**Software Requirements and Design Document**

**For**

**Group <7>**

Version 1.0

**Authors**:

Jacqueline Vermette

Meghan Cox

Madelyn Yarber

# **1.** **Overview (5 points)**

*Give a general overview of the system in 1-2 paragraphs (similar to the one in the project proposal).*

Currently, our system runs on a local host and when it runs it shows our habit tracker. When you first open the habit tracker you are met with a list of your daily, weekly, and monthly habits. You will also see motivational quotes and notes. If you look to the left side of the screen, you can see a navigation bar where there is a dashboard of tasks, calendar view, data, and edit habits tab. There is an API currently being worked on in order to store all the data from users when they use the habit tracker.

# **2.** **Functional Requirements (10 points)**

*List the* ***functional requirements*** *in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

A user will be made up of an ID and be able to login /register: High

The user should be able to write notes: Medium

It should be able to pull in database information from the API: High

User should be able to navigate to dashboard and change to the different pages: High

Have habits show up on dashboard: low

Be able to add to the calendar: low

Pie chart to demonstrate completion of tasks/habits: low

Edit and add custom habits to the daily, weekly and monthly habits: low

# **3.** **Non-functional Requirements (10 points)**

*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

Encrypting user data. We will do this by hashing the data. It generates a jwt token for authentication.

The software will run smoothly with no input delay.

The software will save all user input.

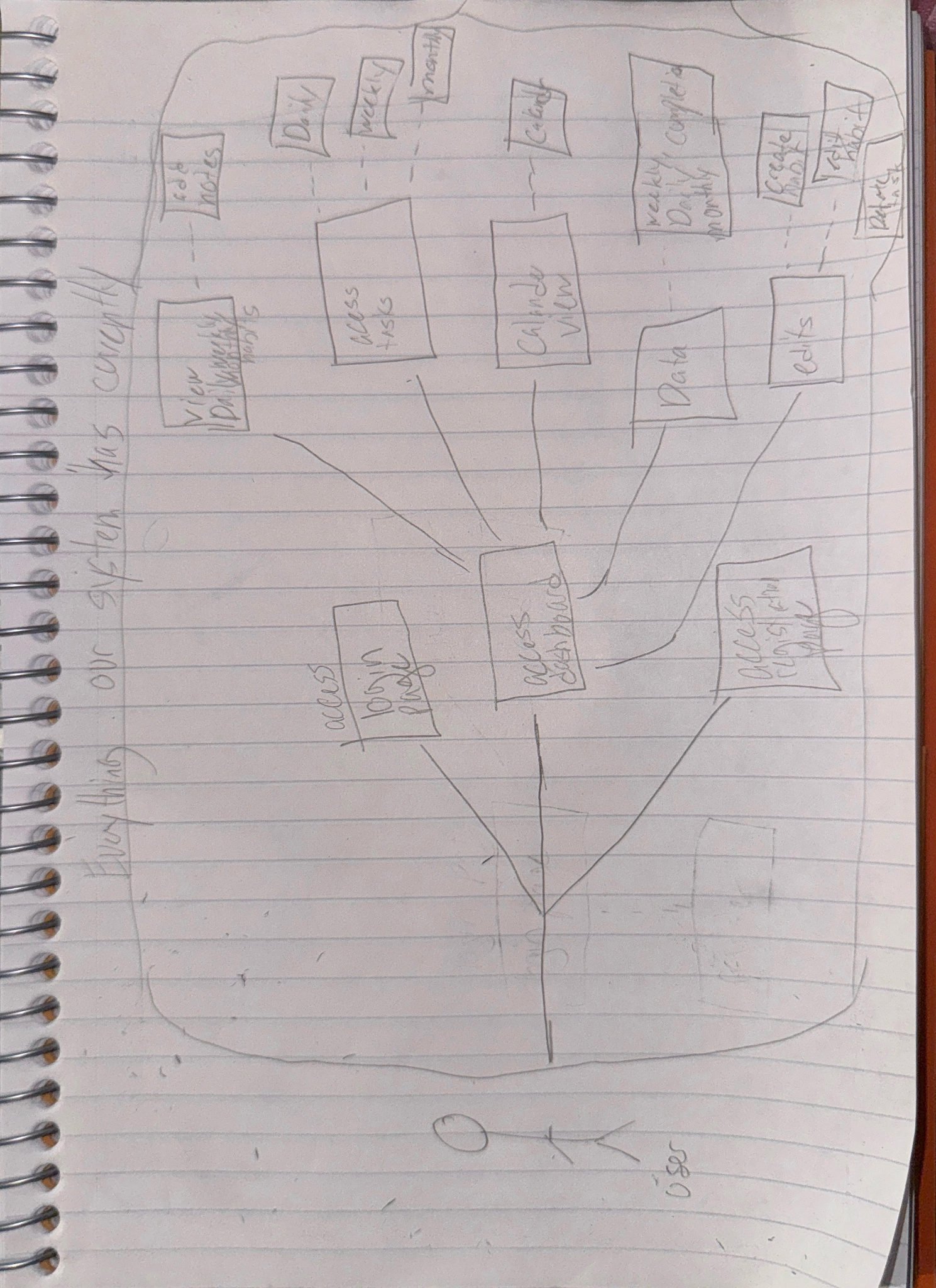
The software will not crash or lose data.

The software will be safe and secure data.

# **4.** **Use Case Diagram (10 points)**

*This section presents the* ***use case diagram*** *and the* ***textual descriptions*** *of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*

***Textual descriptions of use cases****: For the first increment, the textual descriptions for the use cases are not required. However, the textual descriptions for all use cases discovered for your system are required for the second and third iterations.*

**

# **5.** **Class Diagram and/or Sequence Diagrams (15 points)**

*This section presents a high-level overview of the anticipated system architecture using a* ***class******diagram*** *and/or* ***sequence diagrams****.*

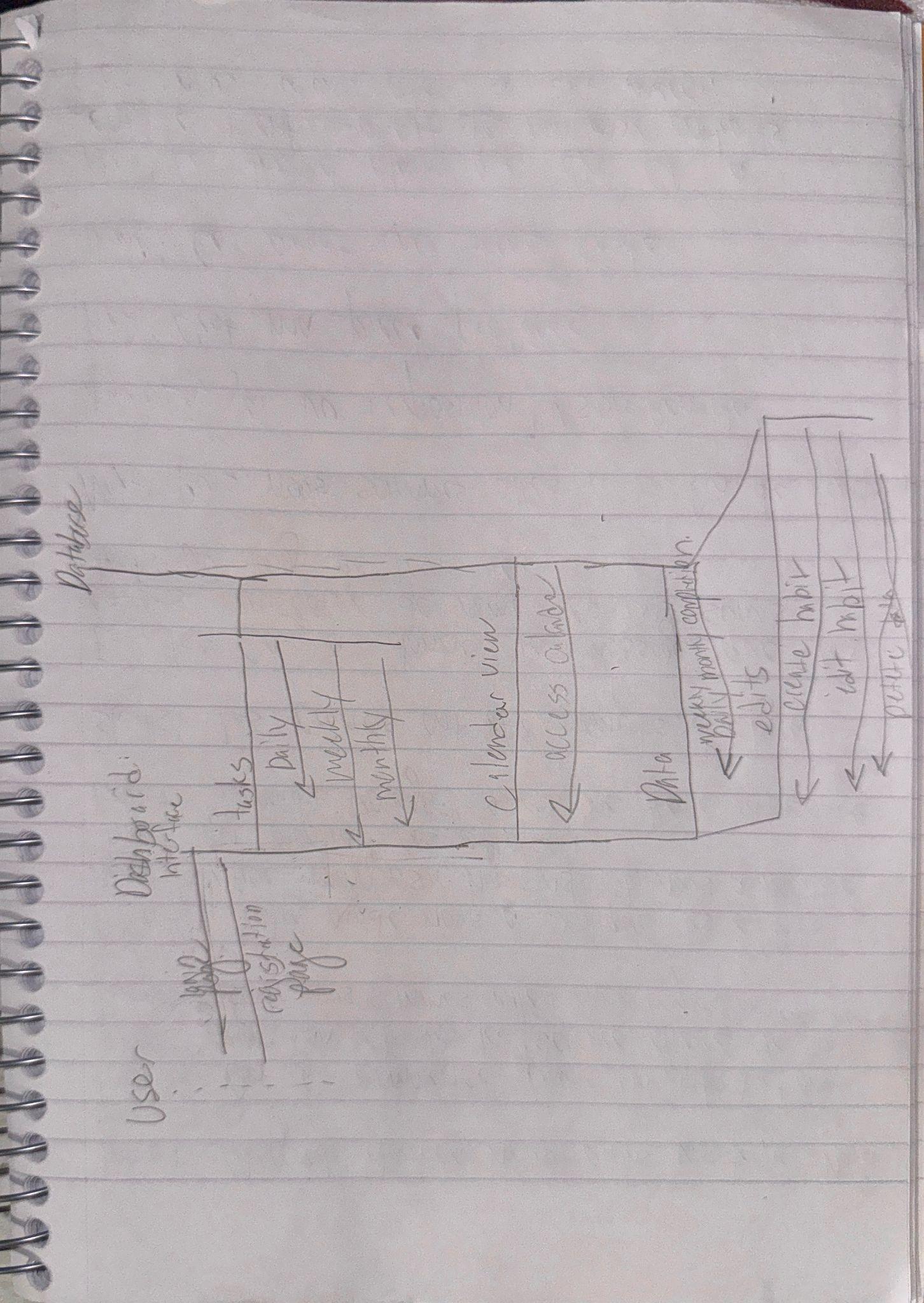
*If the main* ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram******of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

*o*

*If the main* ***paradigm*** *in your system is* ***not Object Oriented*** *(i.e., you* ***do not*** *have classes**or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

***Class Diagrams*** *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

*A* ***Sequence Diagram*** *simply depicts* ***interaction******between objects*** *(or* ***functions -*** *in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

**

# **6.** **Operating Environment (5 points)**

*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.*

It is currently just a webpage however we are hoping for it to be cross platform. Currently our webpage is run on Windows 11 version 24h2. We are using an API database.

# **7.** **Assumptions and Dependencies (5 points)**

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*

I believe that some assumed factors that could affect the above requirements would be if the database does not link properly. Also, I believe that if using CSS instead of C# causes too many problems we may need to change everything to C#. In addition to this, the encryption method may need to be altered if it does not properly encrypt the user's input for login/registration.