|  |
| --- |
| Pastel workspace setup |
| Business Requirements Document (BRD)  Capstone Project BRD |
| |  |  |  | | --- | --- | --- | | Tyler L. Kemnitz | 9/11/23 | Capstone: Application Development | |

Business Requirements Document

Last Updated: 9/14/2023

# Functional Requirements:

## Detailed description of the system's functionality.

* + Study Buddy must have a flashcard creation function for the user.
  + Study Buddy must have a flashcard usability activity.
  + Study Buddy must have a flashcard correct/incorrect designation.
  + Study Buddy must have flashcard tracking and history viewing with filters.
  + Study Buddy must have a flashcard grouping system (Decks/DeckGroups).
  + Study Buddy must have a public card search.

# Non-Functional Requirements:

## Requirements related to performance, scalability, security, usability, and other non-functional aspects.

* + Study Buddy will have a flashcard management system populated by the user. (CRUD)
    - Users will not be allowed to create a flashcard without first creating a DeckName to house it under.
      * Only once a Deck is created may it be placed in a created DeckGroup.
  + Study Buddy will have a StudySession facet of the application where you build your own Study Sessions.
    - Study Sessions will allow users to create, import, remove and record their study performance in a customized flashcard session. (CRUD)
    - Study Sessions will provide the data wherein correct/incorrect cards are designated.
  + Study Buddy will have a both historical and performance tracking.
    - Tracking will include recent cards, study priority cards, and a stat meter.
  + Study Buddy will enable any user to obtain any card created by any user on the platform through a deck-building flashcard search process.

## Quality of Service (QoS) feature expectations.

* + Study Buddy should have a feature where users can:
    - * Allow users to populate flashcards with text and image data.
      * Allow users to have flashcards with language support.
      * Allow users to have flashcards read aloud.
      * Allow users to designate flashcards, decks or deckgroups as read only/edit.
      * Allow users to share flashcards, decks or deckgroups with other users.
      * Allow users to share decks or deckgroups in person with a QR Code

# Data Requirements:

## Description of data needs, including sources, formats, and storage requirements.

* + Study Buddy must have an API backend with a Dynamic Link Library (DLL) for data access.
  + Study Buddy architecture will be configured through MVC and .NET Maui.
  + Study Buddy must have functionality on both mobile and web platforms.
  + Study Buddy must use the backend Web API with a DLL for RESTful data requests.
  + Study Buddy must enable users to store their own inventory while also being able to interact with a public inventory created by all app users.

## Data migration or integration requirements.

* + Data migration will be done from a Database through HTTP requests in the API.
  + Integration of the API/DLL, MVC Website, NET Maui Mobile App, and Database must be successful.
    - Each part will be tested separately prior to integration. (GET, POST, PUT, AND DELETE)
    - Each team member working on a part will transition gradually with other team members for continuity, expedience and successful compilation.

# Timeline and Milestones:

## Sprint 1: 9/19/2023

* **Project Initiation**
  + Define project scope and objectives.
  + Identify stakeholders and team members.
  + Create a business requirements document.
* **Requirements Gathering**
  + Conduct user interviews.
  + Document user stories and use cases.
  + Define functional and non-functional requirements.
* **Backend Development - Phase 1**
  + Set up the development environment.
  + Develop user registration and authentication.
  + Implement the basic data model (e.g.User, Deck).
* **Frontend Development - Phase 1**
  + Set up the development environment for .NET MAUI.
  + Design the initial user interface (UI) wireframes.
  + Develop UI components for user registration and login.

## Sprint 2: 10/12/2023

* **Backend Development - Phase 2**
  + Implement flashcard and deck management.
  + Develop the Web API for flashcard operations.
  + Push to production server.
* **Frontend Development - Phase 2**
  + Continue UI design and development.
  + Create UI components for flashcard and deck management.
  + Implement navigation between different app sections.
* **Database Integration**
  + Connect the backend API with the database.
  + Develop data retrieval and storage functionality.
  + Ensure data consistency and integrity!!

## Sprint 3: 11/2/2023

* **Study Session Creation and Taking**
  + Implement the ability to create sessions from decks, deck groups, or flashcards.
  + Develop the flashcard-taking feature and scoring system.
  + Record session results and statistics.
* **Deck Organization**
  + Allow users to create and manage deck groups.
  + Implement features for organizing decks into groups.
  + Enable basic deck-sharing functionality.
* **User Testing and Quality Assurance - Part 1**
  + Conduct initial user testing with a focus on registration, flashcard, and deck management.
  + Identify and address usability issues and bugs.

## Sprint 4: 11/30/2023

* **Sharing and Collaboration**
  + Expand deck-sharing capabilities, allowing users to share decks with others.
  + Implement permissions for shared decks.
  + Enable collaborative deck editing (if applicable).
* **Study Tracking and Statistics**
  + Develop features for tracking user study sessions.
  + Record statistics, including correct and incorrect flashcards.
  + Provide insights and progress tracking for users.
* **User Testing and Quality Assurance - Part 2**
  + Conduct comprehensive testing of all features.
  + Gather feedback from users and fix any remaining issues.

## Sprint 5: 12/14/2023

* **Final Testing and Bug Fixing**
  + Conduct thorough testing of the entire application.
  + Address any remaining bugs or issues.
* **Deployment and Launch**
  + Prepare for the official app launch.
  + Deploy the backend API to a production server.
  + Publish the .NET MAUI app to app stores (iOS, Android).

# Approval and Sign-off:

* + Approval and Sign-off of daily work: Tyler Kemnitz.
  + Approval and Sign-off verification of sprint deliverables: Zach Thunder.
  + Approval and Sign-off validation of sprint deliverables and final deliverables: Dominic. Gruetzmacher.

# Glossary:

