STUDY BUDDY
SYSTEM DESIGN

By: Ahmad Fakhry, Elhan Syed, Anthony Chu, Gagandeep Bhangal, Shaun Aranha,

Asad Ali, and Osama Hafez

TABLE OF CONTENTS

SOFTWARE ARCHITECTURE DIAGRAM

client-side/frontend   server-side/backend   database

FRONT-END: React

- The user interface. The user interacts with this to manipulate their data via the backend/server
- Users are able to login, start a timer, add a task to a list
- The React client makes requests via the **Axios** package to this server to perform all operations the user wants.

BACK-END: Express Server (Node)

- The Application Programming Interface (API)
- Processes the requests from the user, parses them for errors, makes sure they're authenticated
- Sensitive content is encrypted by the **Bcrypt** Package
- Uses **Mongoose** to manipulate the database

DATABASE: MongoDB

- Holds all data processed by the backend from the user
- This is where all CRUD operations are done

Routes/Task

• Create a task • Delete a task • Update a task • Update a task • Output all tasks	• Task Item

Routes/User

• Create a new user account • Login to an existing user account	

Models/Task

• Creates the task with its respective properties in mongoDB	

Models/User

• Creates the user in the database • Encrypts the user's password before adding to the database	• User Item

client/src/container/app

• Routes url to particular components	• client/src/containers/authpage • client/src/components/dashboard

client/components/Timer	App
- Starts the timer when the start button is clicked - Stops the timer when the stop button is clicked - Resets the timer back to the start time when the reset button is clicked - Triggers a sound to notify the user when the time is up or when the time has started	client/src/containers/app

Routes/List	
- Create a list - Get a list - Delete a list - Update a list title - Add task(s) to a list	

Models/List	
Sets the schema to define what a list is with its respective properties in MongoDB	- Routes/List - Routes/Task

client/src/components/login/login	
- Creates a login form for the user - Posts the data inputted by the user to the back-end - Processes the response from the back-end	client/src/containers/app

client/src/components/register/register

- Creates a registration page for the user to create an account to use the application

- client/src/containers/app

client/src/components/SessionList/SessionList

- Display List view of tasks
- Allow user to manipulate progress of tasks

- client/src/containers/app

client/src/components/Task/Task

- Request an Update to a Task
- Allow the user to interact with the task

- client/src/containers/app

HOC/Auth
client/src/components/login, client/src/containers/authpage

- Checks the clients cookies for the JWT token for authorization

- client/src/components/login
- client/src/containers/authpage

client/src/Containers/AuthPage/Authpage
client/src/Containers/Login

- Displays a login form and verifies user input

- client/src/Containers/App/App

client/src/Components/LogoutButton
client/src/Components/Navbar/Navbar

• Allows users to logout from the application	• client/src/Components/Navbar • client/src/Components/Dashboard

client/src/components/profilepage
client/src/components/navbar

• Displays a users information and allows them to make changes	• client/src/components/navbar,

client/src/components/dashboard
client/src/components/timer, client/src/components/navbar, client/src/components/sessionlist, client/src/components/clearbutton, client/src/components/list

• The main app for the users, displays the timer, list of tasks, and navigation bar	• client/src/components/timer • client/src/components/navbar • client/src/components/sessionlist • client/src/components/clearbutton • client/src/components/list

src/group/server.js
src/utils/messages, src/utils/users

• Routes incoming user traffic for the sockets portion of the application, • Accepts and disconnects users	• client/GroupStudy/GroupStudy/GroupStudy.js

client/src/GroupStudy/GroupStudy/GroupStudy
client/src/GroupStudy/GroupTimer/Timer, client/src/GroupStudy/Chat/Chat

• Lays out the timer and chatbox for participants in a session	• client/src/GroupStudy/GroupTimer/Timer, • client/src/GroupStudy/Chat/Chat

	Timer.js
client/src/Components/Notifications.js	
• Notifies the user when the timer is done	

	client/src/GroupStudy/GroupStudy/GroupStudy
client/src/GroupStudy/GroupTimer/Timer	
• Creates a timer that is usable by multiple participants	

	client/src/GroupStudy/GroupStudy/GroupStudy
client/src/GroupStudy/Chat/Chat	
• Creates a usable chat for participants in a group study session	

	client/components/Dashboard
client/components/CalendarPage	
• Creates a calendar page for the user	

client/containers/gateway
• Allows the user to choose between solo studying and group studying