

5a. A user attempts to join an expired session → System displays an error.

6a. Group members decline or do not join → Session proceeds with available participants.

7a. Connection issues occur → System tries reconnection or saves progress locally.

8a. AI Study Buddy provides additional explanations or questions if enabled.

Special Requirements (Non-Functional Requirements):

The system should support a minimum of 5 and a maximum of 50 participants per session.

The session must update in real-time with less than 1-second latency.

Data security must be ensured for shared study materials.

The system should work across multiple devices (mobile, desktop, tablet).

Technology & Data Variations List:

Cloud-based synchronization for real-time collaboration.

AI integration for personalized study suggestions.

Support for multiple file formats (PDFs, images, text, etc.).

Frequency of Occurrence:

Expected to be frequent, especially before exams and group projects.

May influence system load balancing and server capacity planning.

Miscellaneous (Open Issues):

Should sessions be recorded for later review?

How should AI moderation be handled if users misuse the session?

What data should be stored after a session ends (e.g., chat logs, study progress)?

3.Use Case Name: Initiate Flash Card Battle

Scope:

Anki-based collaborative learning software

Level:

User goal

Primary Actor:

Student/Learner (User who initiates the flash card battle)

Secondary Actor(s):

Opponent (Other Student/Group Member) (The participant who joins the battle)

AI Study Buddy (If enabled, provides AI-generated questions or acts as an opponent)

Stakeholders and Interests:

Students: Want an engaging and competitive way to test their knowledge.

Group Members: Need real-time interaction and fair scoring mechanisms.

System Admin: Ensures that the battle system runs smoothly and fairly.

Preconditions:

The student must be logged into the system.

The student must have at least one deck of flashcards available.

Internet connectivity is required for real-time competition.

At least one available opponent (human or AI) is needed.

Success Guarantee (Postconditions):

A flash card battle is successfully initiated.

Opponents receive and answer flash card questions in real-time.

A winner is declared based on accuracy and response time.

Main Success Scenario:

The student selects the "Initiate Flash Card Battle" option.

The system prompts the student to choose a deck for the battle.

The student selects an opponent (from friends, random players, or AI).

The opponent accepts the battle invitation.

The system starts the battle, presenting flashcards alternately to both players.

Players answer within a time limit, and scores are assigned based on correctness and speed.

The battle continues for a predefined number of rounds or until a set score is reached.

The system announces the winner and updates rankings/leaderboards.

Players can review mistakes and request a rematch.

Extensions (Alternative Scenarios):

4a. The selected opponent declines the battle → System allows the student to choose another opponent or cancel.

4b. No available opponent → System offers AI as an opponent.

5a. A player disconnects mid-battle → System either waits for reconnection or declares a forfeit.

6a. A player doesn't respond within the time limit → System marks the answer as incorrect.

8a. AI Study Buddy provides learning insights after the battle (if enabled).

Special Requirements (Non-Functional Requirements):

The battle must operate in real-time with latency below 500ms.

A fair scoring system should be implemented.

The system should prevent cheating (e.g., restricting external help).

User data security must be ensured.

Technology & Data Variations List:

Cloud-based real-time synchronization for multiplayer battles.

AI-generated adaptive questions for AI opponents.

Customizable battle settings (difficulty level, timer options).

Frequency of Occurrence:

Expected to be moderate to high, especially during competitive study sessions.

May impact server load balancing and matchmaking algorithms.

Miscellaneous (Open Issues):

Should battle stats be recorded permanently or only for ranking updates?

How should tie-breakers be handled?

Should there be a penalty system for repeated disconnections?

4.Use Case Name: Compete Daily Challenge

Scope:

Gamified learning platform

Level:

User goal

Primary Actor:

Student/Learner (The user who competes in the daily challenge)

Secondary Actor(s):

Gamification Engine (Manages the setup of the challenge, tracks progress, and applies rewards)

AI Study Buddy (Provides assistance if configured to help with hints or explanations)

Stakeholders and Interests:

Students: Desire to complete daily challenges for rewards, motivation, and learning reinforcement.

System Admin: Wants to ensure the fairness of challenges and monitor engagement.

Teachers (if applicable): Interested in ensuring students engage with daily content to enhance their learning.

Preconditions:

The student must be logged into the system.

The system must present an active daily challenge for the student.

The student must not have completed the challenge already for the day.

Success Guarantee (Postconditions):

The student has competed in the challenge and received feedback or rewards.

The student's leaderboard rank or progress has been updated.

Main Success Scenario:

The student accesses the Daily Challenge section of the platform.

The system presents the daily challenge based on the student's current level or history.

The student reads through the instructions or challenge details.

The student competes by answering questions or solving problems presented in the challenge.

The system evaluates the student's responses and provides immediate feedback.

If the student completes the challenge successfully, the system grants them rewards, such as XP, points, badges, or leaderboard placement.

The system updates the student’s progress and leaderboards.

The student may review feedback or see recommendations to improve further.

Extensions (Alternative Scenarios):

3a. The student cannot access the challenge due to connectivity issues → The system saves the progress and allows the student to resume later.

4a. The student fails to submit answers or abandons the challenge → No progress is recorded, and the challenge can be attempted again later.

5a. The AI Study Buddy assists the student → Offers hints or helpful suggestions during the challenge.

6a. The student does not complete the challenge within the time limit → The system records incomplete status and may allow for a retry on a future date.

7a. The student earns rewards (e.g., badges, points, or XP) → The system notifies the student of their achievements, and the leaderboard is updated.

Special Requirements (Non-Functional Requirements):

Real-time feedback should be provided to the student for immediate learning.

The challenge platform must handle high concurrency for simultaneous access by many students.

The system must ensure that reward distribution is fair and motivating.

The challenge difficulty should adjust based on the student's previous performance.

Technology & Data Variations List:

Support for multiple types of challenges, including quizzes, games, and interactive problem-solving activities.

Integration with AI-driven systems for personalized difficulty and real-time hints.

Offline mode for students to download and compete with challenges, with later sync.

Frequency of Occurrence:

Daily, as the system presents a new challenge each day.

Expected high frequency of engagement, driving daily logins and student retention.

Miscellaneous (Open Issues):

How should challenges scale in difficulty for students with varying levels?

Should challenges be optional for students, or mandatory for consistent progression?

- What metrics should be used to track challenge participation (time spent, attempts, etc.)?

Casual Format of Use Cases

1.Use Case Name**:** Complete Daily challenges

Main Success Scenario:

The system generates a daily flashcard challenge based on the user's last studied subject.

The user completes the challenge within the given time.

Streak points are awarded for consistency.

Progress is synced with the user's profile.

Alternate Scenarios:

User skips the challenge: No streak points are awarded, and streak progress may reset.

System fails to generate a challenge: A default challenge is provided instead.

User completes the challenge late: Streak points are awarded based on app rules

2. Use Case Name: Initiate Flashcard Battle

Main Success Scenario:

The user selects the battle mode and invites a friend.

The friend accepts the challenge, and the game begins in real time.

Each player answers flashcard questions within a time limit.

The system records the number of correct/incorrect answers and time taken.

The game ends, and the winner is announced.

A leaderboard displays ranking based on past wins.

The winner is awarded a ranking badge.

Alternate Scenarios:

Friend does not accept the challenge: The user can invite another friend or play a solo challenge.

Connection issues occur during the game: The system pauses or ends the match based on reconnection rules.

Both players tie: A tie-breaker round is initiated

3. Use Case Name: Review AI-driven recommendations

Main Success Scenario:

The user completes a study session.

The system analyses performance and identifies weak areas.

AI provides personalized study recommendations and motivational feedback.

The user can interact with AI for a quick recap session.

Alternate Scenarios:

User skips the AI recommendations: The system saves the analysis for future reference.

AI fails to generate recommendations: A default study tip is provided instead.

User requests an extra recap session: The AI provides additional flashcards or summary points.

4.Use Case Name: Create Shared Study Session

Main Success Scenario:

A group member creates a shared study session and selects a topic.

The system notifies all group members about the session.

Members join the session in real time.

The session begins with shared flashcards and interactive quizzes.

Members answer questions, discuss topics, and track progress together.

The system records participation and updates group statistics.

Alternate Scenarios:

No members join the session: The system cancels the session or allows the creator to study solo.

A member joins late: They enter the session with progress synced.

Internet connection issues: The system attempts reconnection or allows offline mode with delayed

Brief format:

Complete Daily Challenge

Anki helps users build a strong study habit by encouraging them to complete a set number of flashcards each day. This feature uses spaced repetition, meaning it shows you difficult cards more often while spacing out easier ones. This helps with long-term memory retention and makes learning more effective over time.

Initiate Flashcard Battle

This feature lets users compete with friends or AI in a fun, quiz-style game using flashcards. Players take turns answering questions, and the system keeps track of scores. This makes studying more interactive and engaging, turning learning into an exciting challenge rather than just a solo task.

Review AI-Driven Recommendations

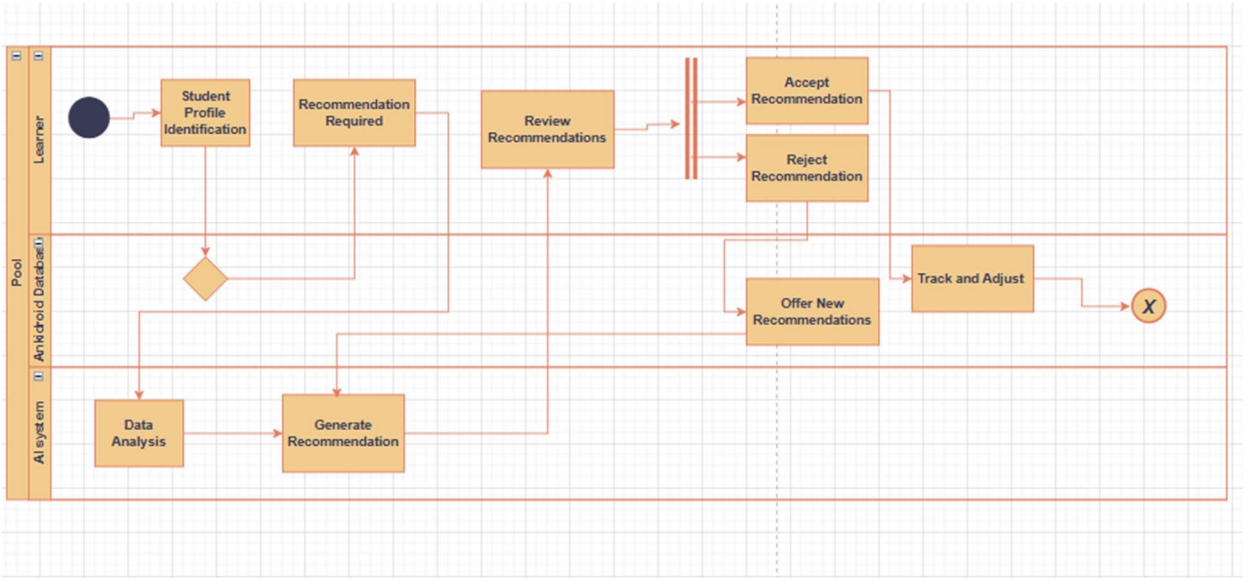
Anki's smart AI analyses your learning patterns and suggests the flashcards you need to focus on the most. If you struggle with certain topics, the AI will bring them up more often, while showing you less of what you already know. This helps optimize study time and ensures you are always improving.

Create Shared Study Sessions

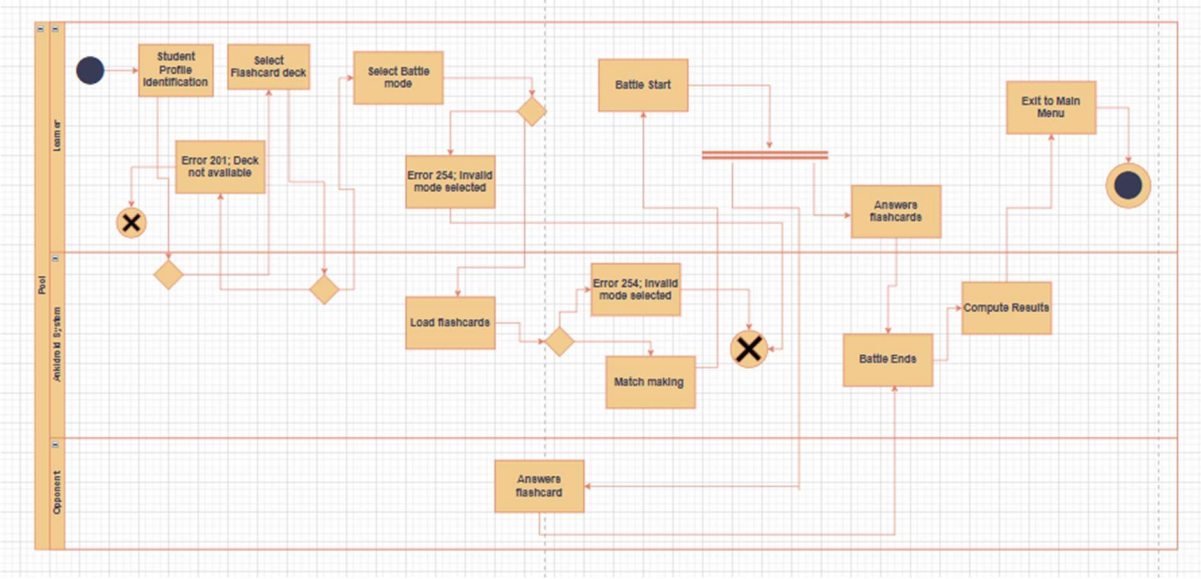
Users can join real-time study sessions with friends, allowing them to review flashcards together. This feature makes learning more social and effective, as discussing answers and explaining concepts to others can help reinforce understanding. It is especially useful for group study, test preparation, or class reviews.

Activity Diagram:

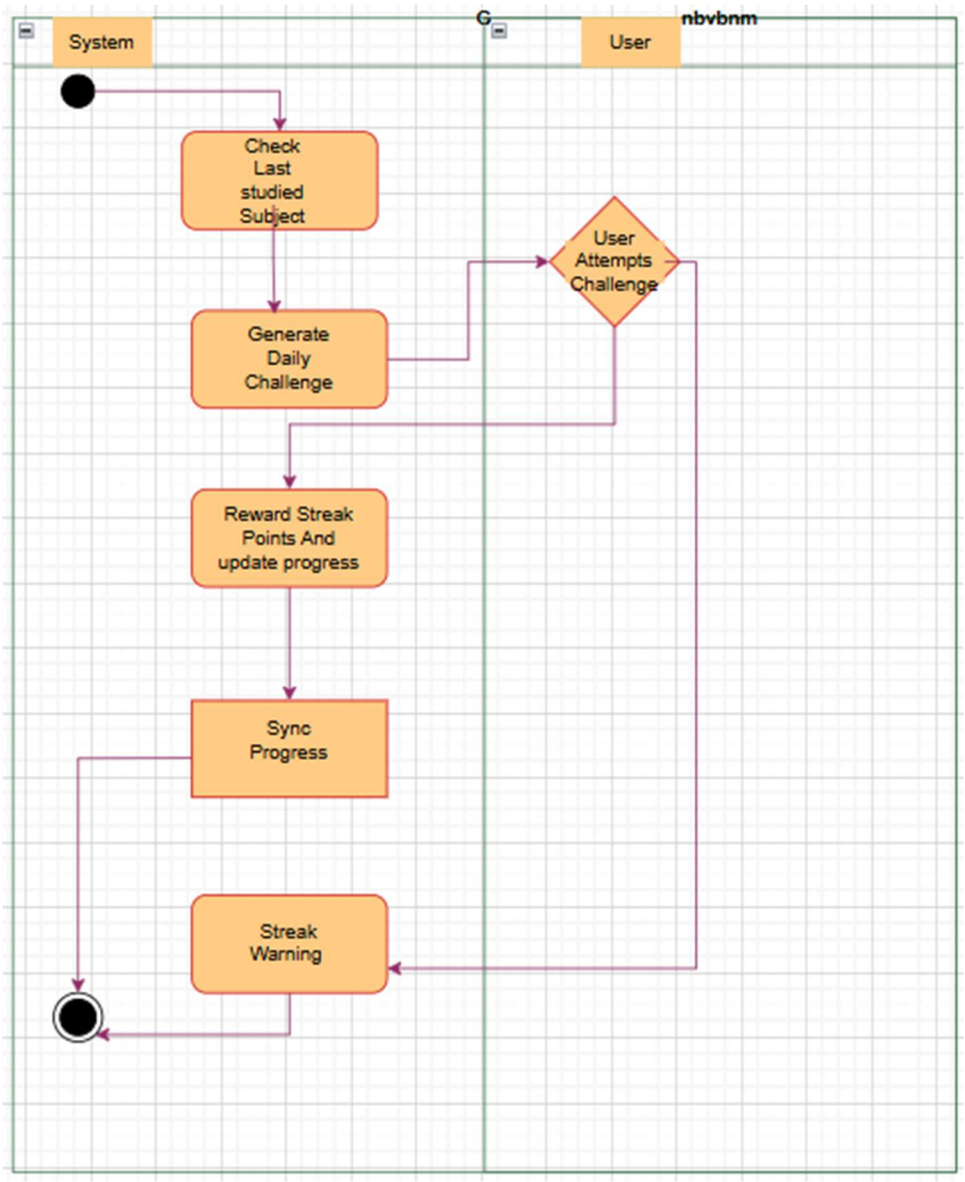
1. AI driven recommendations



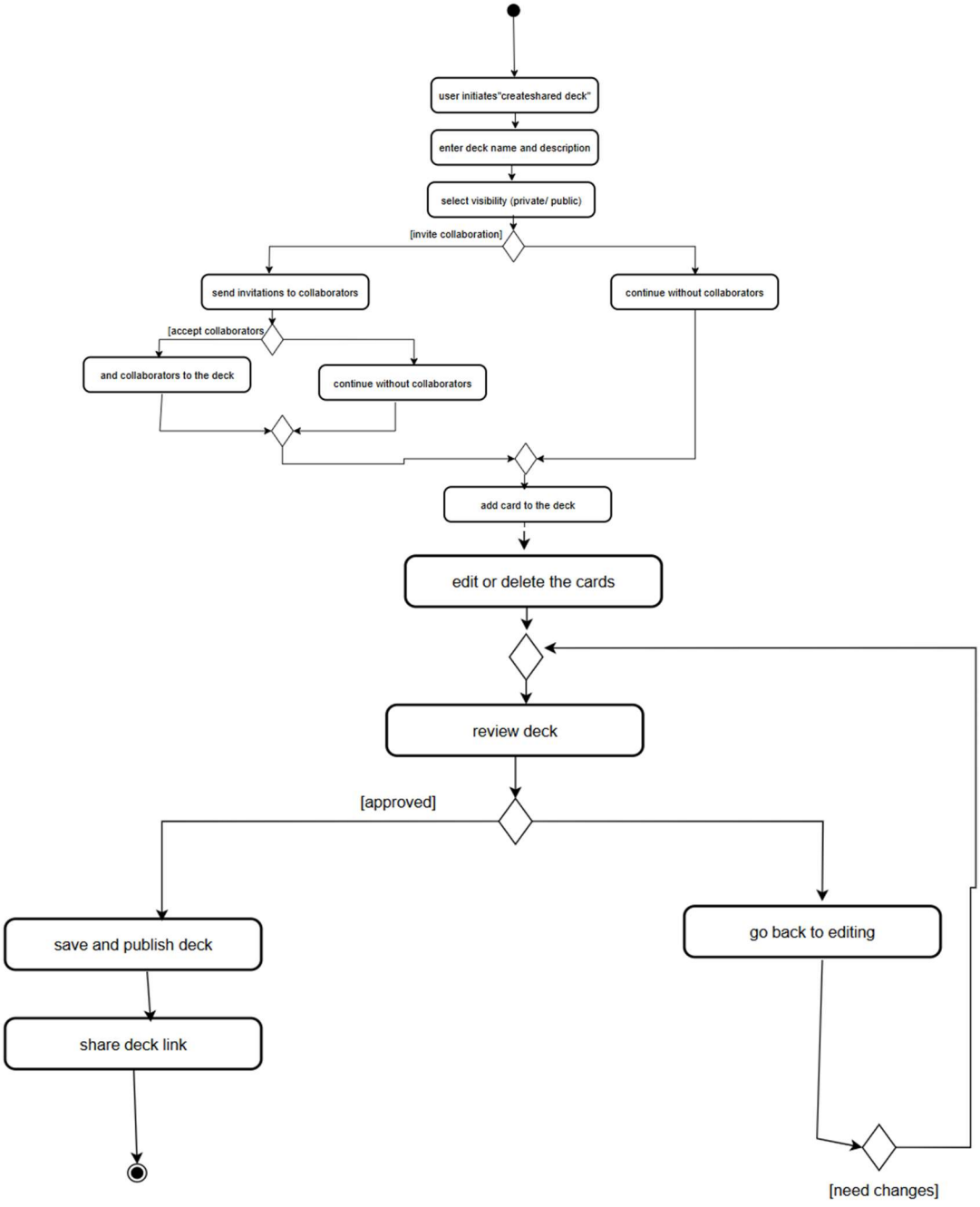
2. Initiate Flashcard Battles



3. Complete Daily Challenges



4. Deck sharing


user initiates"createshared deck"
enter deck name and description
select visibility (private/ public)
[invite collaboration]
send invitations to collaborators
continue without collaborators
[accept collaborators
and collaborators to the deck
continue without collaborators
add card to the deck
edit or delete the cards
review deck
[approved]
save and publish deck
share deck link
go back to editing
[need changes]

Actor(s) Identified in Old FRs :

1. User (Student/Learner): The primary user who interacts with the Ankidroid app to create, review, and manage flashcards.

Goals of Primary Actors:

- **User (Student/Learner):**
    - Goal 1: Create, edit, delete flashcards with multimedia and formatting.
    - Goal 2: Review flashcards to retain information.
    - Goal 3: Sync decks with AnkiWeb and enable offline usage.
    - Goal 4: Customize learning settings (e.g., intervals, card order).
    - Goal 5: View learning progress with statistics, graphs, and retention rates.
    - Goal 6: Organize decks, rename, delete, import/export, and share with others.
    - Goal 7: Adjust study modes, difficulty levels, and intervals for personalized learning.
    - Goal 8: Quickly find specific flashcards/notes using search and filtering.
    - Goal 9: Use accessibility features (screen readers, large text) and multi-language support.
    - Goal 10: Receive reminders for daily reviews and customize notification settings.
    - Goal 11: Backup and restore flashcards locally, export/import decks in `.apkg` format.
    - Goal 12: Organize decks, rename, delete, import/export, and share with others.

Use Cases of Old FRs:

Use Case ID	Use Case Name	Primary Actor(s)	Goal

UC-01	**Create Flashcards**	User (Student/Learner)	Create, edit, delete flashcards with multimedia and formatting.
UC-02	**Edit Flashcards**	User (Student/Learner)	
UC-03	**Delete Flashcards**	User (Student/Learner)	
UC-04	**Manage Decks**	User (Student/Learner)	Organize decks, rename, delete, import/export, and share with others.
UC-05	**Review Flashcards in deck**	User (Student/Learner)	Review flashcards to retain information.
UC-06	**Mark a flashcard as easy good or hard during review**	User (Student/Learner)	
UC-07	**Customize Study Sessions**	User (Student/Learner)	Adjust study modes, difficulty levels, and intervals for personalized learning.
UC-08	**Search and Filter**	Registered User	Quickly find specific flashcards/notes using search and filtering.

UC-09	**Sync Flashcards and progress with Anki Web**	Registered User	Sync decks with AnkiWeb and enable offline usage.
UC-10	**Adjust review intervals and card order**	Registered User	Customize learning settings (e.g., intervals, card order).
UC-11	**Track Progress**	Registered User	View learning progress with statistics, graphs, and retention rates.
UC-12	**Accessibility & Language Support**	Registered User	Use accessibility features (screen readers, large text) and multi-language support.
UC-13	**Manage Notifications**	Registered User	Receive reminders for daily reviews and customize notification settings.
UC-14	**Backup and Restore**	Registered User	Backup and restore flashcards locally, export/import decks in `.apkg` format.

Use Case Testing of Old FR's:

UC#	Name	EBP test	Boss Test	Size test	Comments
UC-01	Create Flashcards	P	P	F	Adds significant value (EBP). The boss would approve. It's a single step task.
UC-02	Edit Flashcards	P	P	P	Essential for maintaining flashcards (EBP). The boss would approve. It's a small, atomic task.
UC-03	Delete Flashcards	P	P	F	Necessary for managing content (EBP). The boss would approve. It's a small, single step atomic task.
UC-04	Manage Decks	P	P	P	Organizing decks is a complete process (EBP). The boss would approve. It's a well-sized task.

UC-05	**Review Flashcards in deck**	**P**	**P**	**P**	**Core functionality of the app (EBP). The boss would approve. It's a well-sized task.**
UC-06	**Mark a flashcard as easy, good, or hard**	**P**	**P**	**F**	**Adaptive learning is valuable (EBP). The boss would approve. It's a small, single step atomic task. merged it with UC-5**
UC-07	**Customize Study Sessions**	**P**	**P**	**P**	**Personalization adds value (EBP). The boss would approve. It's a well-sized task.**
UC-08	**Search and Filter**	**P**	**P**	**P**	**Improves usability (EBP). The boss would approve. As adding filter along with search does not leave it to be a single step task**
	Sync Flashcards	**P**	**P**	**P**	**Essential for cross-device**

UC-09	**and progress with AnkiWeb**				**usage (EBP). The boss would approve. It's a well-sized task.**
UC-10	**Adjust review intervals and card order**	**P**	**P**	**F**	**Personalization adds value (EBP). The boss would approve.As it is a single step Task so we will merge it with UC-5 i.e. reviewing flashcards in deck**
UC-11	**Track Progress**	**P**	**P**	**P**	**Provides insights into learning (EBP). The boss would approve. It's a well-sized task.**
UC-12	**Accessibility & Language Support**	**P**	**P**	**P**	**Improves inclusivity (EBP). The boss would approve. It's a well-sized task.**
UC-13	**Manage Notifications**	**P**	**P**	**P**	**Improves user engagement (EBP). The boss would approve. It's a well sized task.**

UC-14	**Backup and Restore**	**P**	**P**	**P**	**Critical for data integrity (EBP). The boss would approve. It's a well-sized task.**

In order for UC-1 and UC-3 to pass All test we merged UC-1 2 and 3 in one use case Manage Cards Then it passes All Cards.

Brief Format of Use Cases:

1. Title: Manage Decks

Main success scenario:

The user accesses the "Manage Decks" section, where the system displays a list of available decks. The user selects a deck and chooses an action such as renaming, deleting, importing, exporting, or sharing. If renaming, the user enters a new name, and the system updates it. If deleting, the system asks for confirmation before removal. For import/export, the user selects a file, and the system processes the request. When sharing, the user specifies recipients, and the system sends the deck. Once the action is completed, the system updates the deck list and provides confirmation to the user.

2. Title: Review Flashcards in deck

Main success scenario:

The student selects a deck for review. The system displays flashcards according to the **spaced repetition schedule**. The student recalls the answer, flips the card, and **rates their recall (Again, Hard, Good)**. The system uses this input to reschedule the card's next appearance. The process continues until the student finishes reviewing. After the session, the system displays **progress statistics** like the number of cards reviewed and time spent.

3. Title: Sync Flashcards and progress with AnkiWeb

Main success scenario:

The system synchronizes the user's flashcards and progress with AnkiWeb. When the user initiates a sync, the system uploads local changes and downloads updates from AnkiWeb. If the user is offline, the

system saves progress locally and syncs automatically when a connection is available. The synchronization ensures that decks remain up-to-date across devices.

4. Title: Track progress

Main success scenario:

The system allows registered users to track their learning progress. When the user navigates to the progress section, the system displays statistics, including reviewed flashcards, time spent, and retention rates. The system also generates graphs and trends based on the user's study history, helping them monitor their performance over time.

Casual Format of Use cases

1. Title: Manage Decks

Main success scenario:

The user accesses the "Manage Decks" section, where the system displays a list of available decks. The user selects a deck and chooses an action such as renaming, deleting, importing, exporting, or sharing. If renaming, the user enters a new name, and the system updates it. If deleting, the system asks for confirmation before removal. For import/export, the user selects a file, and the system processes the request. When sharing, the user specifies recipients, and the system sends the deck. Once the action is completed, the system updates the deck list and provides confirmation to the user.

Alternative Scenarios:

- **User tries to create a deck with a duplicate name:** The system alerts the user and suggests renaming.
- **User accidentally deletes a deck:** The system asks for confirmation and may offer an undo option.
- **Import fails due to an unsupported file format:** The system notifies the user and provides supported formats.
- **User has no decks available:** The system displays a message encouraging them to create or import one.
- **Sharing fails due to network issues:** The system notifies the user and allows retrying later.

2. Title: Review Flashcards in deck

Main success scenario:

The student selects a deck for review. The system displays flashcards according to the **spaced repetition schedule**. The student recalls the answer, flips the card, and **rates their recall (Again, Hard, Good)**. The system uses this input to reschedule the card's next appearance. The process continues until the student finishes reviewing. After the session, the system displays **progress statistics** like the number of cards reviewed and time spent.

Alternative Scenarios:

- **User skips a card:** The system marks it as "not reviewed" and moves to the next.
- **User closes the app mid-session:** The system saves progress automatically.
- **No flashcards available in the deck:** The system notifies the user to add or import flashcards.
- **Network issues prevent sync:** The system allows offline review and syncs progress later.

3. Title: Sync Flashcards and progress with AnkiWeb

Main success scenario:

The system synchronizes the user's flashcards and progress with AnkiWeb. When the user initiates a sync, the system uploads local changes and downloads updates from AnkiWeb. If the user is offline, the system saves progress locally and syncs automatically when a connection is available. The synchronization ensures that decks remain up-to-date across devices.

Alternative Scenarios:

- **Sync fails due to poor internet:** The system notifies the user and retries when the connection is restored.
- **User has conflicting changes on multiple devices:** The system asks the user to choose between local and cloud data.
- **AnkiWeb server is down:** The system allows offline study and attempts to sync later.

4. Title: Track progress

Main success scenario:

The system allows registered users to track their learning progress. When the user navigates to the progress section, the system displays statistics, including reviewed flashcards, time spent, and retention rates. The system also generates graphs and trends based on the user's study history, helping them monitor their performance over time.

Alternative Scenarios:

- **User has no study history:** The system displays a message encouraging the user to start studying.
- **Statistics fail to load due to an error:** The system provides available data and allows the user to refresh manually.
- **User wants more detailed insights:** The system offers filters to customize data views.
- **User takes a long study break:** The system adjusts progress trends to account for inactivity.

Fully-Dressed Format of Use cases:

1.Title: Manage Decks

Use Case Name	Manage Decks
Scope	Ankidroid Enhanced Learning System
Level	User Goal
Primary Actor	Learner/Student
Stakeholders and Interests	- **Student:** Wants to efficiently manage decks for better learning organization. - **System Administrator:** Ensures deck management operations function smoothly. - **Institution (if applicable):** May track user engagement with decks if consent is given.
Preconditions	- The student must be logged in. - At least one deck must be available for management.
Success Guarantee (Postconditions)	- The system successfully updates the decks based on the performed action. - User receives confirmation of the modification.
Main Success Scenario	1. The user accesses the "Manage Decks" section. 2. The system displays a list of available decks. 3. The user selects a deck and an action (rename, delete, import/export, share). 4. If renaming, the user enters a new name, and the system updates it. 5. If deleting, the system requests confirmation before removing the deck. 6. If importing/exporting, the user selects a file, and the system processes it. 7. If sharing, the user enters recipient details, and the system sends the deck.

	8. The system updates the deck list and provides confirmation to the user.
Extensions (Alternate Success/Failure Scenarios)	- **Invalid Deck Name:** The system prevents renaming to an empty or duplicate name. - **Deck Deletion Warning:** The system asks for confirmation before deletion. - **Import Failure:** If the file format is incorrect, the system notifies the user and rejects the import. - **Sharing Error:** If the recipient is invalid or sharing fails, the system alerts the user.
Special Requirements	- Deck operations should complete within **2 seconds** for a smooth experience. - The system should **support multiple concurrent users** managing decks. - Data security must be maintained for shared decks.
Technology & Data Variations List	- Decks may be stored **locally or in the cloud** based on user preferences. - Import/export may support **Anki-specific formats**.
Frequency of Occurrence	- Learners may manage decks **daily or weekly** based on study needs.
Testing Considerations	- Ensure **rename, delete, import/export, and share** work as expected. - Test **large deck files** for performance stability.
Implementation Timing	- Deck management can be **iteratively enhanced** with cloud storage options and AI-powered recommendations in future updates.
Miscellaneous (Open Issues)	- Should the system allow **deck recovery after deletion**? - How to prevent **accidental sharing of sensitive study materials**?

2.Title: Review Flashcards in a Deck

Use Case Name	Review Flashcards in Deck
Scope	AnkiDroid Enhanced Learning System
Level	User Goal
Primary Actor	Learner/Student
Stakeholders and Interests	- **Student:** Wants an efficient review system that optimizes learning. - **System Administrator:** Ensures the review system functions correctly and spaced repetition scheduling works. - **Institution (if applicable):** May analyze student engagement trends (if permitted).
Preconditions	- The user must have at least one deck with scheduled flashcards available. - The spaced repetition system must be active.

Success Guarantee (Postconditions)	- The student completes a review session with flashcards scheduled based on their difficulty ratings. - The system updates the user's progress statistics and adjusts future review schedules.
Main Success Scenario	1. The student selects a deck for review. 2. The system presents a flashcard based on the spaced repetition schedule. 3. The student recalls the answer and flips the card to reveal the correct response. 4. The student rates their recall (**again, hard, or good**). 5. The system updates the review schedule for the card based on the selected rating. 6. Steps 2-5 repeat until all scheduled cards are reviewed or the student exits. 7. The system updates the student's progress and displays statistics (e.g., number of reviewed cards, time spent).
Extensions (Alternate Success/Failure Scenarios)	- **No Cards Available:** The system notifies the student that no flashcards are due for review. - **Early Exit:** The student can exit the session at any time, and progress will be saved. - **Review Customization:** The student may change the display settings, such as enabling auto-flip or dark mode. - **Error Handling:** If a flashcard fails to load, the system displays an error message and allows the user to skip it.
Special Requirements	- The system must ensure that flashcards are presented **smoothly and efficiently**. - The spaced repetition algorithm must adjust **review frequency** accurately based on recall ratings. - The system should display **progress statistics** at the end of each session.
Technology & Data Variations List	- Spaced repetition scheduling may use the **SuperMemo 2 (SM-2) algorithm** or a similar system. - The system may support different flashcard formats, including **text, images, and audio**.
Frequency of Occurrence	- Students may review flashcards **multiple times per day** based on their learning schedule.
Testing Considerations	- Validate **spaced repetition accuracy** to ensure cards are scheduled correctly. - Ensure the **progress statistics update in real-time**. - Test performance with **large decks** to ensure stability.
Implementation Timing	- The basic spaced repetition review system should be implemented first, followed by additional features like **AI-powered review recommendations**.
Miscellaneous (Open Issues)	- Should the system allow users to **customize their review schedule** (e.g., override the algorithm)?

	- Should the system provide **hints or explanations** for incorrect answers? - How should the system handle **missed review sessions** (e.g., rescheduling overdue cards)?

3.Title:Sync flashcards and progress with AnkiWeb

Use Case Name	Sync Flashcards and Progress with AnkiWeb
Scope	AnkiDroid Enhanced Learning System
Level	User Goal
Primary Actor	Registered User
Stakeholders and Interests	- **User:** Wants their flashcards and progress to be accessible across devices. - **System Administrator:** Ensures sync functionality works reliably without data loss. - **AnkiWeb Service:** Maintains cloud backups and syncs user data securely.
Preconditions	- The user must have an AnkiWeb account and be logged in. - The device must have internet access for syncing (unless working offline).
Success Guarantee (Postconditions)	- The user's flashcards and progress are successfully backed up and synced with AnkiWeb. - If offline, the system stores progress locally and syncs once online.
Main Success Scenario	1. The user selects the sync option in the app. 2. The system checks for an internet connection. 3. If online, the system compares local and AnkiWeb data. 4. The system uploads local changes and downloads updates from AnkiWeb. 5. The sync completes successfully, and the user's data is up to date across devices. 6. If offline, the system saves progress locally and schedules a sync for later.
Extensions (Alternate Success/Failure Scenarios)	- **No Internet Connection:** The system notifies the user that sync is unavailable and saves changes locally. - **Sync Conflict:** The system detects discrepancies between local and cloud data and prompts the user to choose which version to keep. - **Server Downtime:** If AnkiWeb is unreachable, the system retries later and notifies the user of the issue. - **Partial Sync:** If some data fails to sync, the system logs the error and retries in the next session.
Special Requirements	- The system must **encrypt user data** before syncing to ensure security. - Sync operations should **not disrupt the user experience**, running in the background when possible.

	- Offline progress storage must be **efficient and reliable**, preventing data loss.
Technology & Data Variations List	- Local progress may be stored using **SQLite or another embedded database** for offline access.
Frequency of Occurrence	- Users may sync **multiple times daily** or rely on **automatic periodic sync**.
Testing Considerations	- Test **sync reliability under poor network conditions**. - Ensure **no data loss or corruption occurs** during sync. - Verify **offline progress is correctly applied** once online.
Implementation Timing	- Initial phase should focus on **basic one-way sync**, followed by improvements like **automatic background sync**.
Miscellaneous (Open Issues)	- Should users be able to **manually resolve sync conflicts**, or should the system handle them automatically? - How should **deleted decks/cards** be managed to avoid unintended data loss? - Should **progress sync be configurable**, allowing users to sync only selected decks?

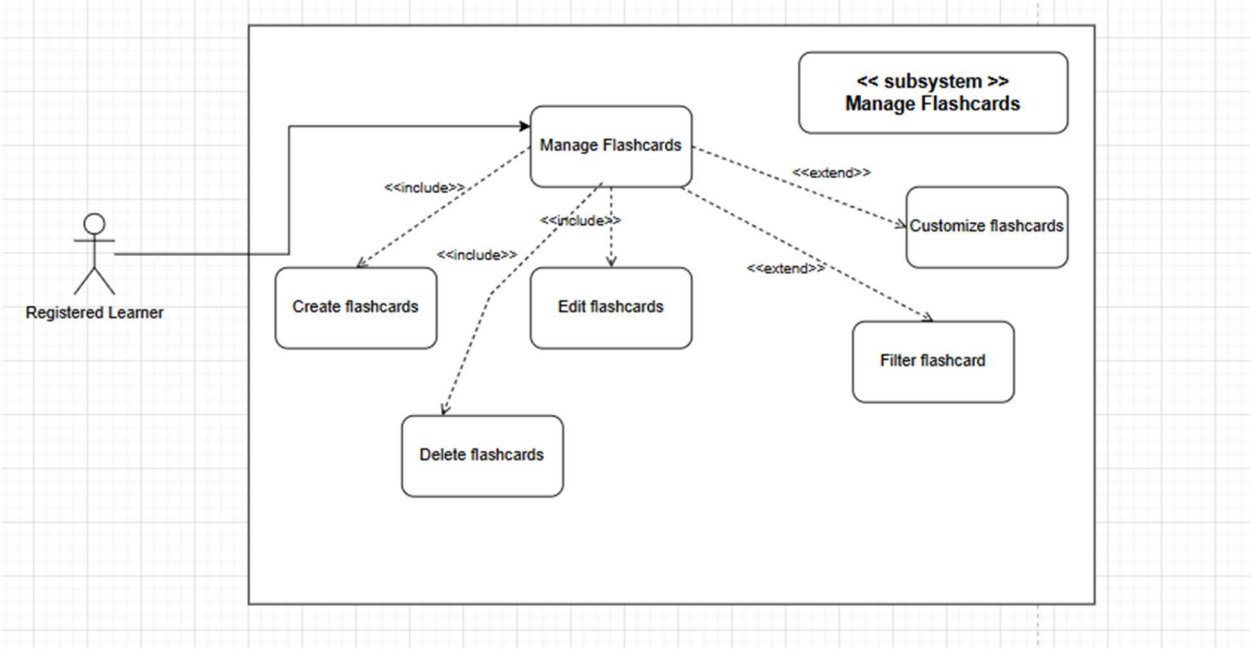
4.Title: Track progress

Use Case Name	Track Progress
Scope	AnkiDroid Enhanced Learning System
Level	User Goal
Primary Actor	Registered User
Stakeholders and Interests	- **User:** Wants to track study performance and improve retention. - **System Administrator:** Ensures accurate data tracking and visualization. - **Educational Institutions (if applicable):** May provide insights on user learning trends.
Preconditions	- The user must have completed at least one study session. - The system must have recorded study data for progress tracking.
Success Guarantee (Postconditions)	- The user successfully views their study statistics, including reviewed flashcards, time spent, and retention rates. - The system generates accurate progress graphs based on study history.
Main Success Scenario	1. The user navigates to the **Progress Tracking** section. 2. The system retrieves stored study data. 3. The system displays key statistics, such as flashcards reviewed, time spent, and retention rates. 4. The system generates graphical representations of study trends. 5. The user analyzes their progress using the provided data. 6. The system updates statistics dynamically as the user continues studying.

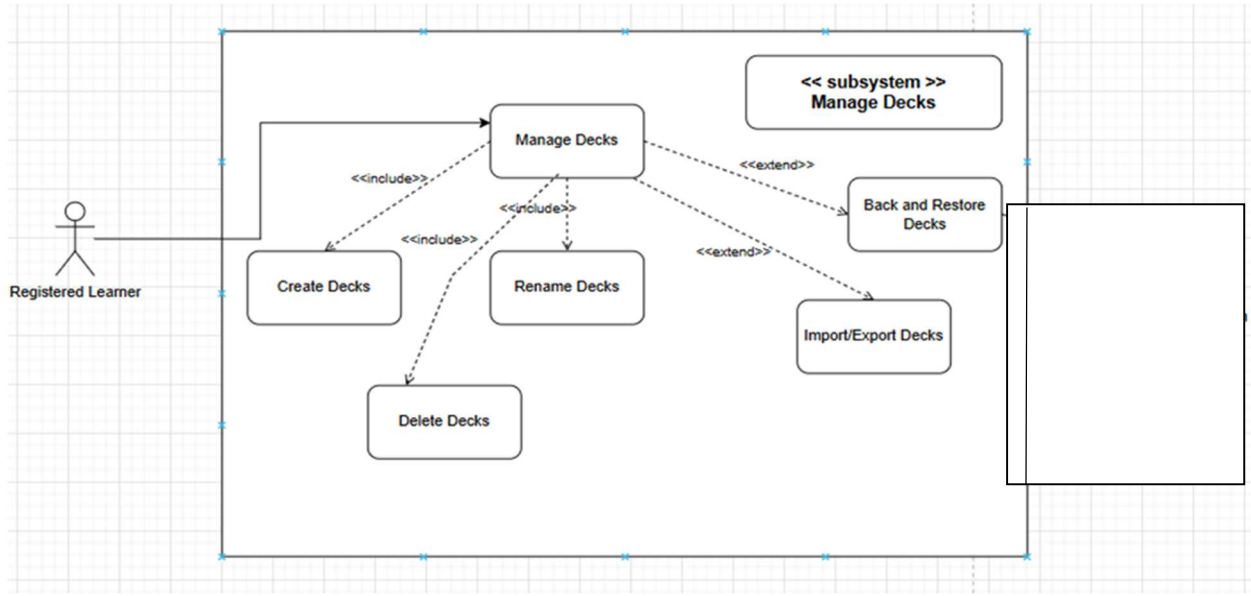
Extensions (Alternate Success/Failure Scenarios)	- **No Study Data Available:** If the user has no recorded study sessions, the system displays a message encouraging them to start studying. - **Partial Data Retrieval:** If some data is missing, the system provides available statistics and notifies the user of any missing information. - **Incorrect Statistics:** If the user suspects an error, they can request a data refresh or contact support.
Special Requirements	- The system must ensure **data accuracy and real-time updates** for progress tracking. - The UI should be **visually engaging**, using charts and graphs for better understanding. - Data should be stored **securely and efficiently** to prevent loss.
Technology & Data Variations List	- The system may use **SQLite** or **cloud-based storage** to log progress data. - Graphs may be generated using **built-in UI libraries** or external visualization tools.
Frequency of Occurrence	- Users may check their progress **daily or weekly** based on their study habits.
Testing Considerations	- Verify that **progress calculations are accurate**. - Ensure that **graphs update correctly** after new study sessions. - Test **performance with large datasets** to maintain a smooth user experience.
Implementation Timing	- Initial implementation should focus on **basic statistics**, followed by **advanced analytics and trends**.
Miscellaneous (Open Issues)	- Should users be able to **customize statistics** (e.g., choose what data to display)? - How should the system handle **long study breaks** (e.g., resetting trends vs. maintaining historical data)? - Should users receive **progress notifications** to encourage continued studying?

Use Case Diagrams:

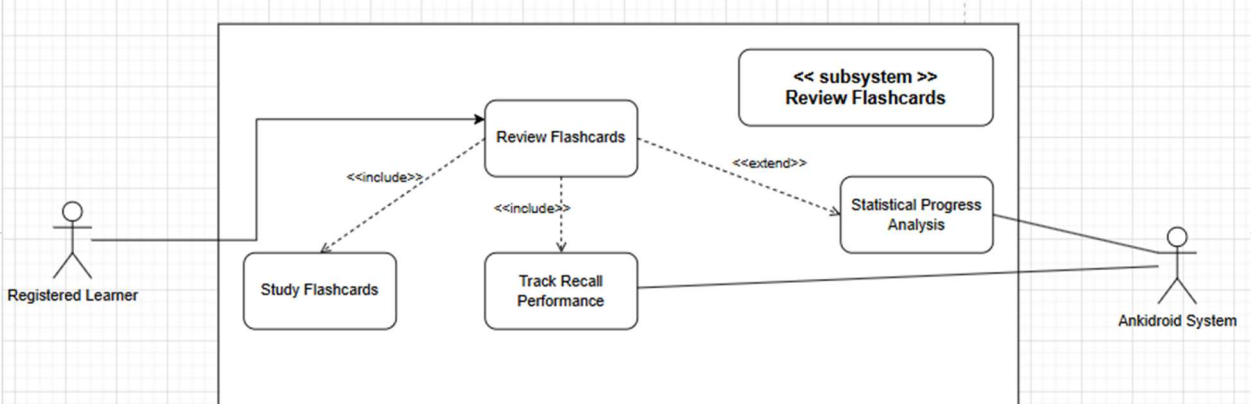
Use case 1,2,3:



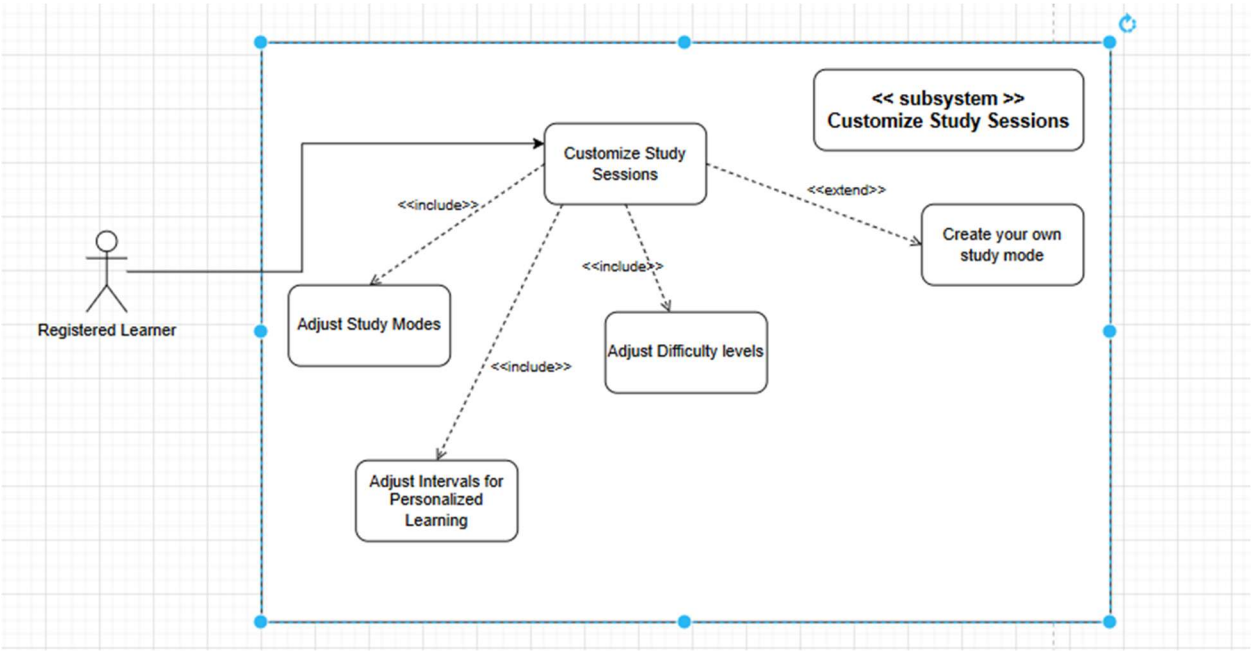
Use Case 4:



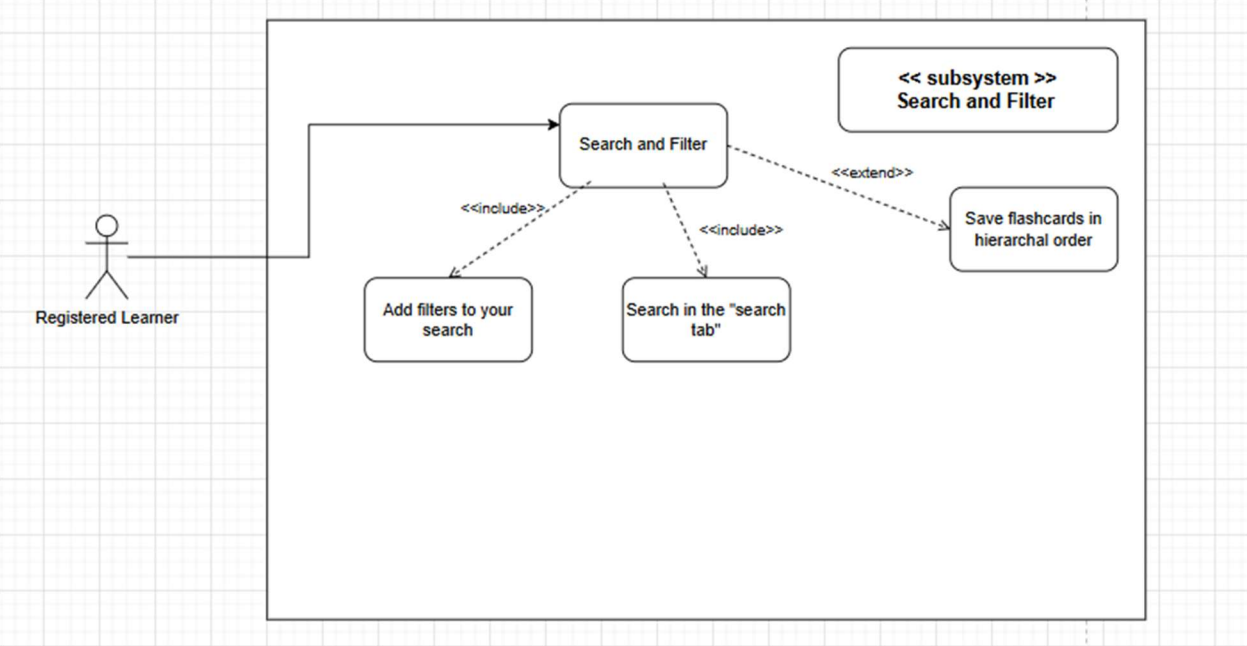
Use Case 5,6:



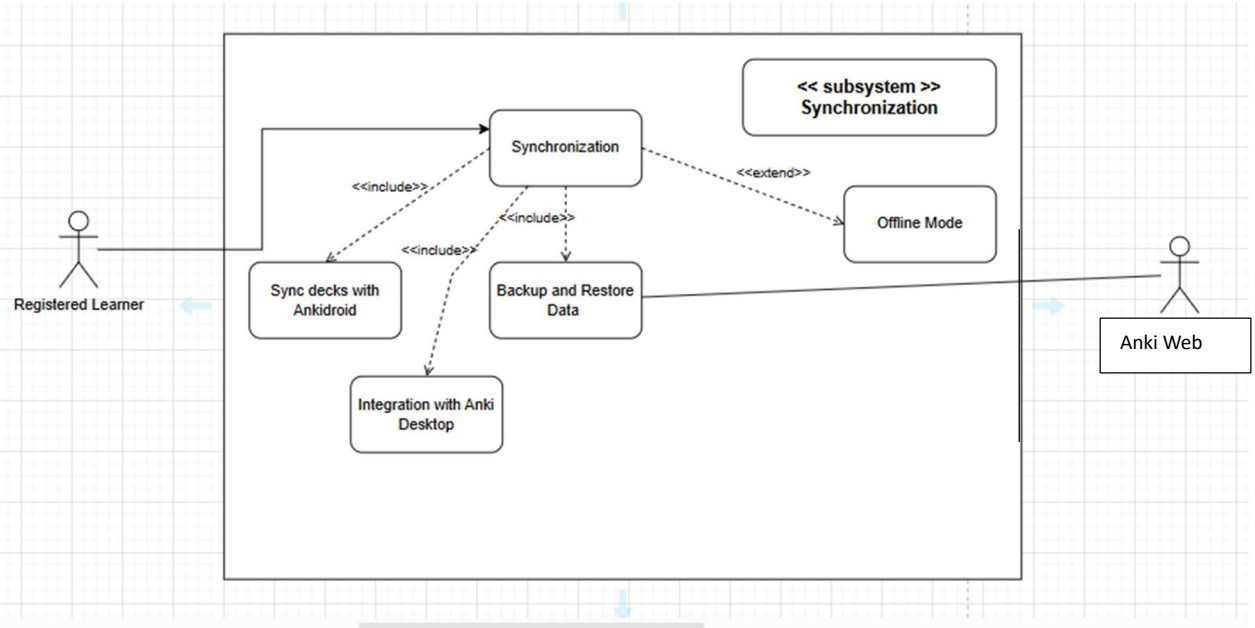
Use Case 7:



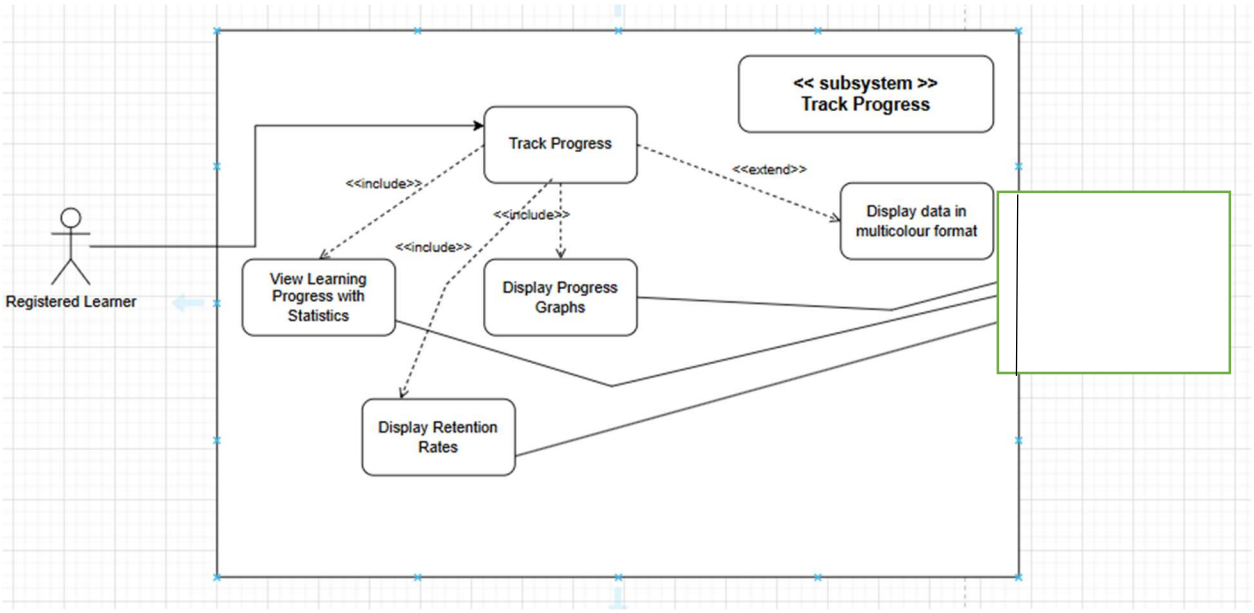
Use Case 8:



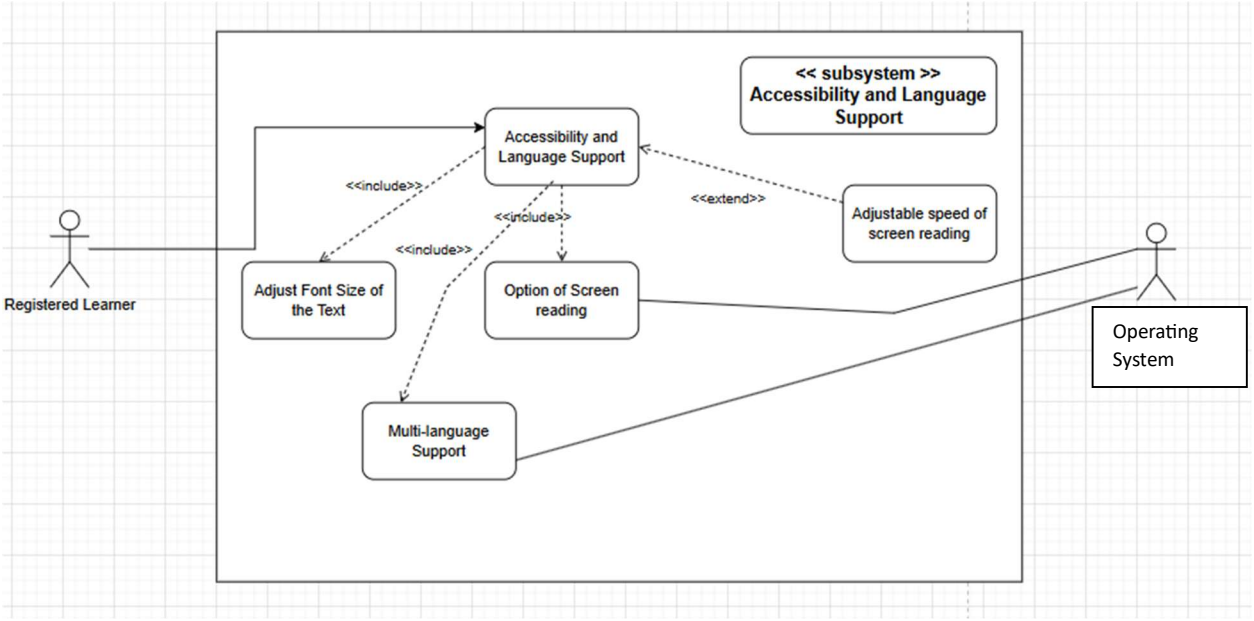
Use Case 9:



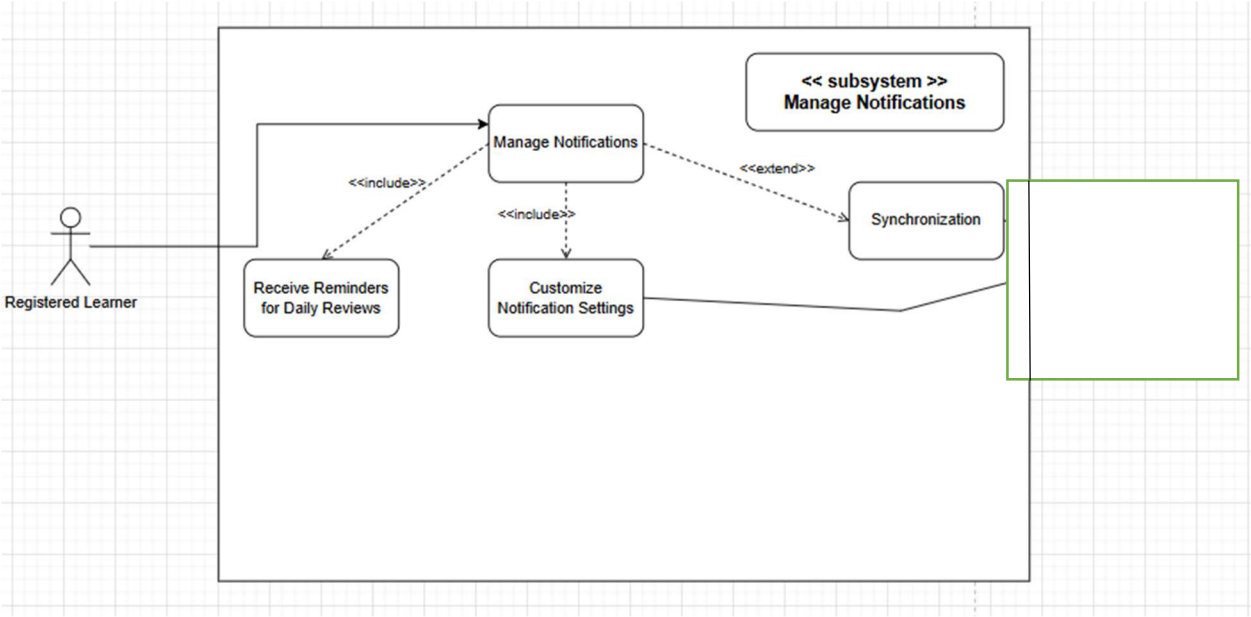
Use case 11:


<< subsystem >>
Track Progress
Track Progress
<<include>>
<<include>>
<<include>>
<<extend>>
Display data in multicolour format
Registered Learner
View Learning Progress with Statistics
Display Progress Graphs
Display Retention Rates

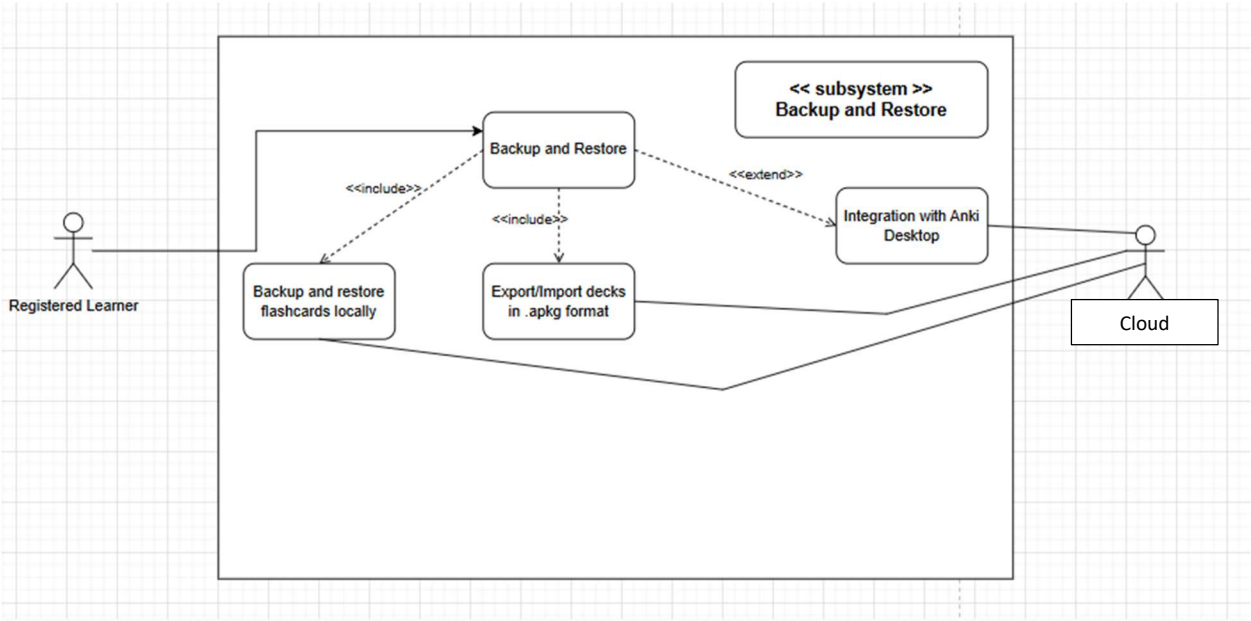
Use Case 12:


<< subsystem >>
Accessibility and Language Support
Accessibility and Language Support
<<include>>
<<include>>
<<include>>
<<extend>>
Adjustable speed of screen reading
Registered Learner
Adjust Font Size of the Text
Option of Screen reading
Operating System
Multi-language Support

Use case 13:



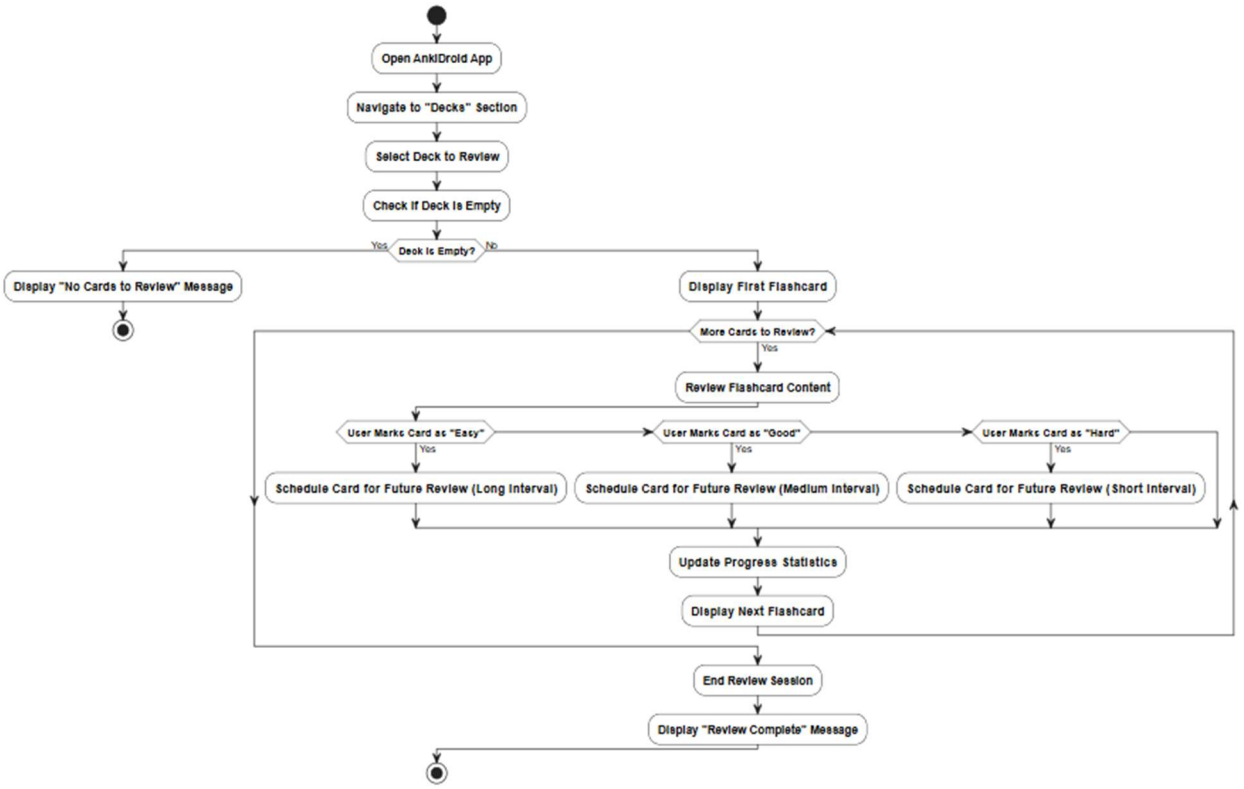
Use Case 14:



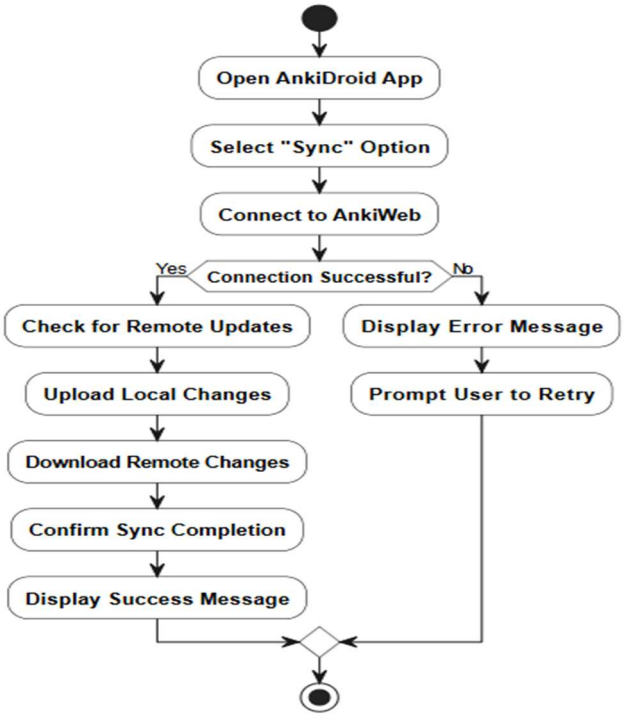
Activity Diagram:

1. Manage Flashcards:

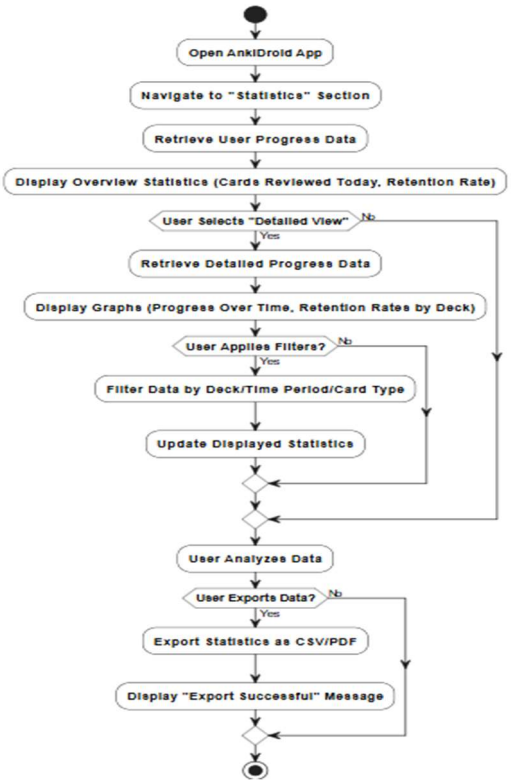
2. Review Cards in Deck:



3. Sync Flashcards and Progress



4. Track Progress:



Glossary

- **Streak**: Consecutive days of app activity.
- **Leaderboard**: Ranked list of users by achievements.
- **Spaced Repetition**: Algorithm scheduling flashcards for optimal review.