Software Requirements Specification

For

Time Estimation App

Prepared by Audrey Eley

Florida Institute of Technology, Senior Design 2024-2025

September, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The Time Estimation App aims to help users, particularly students, accurately estimate the time required to complete academic tasks. Accurate time estimates for completing tasks reduce the stress and inefficiency associated with poor time management, a common issue among students who miscalculate how long a task will take, such as completing homework or studying for an exam. Existing time management tools are neither convenient to use nor sufficiently personalized to the user's specific pacing. This application will address these issues by providing a simple tool for students to plan their coursework schedule while taking their individual working pace into account. This will allow students to plan more effectively, which may help students avoid high-stress situations and achieve higher grades.

1.2 Document Conventions

The term “shall” will be used in requirement descriptions to denote that the system must have the functionality described thereafter.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is composed of three parties. The first of these is the development team for this project. The second party is the client/project advisor for the Time Estimation App. The final party is the faculty overseeing the CS/SE senior design program.

The remainder of this document describes the specifications and requirements of the Time Estimation App. Information most pertinent to developers and client/project advisor is within sections 3-5. Each section is equally pertinent to the faculty overseeing the project.

1.4 Product Scope

This app assists students in tracking their study hours, along with their predictions for how long it will take them to complete a given task. The discrepancies between the predicted and actual time taken to complete academic tasks are tracked and later displayed to the user, with the intention that this information will help the student improve their time management. This is similar to a service provided by Florida Tech’s Hub but will offer additional features.

1.5 References

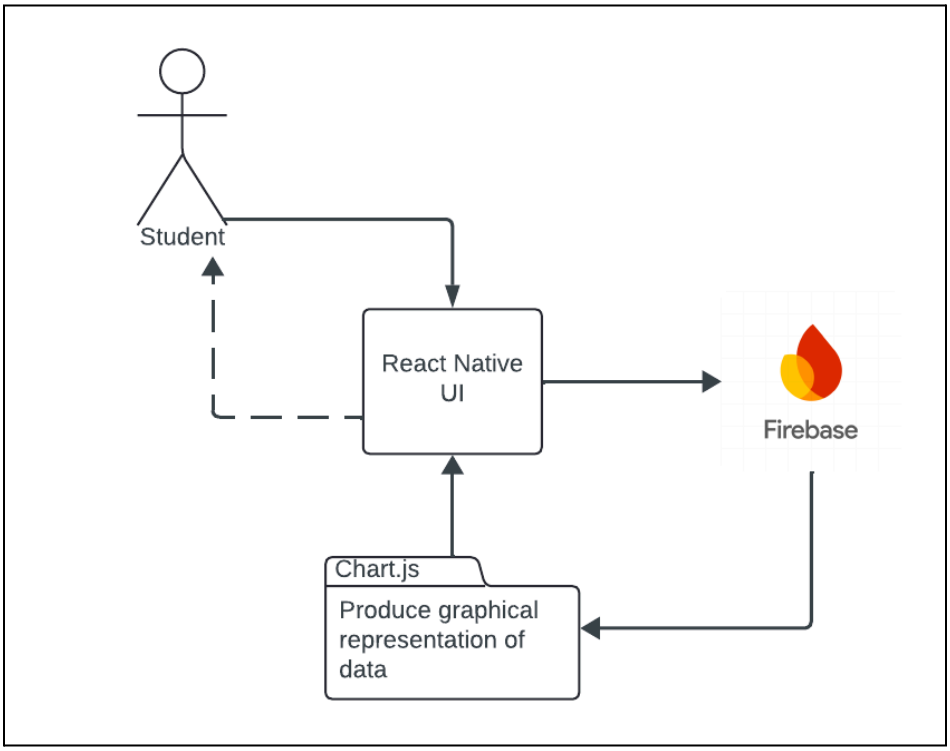
[1] “Introduction · React Native,” *reactnative.dev*, Aug. 15, 2024. Available: https://reactnative.dev/docs/getting-started. [Accessed: Sep. 25, 2024]

[2] “Documentation | Firebase,” *Firebase*, 2019. Available: https://firebase.google.com/docs. [Accessed: Sep. 25, 2024]
[3] “Chart.js Documentation,” *www.chartjs.org*, Aug. 26, 2024. Available: https://www.chartjs.org/docs/latest/. [Accessed: Sep. 25, 2024]
[4] “Firebase Authentication | Firebase,” *Firebase*, 2019. Available: https://firebase.google.com/docs/auth. [Accessed: Sep. 25, 2024]
[5] “Firebase Realtime Database,” *Firebase*, 2019. Available: https://firebase.google.com/docs/database. [Accessed: Sep. 25, 2024]

2. Overall Description

2.1 Product Perspective

The Time Estimation App exists in the context that it is a project for Florida Tech’s Spring 2025 Senior Design Showcase. It is a self-contained program that will improve upon the concepts that are the basis of an aspect of Florida Tech’s Hub called “Study Hours”, which times a user as they work and records the time tracked.



System Diagram of Time Estimation App

2.2 Product Functions

- Time Tracking
- Record Time Estimations
- Compare Estimated and Actual Time Spent on Tasks
- Customizable Interface
- Display Graphical Analytics

2.3 User Classes and Characteristics

This program is intended to be used by only one user type: college students. College students will use all the features of this application. Each student user will only have the privilege to view their own data.

2.4 Operating Environment

The Time Estimation App will operate on Windows, iOS Android. It will use the UI framework, React Native [1]. Data will be stored and user's will be authenticated using Firebase, and data analytics graphics will be created using the Chart.js API [1, 2].

2.5 Design and Implementation Constraints

The primary constraint in this project is the limited experience of the development team with creating web application. The other main constraint is that the program must be tested on Windows, iOS and Android operating systems (or a virtual environment that simulates these).

2.6 User Documentation

Documentation for user reference will be provided on the project website.

2.7 Assumptions and Dependencies

As referenced earlier, this program is dependent on the React Native framework, Firebase, and Chart.js. It is assumed that Firebase is available and scalable and that React Native can support the program's features for Windows, iOS and Android.

3. External Interface Requirements

3.1 User Interfaces

3.1.1

ID: UIR1
Title: Registration Screen
Description: The app shall have a registration screen where users can create an account using a valid email address and a password.
Motivation: Create an account that the user can log into to access app features.
Dependencies: Requirement UAR1.

3.1.2

ID: UIR2
Title: Login Screen
Description: The app shall have a screen where the user can log into an existing account using the associated email address and password.
Motivation: Log into an account so the user can access app features.
Dependencies: Requirements UIR1, SWIR1 and UAR2.

3.1.3

ID: UIR3
Title: Task entry screen
Description: The user interface shall have a form where users can enter the task's title, associated course, the task type, and their time estimation.
Motivation: Record detailed information about academic work for future reference.
Dependencies: None.

3.1.4

ID: UIR4
Title: Task Tracking Screen
Description: The user interface shall have a screen that displays a timer related to a task.
Motivation: Know how long the task is taking to complete.
Dependencies: Requirement UIR3.

3.1.5

ID: UIR5
Title: Graphical Analytics Report Screen

Description: The user interface shall have a screen that displays analytics graphics, including time taken vs. time estimation, and trends related to time spent on academic tasks.
Motivation: View accuracy of predictions, and academic trends.
Dependencies: Requirements SWIR1, SWIR3, CIR1, RTER1, RTER2, and TTCR2.

3.1.6

ID: UIR6
Title: Tasks Overview Screen
Description: The user interface shall have a screen that displays the users tasks.
Motivation: View schedule.
Dependencies: Requirements RTER1, RTER2, and CIR1.

3.2 Hardware Interfaces

3.2.1

ID: HWIR1
Title: Non-local Data Storage
Description: The app shall not store data on the device.
Motivation: Save the device's storage space.
Dependencies: None.

3.3 Software Interfaces

3.3.1

ID: SWIR1
Title: Firebase Realtime Database Storage
Description: The app shall store data using Firebase Realtime Database.
Motivation: Data can be accessed on multiple devices and will not be lost if device fails.
Dependencies: None.

3.3.2

ID: SWIR2
Title: Firebase Authentication
Description: The app shall store user registration credentials and verify user login credentials using Firebase Authentication before allowing the user to access app features.
Motivation: Have a secure account.
Dependencies: Requirement SWIR1.

3.3.3

ID: SWIR3
Title: Chart.js Analytical Graphics
Description: The app shall use Chart.js to produce graphics of analytics based on data collected from user.
Motivation: Visualize data.
Dependencies: Requirement SWIR1.

3.3.4

ID: SWIR4
Title: React Native Framework
Description: The app shall use the React Native framework to build an iOS, Android and Windows user interface.
Motivation: Have a user interface that is compatible with iOS, Android and Windows.
Dependencies: None.

3.4 Communications Interfaces

3.4.1

ID: CIR1
Title: Frontend and Backend Communication
Description: The app shall communicate between frontend and backend using HTTP/HTTPS.
Motivation: User input and time-tracking data are communicated to the backend, and the backend output can be viewed on the frontend.
Dependencies: Requirements SWIR1 and SWIR4.

4. System Features

4.1 User Authentication

Description: The app shall require a user to register for an account with a valid email address and password, and log in to this account before allowing the user to access other features.

Functional Requirements:

4.1.1

ID: UAR1
Title: User Registration
Description: The app shall allow users to register an account using a valid email address and password.
Motivation: Keep information from app usage linked to an email address.

Dependencies: Requirement UIR3, CIR1, and SWIR2.

4.1.2

ID: UAR2
Title: Login Requirement
Description: The app shall require the user to enter an email address and password corresponding to a registered account before allowing the user access to app features or data.
Motivation: Keep user information secure.
Dependencies: Requirement UIR3, CIR1, and SWIR2.

4.1.3

ID: UAR3
Title: Logout
Description: The app shall allow users to logout of the account that they are signed into, and shall require the user to login and be re-authenticated before allowing them to access app features and data.
Motivation: Be able to mitigate the risk of unathorized account access.
Dependencies: None.

4.2 Record Time Estimates

Description: The program shall allow users to initialize tasks with an estimate of the time it will take to complete.

Functional Requirements:

4.2.1

ID: RTER1
Title: Input Task Information
Description: The app shall allow the user to input task titles, associated course, task types, and time estimates.
Motivation: Identifying information pertaining to the task, and estimates of time to completion can be recorded.
Dependencies: Requirement UIR3.

4.2.2

ID: RTER2
Title: Store Task Information
Description: The app shall store the task information entered by the user.
Motivation: Data that is stored can be viewed later on by the user and used for analytics
Dependencies: Requirement RTER1.

4.3 Track Time to Task Completion

Description: The program shall allow users to track the time it takes them to complete a task.

Functional Requirements:

4.3.1

ID: TTTCR1
Title: Timer Display
Description: The app shall display a timer when a user selects to track their time on a task.
Motivation: Able to view the amount of time that has been spent on a task.
Dependencies: Requirement UIR4.

4.3.2

ID: TTTCR2
Title: Timer Data Storage
Description: The app shall record the total amount of time spent when the user selects to end the timer and store that record using Firebase Realtime Database.
Motivation: Able to refer back to time spent on tasks, can be compared to estimated time, and can use the data for analytics.
Dependencies: Requirement SWIR1.

4.3.3

ID: TTTCR3
Title: Timer Record Comparison to Time Estimate
Description: The app shall display the difference between the estimated time and actual time taken for a given task.
Motivation: Can evaluate the accuracy of the time estimate.
Dependencies: Requirement TTTCR2.

4.4 Customizable Interface

Description: The app shall allow users to organize tasks based on different categories.

Functional Requirements:

4.4.1

ID: CuIR1
Title: Task Organization
Description: The app shall allow users to group tasks based on either the associated course or task type.
Motivation: Allows overview of tasks in a manner that is preferred by the user.

Dependencies: Requirement SWIR1.

4.5 Display Analytics

Description: The app shall display reports of analytics based on user data.

Functional Requirements:

4.5.1

ID: DAR1
Title: Time Comparison Visualization
Description: The app shall display a graphical comparison of predicted versus actual time taken to complete tasks.
Motivation: Determine the accuracy of predictions over time.
Dependencies: Requirements UIR5, SWIR1, SWIR3, CIR1, and RTER2.

4.5.2

ID: DAR2
Title: Progress Report Visualization
Description: The app shall display a summary of completed tasks from a time-frame entered by the user.
Motivation: View a graphical representation of productivity.
Dependencies: Requirements UIR5, CIR1, SWIR3, and RTER2.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1

ID: PR1
Title: Time to Produce Analytics
Description: The app should load and display analytical graphics within 10 seconds of request for up to 1,000 tasks.
Motivation: View data graphics without having to wait a long time.
Dependencies: Requirements DAR1 and DAR2.

5.2 Safety Requirements

5.2.1

ID: SR1
Title: User Privileges
Description: The app shall only allow a user to access their own data.
Motivation: Have privacy.
Dependencies: Requirement SWIR1.

5.3 Security Requirements

5.3.1

ID: SeR1
Title: Secure Communication
Description: The app shall use secure communication channels for communication between frontend and backend.
Motivation: Have data be handled securely.
Dependencies: Requirement CIR1.

5.3.2

ID: SeR2
Title: Access Upon Login
Description: The app shall not allow access to app features or data until the user inputs credentials associated with an existing account, and these credentials are authenticated.
Motivation: Have a secure, personal account.
Dependencies: UAR1 and UAR2 .

5.4 Software Quality Attributes

5.4.1

ID: SQR1
Title: Reliability
Description: The app should have at least 95% uptime.
Motivation: Have access to the app most of the time.
Dependencies: None.

5.5 Business Rules

5.4.1

ID: BR1
Title: Free Access
Description: The app shall allow free access to all features.
Motivation: Have an academic tool that does not cost money.
Dependencies: None.

Appendix A: Glossary

- Analytics: Data interpretation performed by a computer, which in this case is used to produce a graphical representation.
- Chart.js: A JavaScript library for charting data [3].
- Firebase: An application development platform made by Google [2].
- Firebase Authentication: A backend service offered by Firebase for authenticating application user credentials [4].
- Firebase Realtime Database: A backend service offered by Firebase which stores and updates data realtime within the cloud [5].
- React Native: A JavaScript-based user interface framework for applications [1].
- Task: A self-contained academic activity to be assigned a time estimate.
- Task type: Category of an academic task, such as “Write Essay”, “Study for Test”, or “Read Textbook Chapter”.
- Time estimation: The amount of time the user predicts it will take them to complete a given task.
- Progress report: A graphical summary of the tasks completed by a user over a specified time-frame.