Team: Ethernet

Architecture Design:

For Life Habitat, we decided to design the application using a layered architecture. This design allows for easier scalability, and the software is designed for use by individual users, so communication between users is less important. Keeping the different layers of the application separate also lets us modify or change things in one layer without breaking something in another.

Architecture Design Diagram:

| User Interface Layer:    Contains the UI of the application. Allows users to interact with the app to create tasks, customize avatar, etc. |
| --- |
| Application Layer:    Contains the logic for adding and deleting tasks  Contains the logic for setting repeat tasks  Contains the logic for gaining and spending points  Contains the logic for user feedback (eg. Programmed responses/AI responses)  Contains the logic for Avatar customization |
| Data Storage Layer:  Holds all user data. Data will be encrypted to ensure privacy.  Data will only be used to repopulate application layer with data on a successful login. |

Design Description:

Life Habitat will have 3 separate layers: The UI layer, The Application Layer, and the Data Storage layer. Each Layer will control different parts of our program.

The UI layer will contain the user interface for our app. This will be where Users will navigate to what they wish to access. React will be used to handle the UI. Here users will be able to navigate to different pages, input data for tasks, and select whether tasks were completed or not. This layer will also house the user’s avatar and shall allow users to select customization objects they have unlocked.

The application layer will be the main part of our program. Here we will have the logic and code for the task list, timers, calendars, and calculations for points from completed tasks. For example, when a user selects they would like to add a task, they will be given a field to enter in what they want on the UI layer, and that data will be taken into the application layer to process and add the task to their list/calendar. Another example will be when a user selects a task as completed or failed, when this happens the app will react accordingly, congratulating the user on a job well done, or giving encouragement when things don’t go as planned.

Finally, the data storage layer will be where all the user data will be stored for security reasons and to allow users to access their data on any machine after logging in. It will store user tasks, information, the user’s avatar, points, and other information. When a user successfully logs in the data will be read and pushed back up into the application layer and UI layer for the user to interact with.