

Time Estimation App

Software Design Document

Audrey Eley, Artur Quarra
09/29/2024

Table of Contents

1. Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Overview	3
1.4 Reference Material	3
1.5 Definitions and Acronyms	3
2. System Overview	4
3. System Architecture	4
3.1 Architectural Design	4
3.2 Decomposition Description	5
3.3 Design Rationale	5
4. Data Design	6
4.1 Data Description	6
4.2 Data Dictionary	6
5. Component Design	6
6. Human Interface Design	7
6.1 Overview of User Interface	7
6.2 Screen Images	8
6.3 Screen Objects and Actions	9
7. Requirements Matrix	10

1. Introduction

1.1 Purpose

This document is prepared to define the architecture of the Time Estimation App for Florida Tech's Spring 2025 Senior Design Showcase. This document is intended as a guide for the development team during the implementation phase. It is also written as a formal statement of the app's design plan for the approval of the project client/advisor, Dr. David Luginbuhl, and the faculty supervisor of the Computer Science Senior Design Program, Dr. Philip Chan.

1.2 Scope

The app described here will assist college students by recording data related to their studies. Specifically, it will allow the user to view the difference between how long the user expects a task to take, versus the task's actual length. This is meant to help students better manage their time by assisting them in more accurately estimating how long it will take them to complete a task.

1.3 Overview

This document will continue in section 2 by providing a brief overview of the capabilities of the system. Then, section 3 will cover the structure of the system. Section 4 will describe the data design. Section 5 will detail the component design. Section 6 will outline the specifications of the user interface, and finally, section 7 contains the requirements matrix.

1.4 Reference Material

- [1] "Chart.js Documentation," *www.chartjs.org*, Aug. 26, 2024. Available: <https://www.chartjs.org/docs/latest/>. [Accessed: Sep. 25, 2024]
- [2] Firebase, "Documentation | Firebase," *Firebase*, 2019. Available: <https://firebase.google.com/docs>. [Accessed: Sep. 25, 2024]
- [3] Google, "Firebase Authentication | Firebase," *Firebase*, 2019. Available: <https://firebase.google.com/docs/auth>. [Accessed: Sep. 25, 2024]
- [4] Google, "Firebase Realtime Database," *Firebase*, 2019. Available: <https://firebase.google.com/docs/database>. [Accessed: Sep. 25, 2024]
- [5] "Introduction · React Native," *reactnative.dev*, Aug. 15, 2024. Available: <https://reactnative.dev/docs/getting-started>. [Accessed: Sep. 25, 2024]
- [6] Eley, Audrey. "Software Requirements Specification for Time Estimation App." Florida Institute of Technology, Senior Design 2024-2025, September 2024.

1.5 Definitions and Acronyms

- 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 [1].
- Firebase: An application development platform made by Google [2].

- Firebase Authentication: A backend service offered by Firebase for authenticating application user credentials [3].
- Firebase Realtime Database: A backend service offered by Firebase which stores and updates data realtime within the cloud [4].
- React Native: A JavaScript-based user interface framework for applications [5].
- 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”.

2. System Overview

With this app the student will be able to enter their academic tasks, along with an estimate of how long it will take to complete them. The user may then select to begin an inputted task, and the time to completion will be tracked. The difference between the estimated and actual time taken to complete the task are compared and displayed to the user. The user can see an overview of their tasks, or analytics graphics generated from their data.

3. System Architecture

3.1 Architectural Design

The architecture of this program will have 3 main components:

- Frontend: The application interface. Built using the React Native framework [5].
- Backend: The application’s associated database. An instance of the Firebase Realtime Database [4].
- Analytics: The method through which data will be analyzed, and data graphics will be created. The Chart.js library will facilitate this [1].

The frontend and backend will be in communication. Chart.js will access the backend data to generate graphics displayed on the frontend.

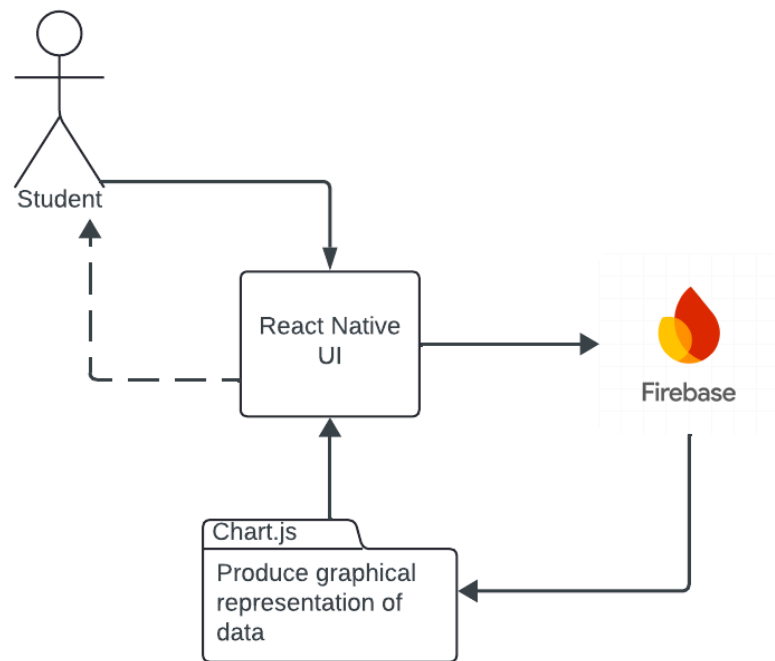


Diagram of Time Estimation App System

3.2 Decomposition Description

The subsystems of the program can be broken down as follows:

- Frontend
 - User Interface
 - Task Management
 - Time Tracking
 - Display Analytics
- Backend
 - Data Storage
 - User Authentication
- Analytics
 - Generate Graphics

3.3 Design Rationale

The architecture described here was chosen because it is a streamlined design that can support multiple platforms, and a realtime database. The React Native framework is easy to use and compatible with the platforms this app will be launched on (iOS, Android, and Windows) [5]. The Firebase Realtime Database provides real-time cloud-based storage that is well suited to holding the data of this system [4].

4. Data Design

4.1 Data Description

The data associated with the app are stored in Firebase Realtime Database. This data is of 2 main types, task data and user data. Task data includes a unique ID, the associated user's unique ID, the course the task is associated with, the type of task, the amount of time the user expects to complete the task in, and the actual time taken to complete the task. The user data exists to differentiate accounts, so the user may only access the information associated with their account. The user data includes the user's unique ID, the user's email address, and the account password. The email address and password will be required for account registration, and subsequent logins. The user credentials will be verified using Firebase Authentication [3].

4.2 Data Dictionary

The data entities in this program are tasks, and users.

- Task
 - Attributes
 - task_id (string): A unique identifier for the task.
 - user_id (string): A unique identifier for the user.
 - course (string): The name of the course the task belongs with.
 - task_type (string): The type of academic task (e.g. study for test, write essay).
 - estimated_time (integer): Time in minutes.
 - actual_time (integer): Time in minutes.
- User
 - Attributes
 - user_id (string): A unique identifier for the user.
 - email (string): The user's email address.
 - password (string): The user's password.

5. Component Design

The components of the system, broken down by functions are:

- Registration Component
 - Description: Allows the user to register by creating an account.
 - Methods: createAccount()
 - Local Data: User data element and attributes (user_id, email, password).
- Sign-in/out Component
 - Description: Allows the user to login to an existing account, and logout of an account they are logged in to.

- Methods: login(), logout()
 - Local Data: User data element and attributes (user_id, email, password).
- Task Management Component
 - Description: Controls creation and maintenance of tasks.
 - Methods: createTask(), updateTask(), deleteTask()
 - Local Data: Task attributes (task_id, user_id, course, task_type, estimated_time, actual_time)
- Task Arrangement Component
 - Description: Allows the user to change the layout of their tasks.
 - Methods: byType(), byCourse()
 - Local Data: Task attributes task_type and course.
- Time Tracking Component
 - Description: Enables the user to track the time taken to complete tasks.
 - Methods: startTimer(), stopTimer(), trackTime()
 - Local Data: Start and end time of timer.
- Analytics Graphics Component
 - Description: Produces analytics graphics based on stored data.
 - Methods: createGraphics(), displayGraphics()
 - Local Data: Tasks and task attributes. Graphics images.

6. Human Interface Design

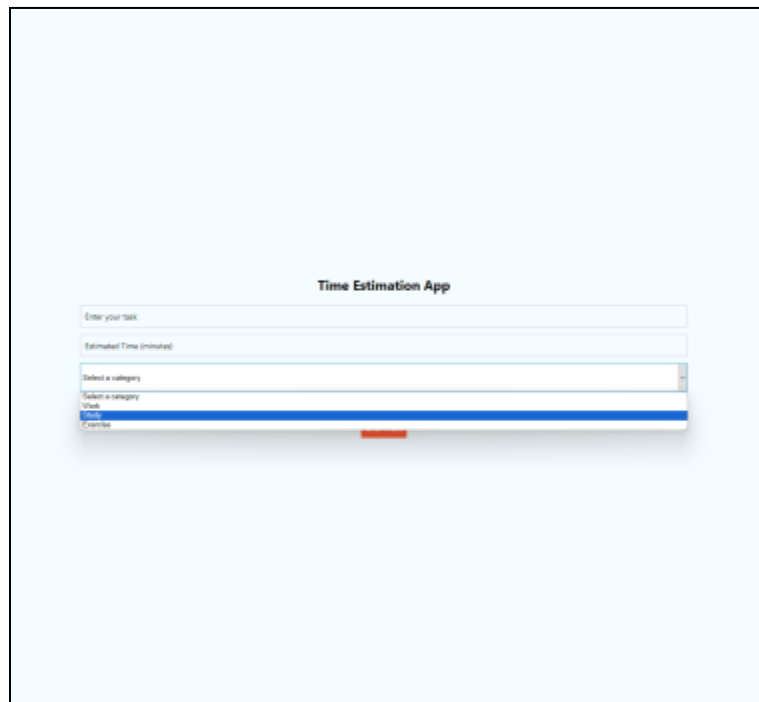
6.1 Overview of User Interface

From a user perspective, the features accessible from the user interface will be:

- Create Tasks
- View Task Dashboard
- Organize Task Dashboard
- Track Tasks via Timer
- View Estimated Versus Actual Time Spent in Tasks
- View Analytics Related to Tasks

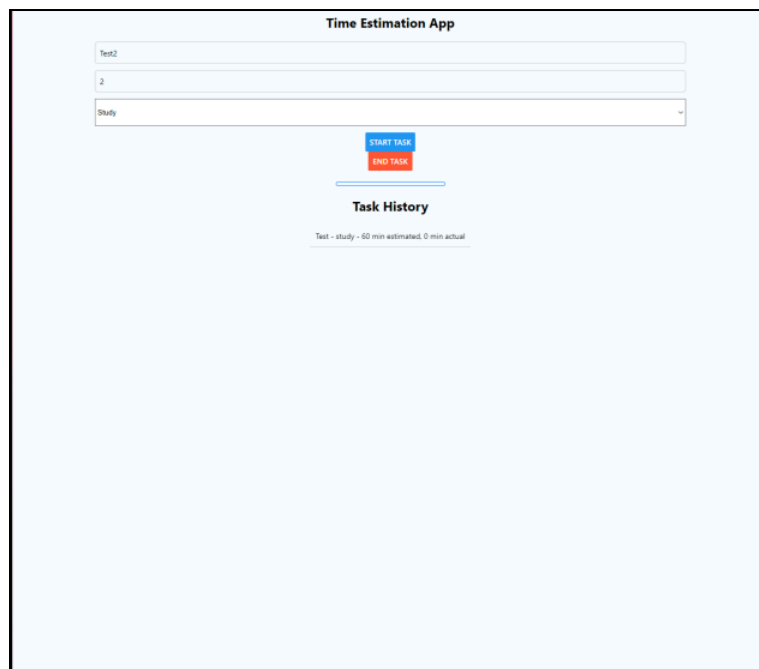
6.2 Screen Images

Provided here are images of the UI's prototype version.



The Task Entry Screen is a light blue rectangular area containing a form titled "Time Estimation App". The form has five input fields: "Enter your task", "Estimated Time (minutes)", "Select a category", "Select a category", and "Start". Below the "Start" field is a red button labeled "Start".

Task Entry Screen



The Dashboard Screen is a light blue rectangular area containing a form titled "Time Estimation App". The form has three input fields: "Text2", "2", and "Study". Below the "Study" field is a red button labeled "START TASK" and a blue button labeled "END TASK". Below the buttons is a section titled "Task History" with a single entry: "Text - study - 60 min estimated, 0 min actual".

Dashboard Screen

The screenshot displays the 'Time Estimation App' interface. At the top, there are three input fields: 'Test2', '2', and 'Study'. Below these fields are two buttons: 'START TASK' (blue) and 'END TASK' (red). A progress bar is shown below the buttons. The main text area displays a green message: 'You finished 2 minutes earlier than expected!' followed by 'Task started at 08:44:06 and ended at 08:44:36.' Below this is a section titled 'Task History' which lists two tasks: 'Test - study - 60 min estimated, 0 min actual' and 'Test2 - study - 2 min estimated, 0 min actual'.

Timer Screen

6.3 Screen Objects and Actions

The interactive elements of the UI are as follows:

- The user can select to input a task.
 - A form will appear for the user to input task details.
 - The user can select to “start” the task.
 - A timer is displayed on the screen.
 - The user is able to select to “end” the task.
 - The expected versus actual times are displayed.
- The user can choose to view their task dashboard.
 - The user can select to group the tasks on their dashboard by course or by task type.
- The user can select to view their analytics.
 - The user can select to view a chart of their estimated versus actual task times.
 - The user can select to view a report of their recently completed tasks.

7. Requirements Matrix

The requirements included here are defined in the SRS document for this project [6].

Requirement(s) ID	Description	Component
UAR1-3	Users must register an account with a username and password. Users must log in and be authenticated before being allowed access to features and data. Users may logout of an account they are logged in to.	User Authentication
RTER1, RTER2, CuIR1	Users can enter and edit tasks and their attributes. Users can organize their tasks on their dashboard.	Task Management
TTTCR1-3	Users can track the duration of their tasks, by choosing to start and end them. The task duration is stored, and compared to the user's initial estimate.	Time Tracking
DAR1, DAR2	Generates graphics for user viewing based on stored data. This includes actual versus estimated time, and recent task progress.	Analytics
SWIR1, SWIR2	Accounts and associated data are stored for authentication. Task data is stored, and is associated with a specific user, so only that user may access it.	Data Storage
UIR1-6	The interface through which users can interact with the system. Contains screens for login, task entry, task tracking, analytics, and task overview dashboard	User Interface