CS 3307
Group Project Stage 1 - Proposal

INTELLECTO

IGNITE MINDS, UNLOCK BRILLIANCE!

Parth Bhandari **Sara Mehravar**
Ashna Mittal **Manpreet Saini**
Siddhant Saraf

Table of Contents

1.0 Description

Welcome to the future of personalized learning with our cutting-edge revision app, designed to transform your study experience! Imagine a tool that not only adapts to your learning style but also employs advanced spaced repetition algorithms to drastically enhance your retention of information.

With Intellecto, you'll have access to a user-friendly flashcard system, meticulously crafted to streamline your study sessions. Moreover, we've infused our app with exciting optional features to enrich your learning journey. Whether you're a student aiming for academic excellence, a professional seeking to expand your knowledge base, or a lifelong learner pursuing new intellectual horizons, our application is your gateway to achieving your learning goals efficiently and easily. Join us as we redefine the art of learning, making it more effective, engaging, and tailored to your unique needs.

2.0 Features

2.1 Required Features

This section outlines the required features that are fundamental to our app. These features constitute the backbone of our minimum viable product (MVP) and are crucial for the basic operation and value proposition of the application.

- **Spaced Repetition Algorithm**: Implementing an algorithm that schedules reviews based on how well the user remembers each item.
- **Flashcards**: Users should be able to create flashcards with a front (question or prompt) and a back (answer or explanation).
- **Deck Organization**: The ability to organize flashcards into decks or categories.
- **Review Sessions**: A system for users to review flashcards, rate their difficulty, and have the application schedule the next review based on their responses.
- **Data Storage**: A robust system for storing user data, including flashcards, review history, and progress.
- **Graphic User Interface**: A clean, intuitive UI that makes it easy to create, review, and organize flashcards.

2.2 Optional Features

This section outlines a set of optional features that would significantly contribute to the app’s completeness, robustness, and overall usability. These features are designed to add depth to the user experience and enhance the practicality and appeal of the final product.

- **Gamification Elements**: Introduce a multiplayer feature where users can compete against each other in quiz challenges for added engagement and fun. Additionally,

incorporate gamification elements such as badges, achievements, or scoreboards to motivate and reward users for consistent learning.
- **Collaborative Feature**: Options for users to share decks, collaborate on creating flashcards, or compete with friends.
- **Advanced Statistics and Reporting**: Offer detailed analytics about user performance and learning progress, displayed at the end of each session, to help users track their improvement over time.
- **Import/Export**: Enable users to import and export decks, facilitating the sharing of study materials and providing backup options.
- **Customization Options:** Provide advanced customization features for flashcards, including card templates, fonts, colours, and layout, allowing users to tailor their learning experience.

2.3 Wishlist Features

This section outlines a set of ideal refinements that, while not essential for the core functionality of our project, would significantly contribute to its polish and finesse. These features are recognized as being more complex and resource-intensive, and as such, may not be feasible within the current scope of development. However, if additional resources and time become available, their incorporation could elevate the overall user experience and sophistication of the project.

- **Accessibility Features:** Making the app usable for people with disabilities, such as colour blindness and screen reader compatibility. We can implement high-contrast colour schemes and non-colour cues (like shapes or labels) to accommodate colorblind users. Additionally, we can ensure full functionality with screen readers, including descriptive alt text for images and proper content structuring.
- **Speech-Text Recognition:** Implementing a Speech-to-Text (STT) system which would have the capability to precisely transcribe users' spoken responses into text, elevating the app's usability and accessibility.
- **Hardware I/O:** Connect physical buttons and/or dedicated display to the Raspberry Pi, allowing handheld or portable operation. It is a moderate feature, requiring additional research and purchase of hardware.

3.0 Risk and Mitigations

Creating this application comes with various potential risks. Identifying, evaluating, and preparing for these risks is vital for the project's success and seamless execution. Below are some of the primary risks to be aware of and strategies for their mitigation:

- **Risk:** Technical Challenges
  Mitigation: Ensuring the app's scalability and performance, especially if it includes media content or syncs across devices.

- **Risk:** User Adoption
  Mitigation: Balancing between advanced features and maintaining a user-friendly interface for beginners.
- **Risk:** Project Management
  Mitigation: Dividing the project into smaller, achievable tasks, establishing attainable milestones, formulating a defined development timeline with set deadlines, and working collaboratively to distribute the workload are effective strategies to maintain project momentum and achieve its objectives.
- **Risk:** Quality and Reliability
  Mitigation: Ensuring the app is bug-free and reliable across different devices and operating systems.
- **Risk:** Data Loss and Corruption
  Mitigation: Implement robust data backup and recovery mechanisms, with automated data validation checks to prevent corruption.

4.0 Additional Notes

In addition to standard C++ libraries, this project can potentially use other libraries and tools for different features of the application:

- Frontend: JavaScript frameworks like React for dynamic UI elements.
- Backend: Node.js with Express for server-side logic.

Additionally, the team intends to utilize a Raspberry Pi, which will be procured from the Computer Science Department.

5.0 References

- Anki Documentation - https://docs.ankiweb.net/getting-started.html