**Software Requirements and Design Document**

**For**

**Group <NoteWorthy>**

Version 2.0

**Authors**:

Lexie W

Humbert T

Daimeun P

Tyler G

David M

Orlando K

**1.**    **Overview (5 points)**

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

*Our system is a web application that streamlines the process of taking notes and generating study material to better help students in their classes. It will facilitate an environment that allows students to sign up and login to our webapp and both take notes and  save materials and information from their classes. We also will allow the students to share notes with other students and generate studying material to better help in the studying process.*

**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.*

*User Accounts*

*1.     Sign up for a user account = Done*

*2.     Log in to existing user account = Done*

*3.     Log out of user account  = Done*

*4.     Allows users to create and save notes = High*

*5.     Allows users to create folders to signify classes = Medium*

*6.     Create and customize profiles (Profile Picture, Bio, Name etc.) = High*

*7.     Recommend tab of users in similar majors and classes = Low*

*8.     Users can list class as private or public = Medium*

*9.     User can set password for private classes = Low*

*10. Make the following pages: “Meet the Team”, “Mission + Vision”, “Contact Us”, “Logout”, “Home”, “Classes”, “Settings = High*

*Proprietary Note taking interface*

*1.     Takes notes from user = High*

*2.     Generates flashcards = Low*

*3.     Definition mode that automatically formats into a term and definition card = Low*

*4.     Suggests additional sources = 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.*

*User Accounts:*

*1.     Protecting User Passwords on the DB using bcrypt = Done*

*2.     Reliable connection to database through “middleware” = Done*

*Proprietary Note taking interface  
1. Send notes associated to user to backend for database storage = High*

*2. Make Note taking page more user friendly = High*

**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.*

A close up of text on a white background

Description automatically generated

**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.***

*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.*

**A close up of text on a white background

Description automatically generated**

**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.*

*Our web app is accessible on all platforms with a network connection. It’ll work on any operating system that is compatible with the latest version of REACT.*

**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.*

*We are dependent on the fact that the frameworks we are using will continue to be supported and that the user has a internet connections*