# Introduction

**Purpose**

**Scope**

**References**

# Overall Description

### **User Needs**

# Requirements

## Functional Requirements

### Core Functional Requirements

### Non-Core Functional Requirements

## Non-Functional Requirements

This section focuses on three main areas of the system: Performance, Usability, and Security. These requirements, while not intrinsic to the functionality, nor core design of the StudyBuddy Hub App, are nevertheless important to the long-term reliability, and user satisfaction of this application.

### Performance

The performance of a system in this context refers to how quickly, and efficiently the system can accomplish a given task. Good performance is essential for smooth workflow, allowing the user to continue using the app to study while experiencing minimal interruptions.

* To reduce loading times, any given screen should not be cluttered with too many elements, especially ones that require lots of storage, such as videos.
* In group-chats, there should be a limit to messages and file sizes to reduce loading times.
* The system should not be entirely dependent on Wi-Fi to run, and have features that can be accessed offline, such as study materials.

### Usability

In this context, usability is a measure of how easy it is for any given user to utilize the core functions of the app. If the app proves too difficult, and/or confusing to use for an average user, it will discourage them from using it. This is counterintuitive to the purpose of the app, which is to make studying an easier experience. This is why ensuring user accessibility is an important aspect of the system.

* The system should minimize the number of steps, and clicks needed to get to any given screen, or perform any given task.
* The system should utilize quick access to the main functions of the app, such as creating study groups.
* While providing quick-access to the main functions of the app is important, functions that involve more permanent changes, such as deleting study groups, should have fail-safes put into place to prevent accidental activation.
* To improve readability, screens should not be overcrowded with text. If a large amount of text is required for the screen, it should be properly formatted, spaced out, and/or, if applicable, divided into different sections or separate screens.
* The system should aim to avoid screen clutter by using minimized UI, improving ease of navigation.
* The system’s UI should be dynamic and be able to support a wide range of screen resolutions. (i.e., be formatted properly even on different screen sizes)
* The system should allow the user to customize the appearance of the app to some extent, to better fit their needs (Such as including themes, like light and dark mode)
* The user-flow/navigation should be straightforward, and functions should be placed in locations where the user would expect to find them. (ex: the “Create Study Group” Function should be under the “Study Groups” tab, as opposed to, say, “Settings”)

### Security

The security of a system refers to the protection of user data, and information from unauthorized access. Regardless of how “good” any given system is at accomplishing their task, it is a poor system if its security is poor, as it puts users at risk of having sensitive information stolen. Necessary precautions should be put into place to maintain the security and confidentiality of users, and their information.

* The system should not ask for, nor store any non-essential information from the user.
* The system should use two-factor authentication, security questions, and/or some other additional ways of verifying the user’s identity to both improve account security and help aid in account recovery should the user lose access to it for any reason.
* Any personal information about the user, such as their real name, or birthday, should be hidden from other users by default.
* At minimum, basic system security should be employed to protect the private information of the user’s account, such as password encryption.