

---

# **Software Requirements Specification**

**for**

## **StudLog**

**Version 1.0 approved**

**Prepared by**

**Gabriel Santos**

**Zach Jilani**

**James Donaldson**

**3Guys**

**14 December 2022**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>3</b>
<b>1. Introduction</b>	<b>1</b>
1.1	Purpose 1
1.2 Document	Conventions 1
1.3 Intended Audience and Reading	Suggestions 1
1.4 Product	Scope 1
1.5	References 1
<b>2. Overall Description</b>	<b>2</b>
2.1 Product	Perspective 2
2.2 Product	Functions 2
2.3 User Classes and	Characteristics 2
2.4 Operating	Environment 2
2.5 Design and Implementation	Constraints 2
2.6 User	Documentation 2
2.7 Assumptions and	Dependencies 3
<b>3. External Interface Requirements</b>	<b>3</b>
3.1 User	Interfaces 4
3.2 Hardware	Interfaces 3
3.3 Software	Interfaces 3
3.4 Communications	Interfaces 3
<b>4. System Features</b>	<b>4</b>
4.1 System Feature	1 4
4.2 System Feature 2 (and so on)	4 4
<b>5. Other Nonfunctional Requirements</b>	<b>4</b>
5.1 Performance	Requirements 8

5.2Safety		Requirements	5
5.3Security		Requirements	5
5.4Software	Quality	Attributes	5
5.5Business		Rules	5
5.6 Software Evolution			
5.7 Time and Resource			
5.8 Software Change Management			
5.9 Future Goals			
<b>6. Other Requirements</b>			<b>5</b>
<b>Appendix A: Glossary</b>			<b>5</b>
<b>Appendix B: Analysis Models</b>			<b>5</b>
<b>Appendix C: To Be Determined List</b>			<b>6</b>

## Revision History

Name	Date	Reason For Changes	Version

# **1. Introduction**

## **1.1 Purpose**

StudLog is an application to guide students to the correct amount of study time required for them to obtain an adequate grade to pass their class. As well as keep track of your own GPA progression throughout your semesters in college. This product was thought up and developed by fellow students who saw the issue with study times not being clearly displayed or communicated to the students. We want to enable the counselors with the ability to give new and incoming students clear guidance on how many hours a week they should be studying. We also want to eliminate so of the confusion with how the GPA system works by giving the students a tool where they can simulate various GPA outcomes throughout the semester to better advocate for their grade and overall GPA they are trying to obtain.

## **1.2 Document Conventions**

In this document, every requirement is to have its own priority.

## **1.3 Intended Audience and Reading Suggestions**

We will give a high-level overview of the application and what the requirements are for developers and project managers that will be involved in the development of StudLog in this current iteration. The SRS will be organized in a sequential order of overall description, specific requirements, non-functional requirements, and finally the organization of the application with given models for better understanding. This document should give the reader a basic understanding of the construction and operation of the application.

## **1.4 Product Scope**

StudLog will be a product generally geared towards the student body, and strongly towards incoming students. We want the students to have the best understanding of what is expected of them and what is required to pass a class. All the hours that we will be displaying for study time will be a bare minimum requirement to obtain an A in your class. We also want to take the guess work out of the calculations of the GPA and allow the students to see the importance of striving for an A in each class. Finally, we implemented a easy to use basic function for counselors to give a general overview of their students semester.

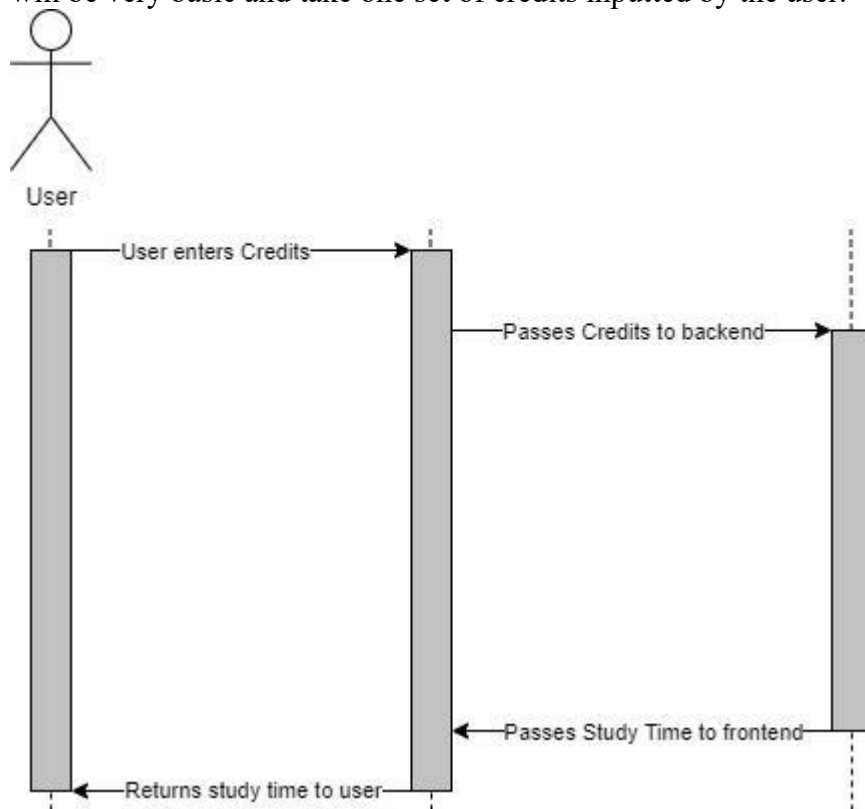
## **1.5 References**

No references currently.

## 2. Overall Description

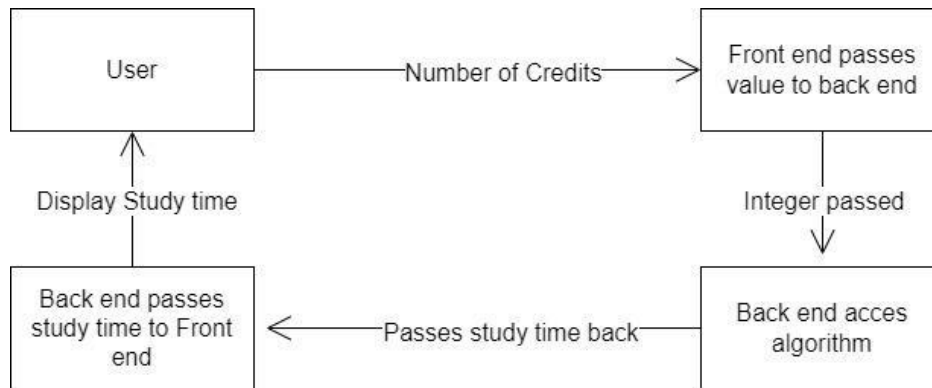
### 2.1 Product Perspective

The creation of StudLog is a brand-new product being developed by 3Guys. The purpose of this product is to help establish a good study habit for students who are just starting their college career. We found the need for this product to help with students that are dropping their major due to poor planning or time management. This product will be a self-contained system that takes the users credits and returns an amount of study time needed for the class. The first iteration of this product will be very basic and take one set of credits inputted by the user.



### 2.2 Product Functions

StudLog will perform multiple functions such as giving a general overview of total study hours per semester to a more in-depth breakdown provided adequate user input. We hope to provide a general overview of study time along with a class break down with difficulty factored in. Finally, we will be providing a GPA calculator to the student for reference.



## 2.3 User Classes and Characteristics

The core users for the product will be freshmen attending college. We expect a smaller number of users as the students move up in grade to continue to use StudLog. But there will be a rather steep drop off in users as college experience is accumulated. We also expect a small number of high school students to use the application as well as counselors for incoming or current students.

### Expected Users:

1. Freshmen in college
2. Higher level college students
3. High school students
4. Counselors

## 2.4 Operating Environment

The application will run through java machines on computers that have up to date java machines on their personal computers. This will be a relatively easy operation environment to obtain due to the widespread availability of java machines on laptops by default. The only operating system that will cause a hindrance is Mac OS. This is due to their closed system for general applications.

## 2.5 Design and Implementation Constraints

The design constraint that we have implemented is that the user will only be able to enter integers into the credit fields. This will help prevent user errors for the application and will also provide a level of security from SQL attacks.

## 2.6 User Documentation

No current manuals needed.

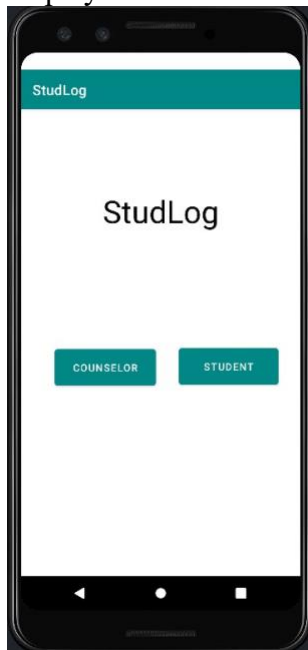
## 2.7 Assumptions and Dependencies

Current dependencies are for the user to have an Android device that can run this application. We developed this application to run on most if not all current Android devices in service today.

# 3. External Interface Requirements

## 3.1 User Interfaces

The user interface is created in Android Studio and is expected to run on an Android device. It is a multi-panel display that the user will transverse through with various touch buttons. They will move through a total of seven screens if they use the application to its full capacity at this time. We will look to improve the interface and capabilities in future versions. All panels of the application will be displayed below as we review the features of the application.



## 3.2 Hardware Interfaces

The current hardware interface that is needed is a Android cellphone.

## 3.3 Software Interfaces

This product is an application that will run on Android devices and will require, and API updated to the earliest version of 30. And model that is more recent than this update will be able to support this application and function as expected.

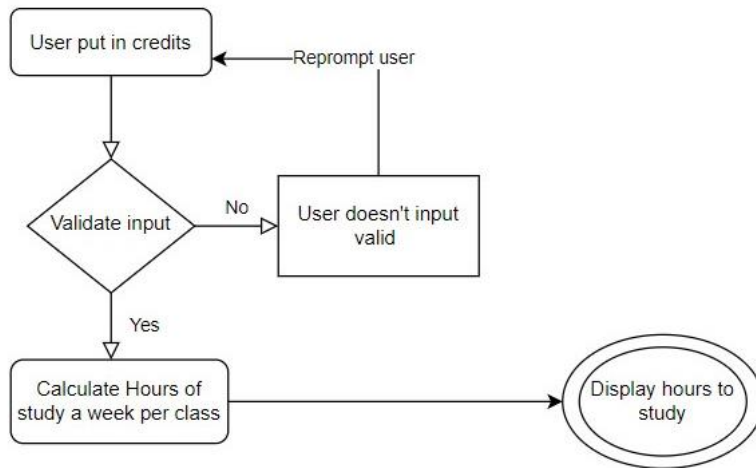
### 3.4 Communications Interfaces

There are no communication interfaces that are required by this application at this time.

## 4. System Features

### 4.1 System Feature

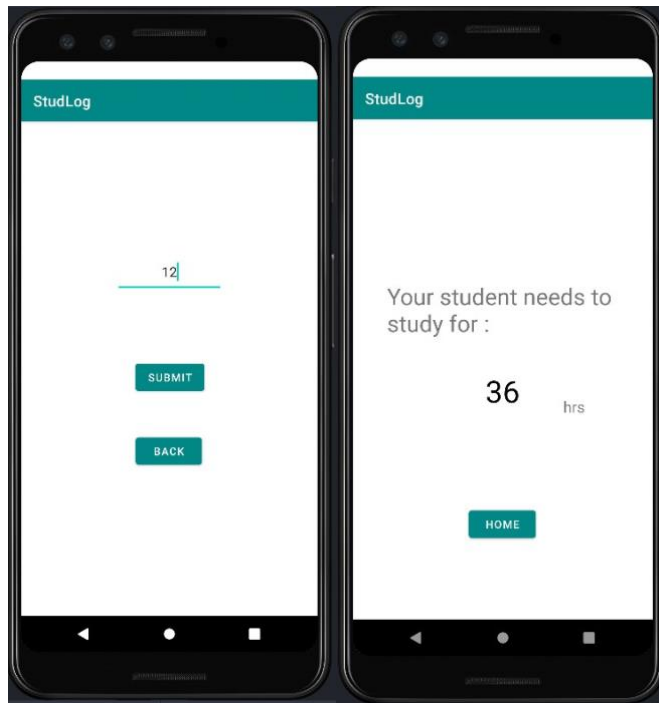
4.1.1 We will prompt the users for their total amount of credits they are taking for this semester. The application will then compute the amount of study time to pass the users classes for the semester. This version became the basis for the counselor portion of the application to give a general basis overview of the semester for the student.



#### 4.1.3 Functional Requirements

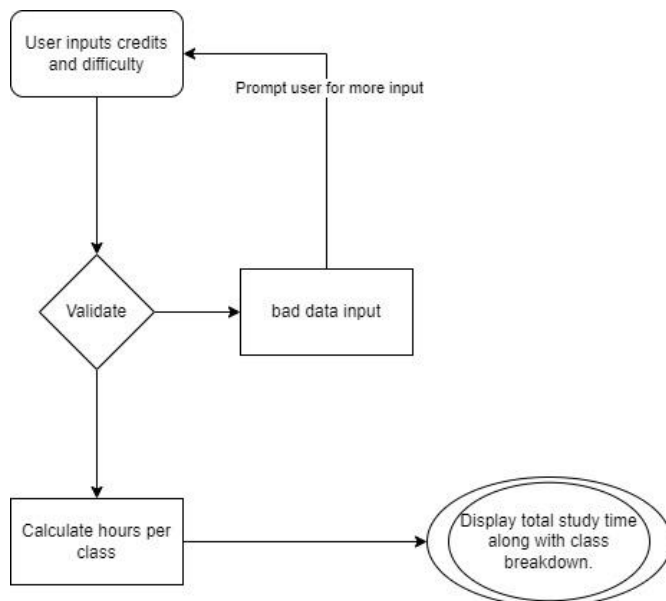
REQ-1: The counselor will be prompted to enter the total amount of credits that the student is taking this semester. And the program will use that to return a base amount of study hours with no variations in difficulty. Screen shots of this application in process are displayed below.





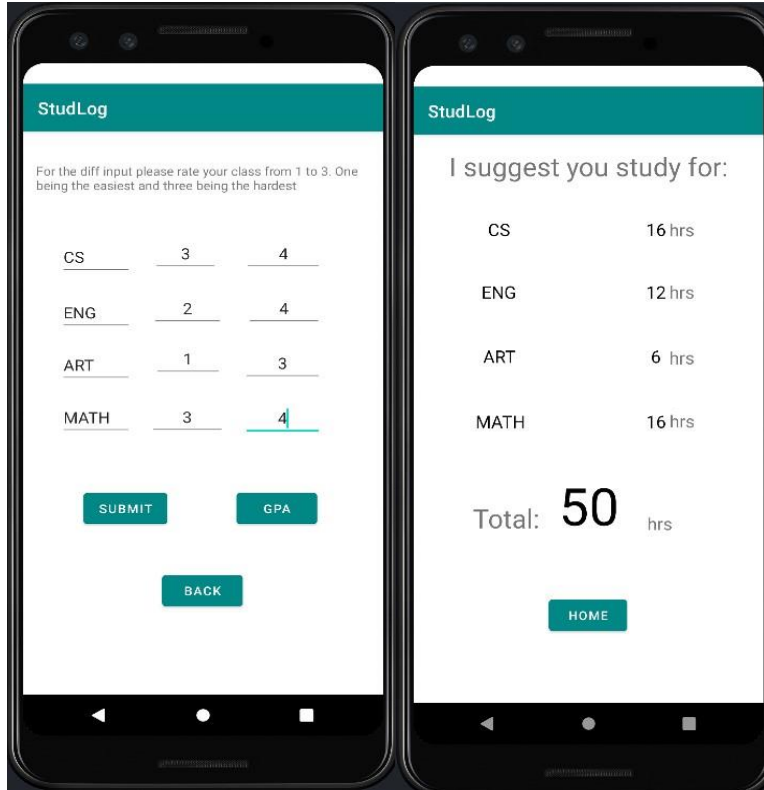
## 4.2 System Feature 2

4.2.1 The second feature that we are providing is geared toward the students themselves and this is the semester portion of the class schedule that is broken down into more detail. The user will be able to list the classes that they are taking along with the expected difficulty level and total amount of credits.



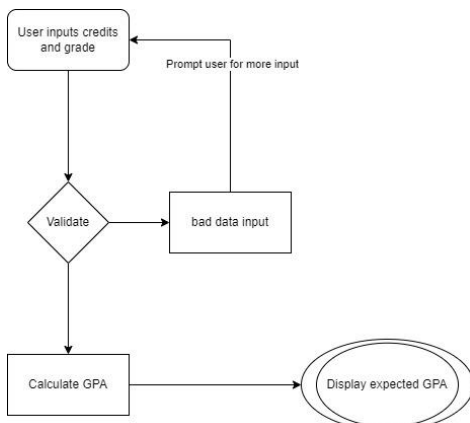
### 4.2.3 Functional Requirements

REQ-1: The student will be required to input their classes by name as well as an integer value for their difficulty level along with their credits per class taken. This is pictured below with screenshots of the application.



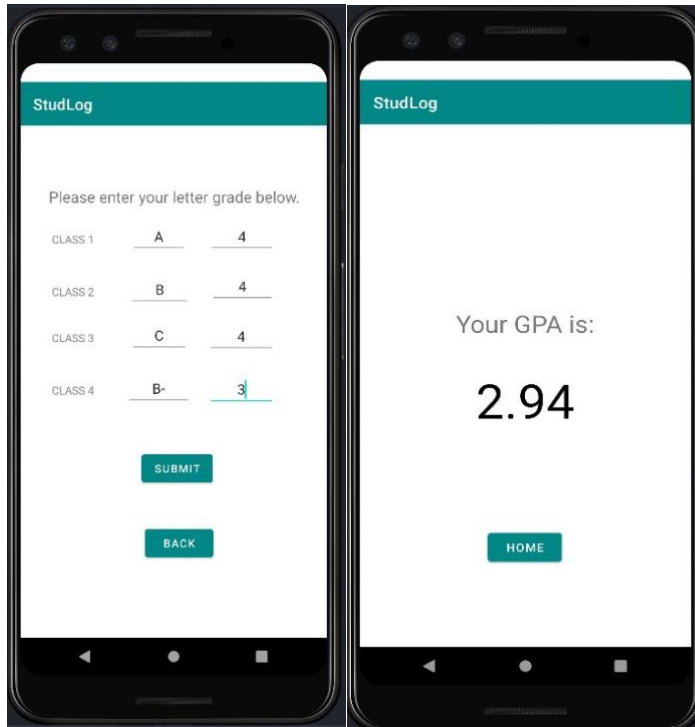
## 4.3 System Feature 3

4.3.1 The third and final feature of the application is the GPA calculator. We will be implementing this calculator to give the students a better understanding of their grades and how it is affecting the overall performance of their semester. We hope that this will give them more control and drive to do well in their courses.



### 4.3.2 Functional Requirements

REQ-1: The students will be required to input their letter grade they are expected to receive in the course along with the total number of credits that the course is worth. Then the application will calculate their GPA with the weight of the letter grade as well as the credits amount. Screen shots of this process are below.



## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

This product will run on 99 percent of Android phones we used a Pixel 3 with the API 30 for this specific reason. We did not want to limit the use to the latest technology to gate keep the product also this product does not require a modern system to operate.

### 5.2 Safety Requirements

No safety standards to meet. The application is a closed system that requires no outside libraries, and no database for storing data.

### 5.3 Security Requirements

No security needed, the application is a single use and forget application. Once you load the credits into the system it will return total study time and forget user input.

## **5.4 Software Quality Attributes**

The software should be reliable. Using the application should be smooth. The calculation should not take long to compute and display to the user.

## **5.5 Business Rules**

We do not have any Business Rules to currently enforce.

## **5.6 Software Evolution**

For our software process we started out trying to explore various platforms and GUI's that could host our product and looked the way we envisioned. Throughout the semester we settled on JavaFX and its open-source GUI with scene builder. It was a steep learning curve for our members and with the stress of other classes and just freshly learning java. Stacking these two tasks of learning another computer language along with multiple IDE's and core support for the GUI's themselves we ultimately ended up going a different route. This led us to Android Studio and using it to create the current iteration of our application with a java core build for Android users. This worked a lot better for our group in completing our project because one of our team members had experience developing in Android Studio's but with Kotlin instead of Java. So, we had to learn Java and not a completely new IDE. Over the last month and a half of the semester, we pushed forward with development and created the product we have today.

## **5.7 Time and Resource**

We tried to use our time to the best of our ability but was the biggest hindrance while developing this application. Everyone in our team had a full semester with multiple CS classes with all heavy workloads. We scheduled a weekly meeting to discuss progress and roadblocks that we encountered along the way. We tried to condense it to a one-week agile period to keep more flexible than the standard two to four weeks used in the industry. We wanted to maximize our time and tried to clear up roadblocks as fast as possible and felt this made us more flexible in the long run given our 14-week deadline. We used discord for most of our meetings that didn't happen in person at the campus than we also used GitHub to store our files that we shared amongst each other. These products that we used eased the process and gave us greater flexibility in adjusting to the pitfalls we inevitably ran into trying to implement JavaFX. This tighter agile schedule of a week allowed us to quickly pivot to the application you currently see today.

## 5.8 Software Change Management

As discussed above we have various changes that we had to implement with the failure of producing a JavaFX application. We had to quickly pivot to a new platform that we had experience with to limit the learning and research required to make an application that we felt meet our standards. We created a table below to show a basic overview of the flow of our work and at the point that it changed from JavaFX to Android Studio, and became a phone application instead of a desktop application.

Month	Application	Meetings
September	Research / JavaFX	4 (once a week) Saturday's
October	JavaFX	4 (once a week) Thursday's
November	Android Studio	4 (once a week) Thursday's
December	Android Studio	2 (once a week) Saturday/ Wednesday

## 5.9 Future Goals

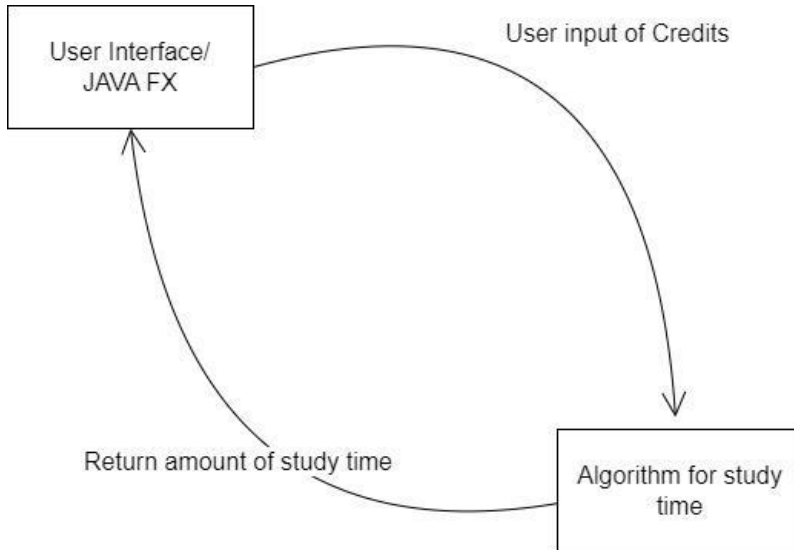
For the future development of this application, we will want to implement a login page along with a database to store information that the user inputs. With added features we will want to implement a notes section that will tie into reminders for the user along with a calendar to hold important dates that deal with the user's classes. For the future data base, we will explore Google Firebase and compare it with other cloud platforms for the flexibility and relatively cheaper cost to maintain the data base. Further down the road when this is implemented, we want to keep a running log of the users grade to keep a running GPA total not just a semester one. This running total will be able to be broken down by semester one if needed to be.

## 6. Other Requirements

There are currently no other requirements not stated in this document.

## Appendix A: Glossary

## Appendix B: Analysis Models



## Appendix C: To Be Determined List

- Functional Requirements 4.1.3 REQ-2 TBD
- Functional Requirements 4.2.3 REQ-2 TBD