Software Requirements Specification

for

<LetuStudy - Let U Study>

Version 2.0

Prepared by

Jordan Davault

Logan Westmoreland

Enkh-Od Zol-Od

Haoming Xu

Instructor:

Dr. Brent Baas

Course:

COSC 3403-01

Software Engineering I

Date: February 6th, 2023

Contents

1 Introduction

Overview

The objective of this LetuStudy app is to provide a personalized and efficient learning experience for users. With a wide range of customization options, it will cater to the unique study needs and preferences of each individual. This app will offer a comprehensive solution for students, making it easier for them to stay organized and focused on their studies. Our goal is to create a user-friendly platform that will enhance the learning experience and improve academic outcomes.

1.1 Document Purpose

The purpose of this document is to serve as a comprehensive guide for the development team, professors, and users of the LetuStudy app. It outlines the requirements, features, functionalities, and specifications of the app, including its design and development objectives, user interface, and the constraints. This document will also be used as a reference throughout the development process to ensure that the app meets the intended requirements and is delivered to the user's satisfaction. The aim is to ensure a thorough and organized progression of the project for the development team."

1.2 Product Scope

The LetuStudy will include the following features:

- A dashboard to view study materials and schedule.
- A calendar to plan and manage study time.
- A library to store and organize study materials.
- A tracker to track flashcards, quizzes and tests.
- A rewards system to incentivize good study habits.

The aim of this product is to revolutionize the way individuals' study by offering a flexible solution that caters to the unique needs of every user. This app will offer an array of functions and customization options that provide the user with only the information they need, without overwhelming them with redundant information.

1.3 Intended Audience and Document Overview

The SRS document is intended for the software development team, professors and related personnel who are involved in the development of the LetuStudy app. This document provides a comprehensive description of the functionality, performance, and user requirements for the app.

2 Overall Description

2.1 Product Perspective

This study app is a mobile application that is designed to provide users with a completely customizable learning experience, and allows users to study using methods that suit them best. The app will be designed to meet the needs and expectations of students of all ages and levels, and will be constantly updated and improved to provide the best possible experience for its users. Multiple accessibility options will be available for users with handicapped users.

2.2 Product Functionality

Main Functions: These functions will be implemented first as they are vital to what the app is.

- Study sets, **Main Function**
- "Distraction Free" mode, study slides showing on the lock screen.
- Text to Speech (Speech to Text)
- Interactive learning with ARKit
- Handicapped users friendly
- Time tracking, schedules, checklists, study & test reminders on calendar, **Main Function**
- Separate class study sets.
- Scan to Import, export and share study sets, notes, reviews, flashcards, etc.
- **Daily check-in,** learning progress tracking (points and rewards motivated), **Main Function**
- Night mode or Blue light filter mode
- Option to delete selected data with confirmation prompts to prevent accidental loss of work. **Main Function**

For further description, please refer to section 3.2.

2.3 Users and Characteristics

Potential users of this app may be:

- Students
- Teachers
- Independent learner
- Handicapped users who have/are:
    - Deaf/hard of hearing
    - Visually impaired/ blind

- Dyslexic
- Distractible
- Epileptic
- Learning disorders
- Limited mobility

The intended user base of the application encompasses individuals at any educational level, both formal and informal. The primary focus is to cater to the needs of higher education students, while consideration for younger students will be evaluated at a later stage of development. It is of lesser priority to accommodate individuals who are not expected to frequently utilize the application, particularly those below college and high school level.

Individual users possess diverse preferences for application interfaces, and these preferences may be influenced by physical disabilities that impact sight, hearing, and touch sensitivity. The application aims to facilitate user interaction with minimal reliance on the touch screen, if necessary, and provides accommodations such as a text reader and speech writer. Additionally, users have varying learning styles, which are accommodated through customization options such as "No Distraction" mode and personalized notifications. While the application is not designed specifically for individuals with disabilities, they are a demographic present among all students, and thus their accessibility requirements have been taken into consideration.

2.4 Operating Environment

The software is designed for operation within the iOS environment and must be compatible with its underlying architecture. The minimum requirement for users is to have an iPhone, with the exact versions yet to be determined and to be updated in the near future. If a version for Android is developed in the future, similar requirements will apply.

2.5 Design and Implementation Constraints

The LetuStudy app will be developed using Swift programming language and integrated development environments such as VS Code or Xcode. The choice of IDE may vary depending on the project needs. The implementation of locking screen features is not permitted by Apple security guidelines. The development team must have a strong proficiency in Apple's macOS and iOS operating systems, as well as the Swift programming language.

One of the significant challenges in this project could be the submission of the app to the App Store, as it must meet the standards set by Apple and there is a risk of rejection. The team will strive to create a functional and impactful app; however, it is possible that the app may not be fully completed before submission.

2.6 User Documentation

The LetuStudy app will provide an instructional tour to familiarize users with the available features and functionalities upon completion. The implementation of the tutorial will be prioritized later in the development process while the initial focus will be on refining the features. The tutorial will be succinct, as it is anticipated that the majority of the options and features will be intuitive and easily understood by the user.

2.7 Assumptions and Dependencies

Personal information will be collected for account creation purposes, including social media account identification, username, first name, last name, title, date of birth, gender, and school information.

The app will be developed using Xcode 14.0 or later, and a machine running macOS 12.5 or later is required to compile the software for the iOS mobile platform.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The LetuStudy app will feature multiple touch screen interfaces, including the Home Screen, Settings, and Flashcards, as well as additional interfaces that may be added at a later time. The Home Screen will display relevant user statistics and provide navigation tabs located at the bottom of the screen. The Settings screen will include various buttons and sliders to adjust the visual and accessibility options of the app. The Flashcards screen will provide the ability to create and manage flashcards, including the option to add new cards through manual input or by importing them from a link, as well as the ability to sort previously created cards into sets defined by the user.

3.1.2 Hardware Interfaces

Hardware Interface 1: Camera

- **Data format**: JPEG, HEIC
- **Communication protocol**: Native API
- **Functionality**: The app will use the device's camera to capture images saved as study notes or flashcards. The images will be stored in JPEG or HEIC format.
- **Error handling**: If the camera fails, the app will display an error message and log the error.

- **Performance requirements**: The camera must provide a minimum resolution of 10 megapixels and capability to continuously take photos as input.
- **Security requirements**: The app must provide the ability to secure the images captured by the camera using encryption.

Hardware Interface 2: AR Sensor

- **Sensors**: An accelerometer, gyroscope, and magnetometer is required to support the AR experience. A mobile device with ToF sensor (LiDAR on iPhone) is optional, but could significantly improve the AR experience.
- **Communication protocol**: Native API
- **Functionality**: The app will use the device's AR Sensor to enable the user's augmented reality experience and interactive learning.
- **Error handling**: If the major sensors fail to enable, the app will display an error message and log the error. If the optional ToF sensor could not be found or fails to enable, the app will display a warning message about potential performance issues, and the user will be able to choose to hide the message for the future.
- **Performance requirements**: The AR sensor must have a response time of less than 50 milliseconds to ensure a seamless and responsive AR experience, which ensures the correct and accurate tracking of the object’s movements and orientation.
- **Security requirements**: The app must provide the ability to secure the 3D scanned data captured by the AR sensors using encryption.

3.1.3 Software Interfaces

Software Interface 1: Data Export

- **Data format**: XML or JSON
- **Functionality**: The app will provide a data export feature that allows users to export data in a XML or JSON format and share with other users or backup.
- **Error handling**: If the export fails, the software will display an error message and log the error.
- **Performance requirements**: The data export must complete within 10 seconds for a maximum of 10,000 records.

Software Interface 2: Data Import

- **Data format**: XML or JSON
- **Functionality**: The software will provide a data import feature that allows users to import data in a XML or JSON format from another app or backup.
- **Error handling**: If the import fails, the software will display an error message and log the error.
- **Performance requirements**: The data import must complete within 10 seconds for a maximum of 10,000 records.

3.1.4 Communications Interfaces

The LetuStudy app incorporates the use of OAuth, a third-party authorization API, to provide users with the option to log in through their social media accounts such as Google, Facebook, and Twitter. To ensure privacy and security, all communications within the app will be managed through the App Transport Security (ATS) feature and conform to the iOS security standards.

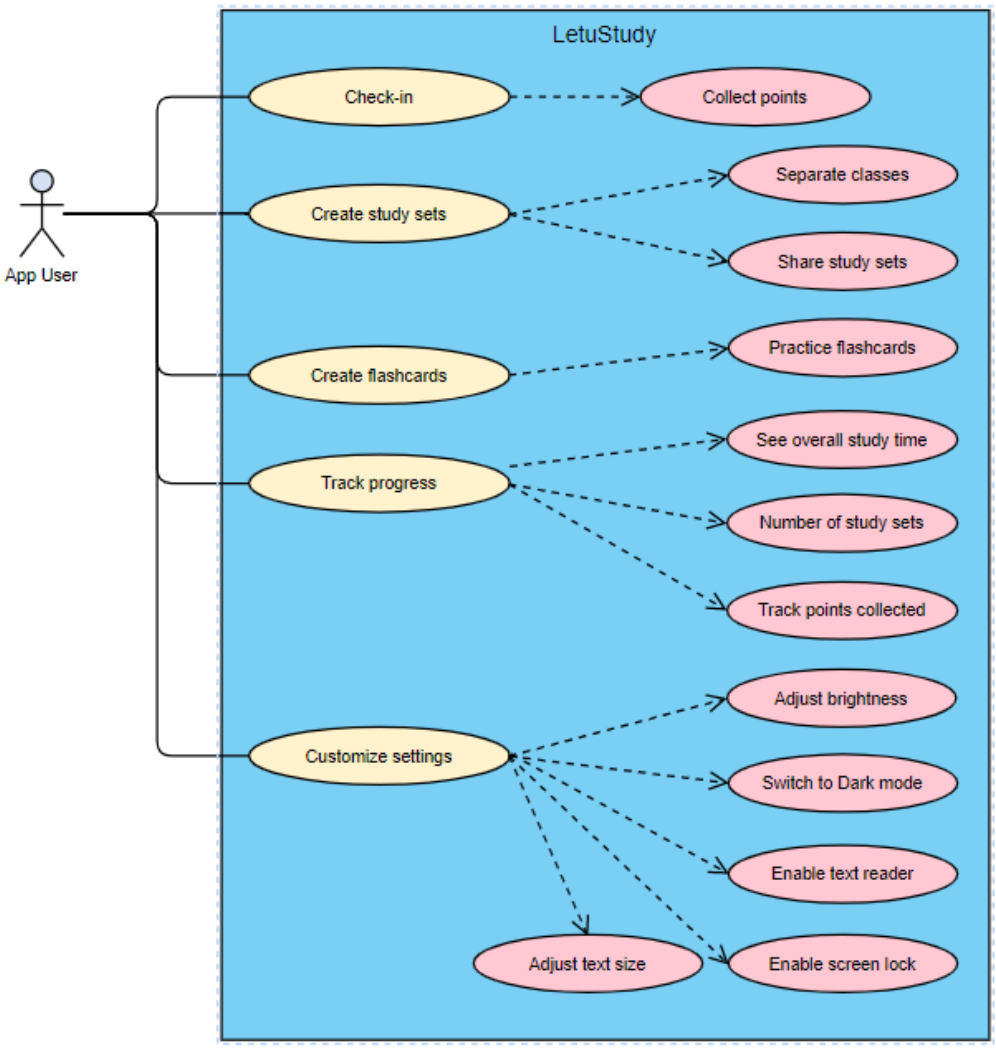
3.2 Functional Requirements

- **Phone Locking**: The implementation of phone locking functionality within the app may not be feasible on the iOS platform. However, the app will offer an opt-in option for users who wish to lock their phone until a specified number of questions are answered. The user will have the option to determine whether the answers must be correct and the number of questions required for unlocking their phone.
- **Separate class study sets**: The feature enables users to organize their study sets by class, facilitating efficient and aesthetically pleasing navigation within the app's interface.
- **Import/Export study sets**: The app will provide the capability for users to import their existing study sets and reviews, as well as export their study sets to others through sharing options. This feature will enhance the user's experience and provide increased flexibility in their study efforts.
- **Calendar**: The app will provide a customizable calendar feature for users to keep track of upcoming test or assignment dates and set reminders. It will also allow users to keep track of the study sets they need to prepare for, to avoid missing any important study material.
- **Checklist**: This feature would enable the user to keep track of their completed tests and assignments, as well as those they have studied for, by marking them off on a list. This provides a clear visual representation of their progress and helps them stay organized.
- **“Distraction Free” mode**: The app will offer users the option to initiate a timed study session, which will help them effectively manage their study time and include built-in break intervals. This feature aims to encourage good study habits and minimize distractions.
- **Text to Speech** (**Speech to Text**): This feature will allow users with vision or hearing difficulties or those who have trouble focusing to input text using their voice or listen to text, making the study app more convenient for them. The Text to Speech (Speech to Text) feature will help users with disabilities to study more efficiently.
- **Time tracking**: This feature will allow users to track their usage time on the app, giving them insight into the amount of time they have dedicated to studying, and allowing them to make adjustments if necessary. This time tracking feature will provide users with a simple and easy-to-use logging system to keep track of their study habits.
- **Interactive learning with ARKit**: The app will feature the use of ARKit, which is Apple's framework for creating Augmented Reality experiences. By utilizing ARKit, students will be able to engage in interactive learning and combine functions while interacting with real-world objects. This will bring a new level of immersion and engagement to the study experience.
- **Handicapped users friendly**: This feature would make the app more accessible for students with hearing and visual impairments, offering special instructions and functions that cater to their needs, making the app easier for them to use.

- **Daily check-in**: This feature will allow users to keep track of their daily learning progress, collect points and earn rewards. The daily check-in will encourage users to stay on track with their studies and monitor their progress over time.
- **Dark mode, brightness/color shift**: The app will have a dark mode and the ability to shift brightness and color, to offer users more control over the visual appearance of the app, helping with easier viewing and/or for aesthetic purposes.
- **Night mode**, **Blue light filter mode**: This feature would allow users to enable a night mode or blue light filter mode to reduce blue light and minimize the strain on their eyes while using the app, especially during nighttime. This can help promote healthy sleep habits and improve the user experience.
- **Image scanner**: The app will have an image scanner feature that allows users to take pictures of physical study materials, such as textbooks or notes, and extract any legible words. This feature makes it easier for users to study by allowing them to digitize and store their materials in one place.

3.3 Behavior Requirements

3.3.1 Use Case View



4 Other Non-functional Requirements

4.1 Performance Requirements

1. The app can function whether the user is connected to the internet. The user will be able to access and use their study materials, schedules, and other data even without an internet connection.
2. The app will ensure that the user's data is secure and accessible from any device once they log in to the app with their account. The encrypted cloud database will protect the user's data from unauthorized access and ensure that the data is secure even if the device is lost or damaged.
3. Users will be allowed to share their flashcards quickly and easily with others in under 10 seconds, ensuring that their study materials are easily accessible to their classmates, friends, or study group.
4. Ensure users are able to set up their preferences quickly and easily for using the app, without having to spend a lot of time configuring settings or navigating through complicated menus. The goal is to make the setup process fast, simple, and straightforward, so that users can start using the app as soon as possible.
5. The app should be accessible and usable for people with disabilities, including those who are visually or hearing impaired. The app should have minimal visual and auditory output and provide alternative ways to access the features, such as through text-to-speech or AR-based interactive learning.
6. Frequent data saving of user study sets is required to mitigate the risk of unintended loss of data, as referenced in the initial requirements.

4.2 Safety and Security Requirements

Safety Requirements: No physical safety requirements, but information safety requirements apply.

- User data must be securely stored to prevent the risk of data loss, including study sets and other important information.
- A confirmation prompt will be implemented to prevent accidental deletion of important information. Users will receive a warning before any data is permanently removed.
- The app's lock feature, if provided, must be implemented in a manner that ensures the safety of the user's device and data. No data should be lost or damaged during the locking process.

Security Requirements:

- The app must ensure the security and protection of login credentials and linked account information through safe storage measures.
- The integrity of user data must be regularly maintained and protected against unauthorized access.

4.3 Software Quality Attributes

The primary focus of the app development will be on usability and availability to ensure the app meets its intended purpose. As the project progresses, the focus will shift towards reliability and reusability to provide a reliable and user-friendly study experience. Data integrity will also be a crucial aspect that will be frequently checked to prevent unauthorized access and ensure the protection of user information. The aim is to continually evolve the app to meet the changing needs and expectations of users.

Appendix A - Data Dictionary

API - Application Programming Interface, allows two or more programs to communicate.

ARKit - An augmented Reality API developed by Apple Inc.

ATS - App Transport Security, a networking feature improves privacy and data integrity by Apple Inc.

HEIC - High Efficiency Image File Format

IDE - Integrated Development Environment

iOS - A mobile operating system developed by Apple Inc.

JPEG - A lossy compression for digital images format

JSON - JavaScript Object Notation, an open standard file format and data interchange format.

LiDAR - Light Detection And Ranging, a 3D laser ToF scanning system.

macOS - A Unix operating system developed and marketed by Apple Inc, formerly Mac OS X.

OAuth - An open standard for access delegation, allows user login with third-party accounts.

Swift - A high-level general-purpose, compiled programming language developed by Apple Inc.

ToF - Time of Flight, a range imaging system for measuring distances between camera and the subject.

VS Code - Visual Studio Code, a source-code editor made by Microsoft, for Windows, Linux and macOS.

Xcode - Apple's integrated development environment (IDE) for macOS

XML - Extensible Markup Language, a markup language and file format for storing, transmitting, and reconstructing arbitrary data.

Appendix B - Group Log

Group Meetings:

First meeting on January 29th, 2022, 4 - 6 pm in AFSC

- Mainly focused on Section 2.2 & 3.2.
- Adding more details to the project and SRS document.

Second meeting on February 4, 1 - 3 pm in Library

- Finishing up SRS document, first design of UI added.
- Finalized name as “LetuStudy”, and slogan “Let U Study”.
- Final Proofreading & Formatting.
- Rough draft of the use-case diagram added.