Concept of Operations

Esin Sari

Project 3: Private Notes

COP4331, Fall, 2020

Contents of this Document

The Current System

Private Notes is a note-taking web application which enables users to store ideas, inspirations, tasks, daily lists, reminders, or any type of information. It provides ability to create, organize, and sort notes while ensuring distraction-free not-taking environment with its clear and easy access interface.

There are similar note-taking management systems found in industry such as Evernote, Google Keep, or Standard Notes. Rather than some systems, Private Notes app is accessible by any user such as Mac users, Google power users, designers, students, and people from any age. It user-friendly, interactive, and easily understandable to offer efficient note-taking experience. Moreover, unlike the other systems, this product does not require a subscription fee to access all its features, which makes Private Notes apps more appealing.

The Proposed System: Motivation

Comparing with other note-taking web application, Private-Notes App will have less features but will focus on only note-taking related features which enables product to serve distraction-free not-taking experience to its users. Furthermore, with its clear user interface, Private Note apps offers easier to use and understand features. It is accessible from any place where user have access to an active internet connection. Another great advantage of this application is its free and will not charge a subscription fee rather than many other systems found in industry.

The Proposed System: Users and Modes of Operation

Description on different class of user;

- Private Notes system does not put restrictions on different class of users.
- Users are not classified as low/high level user or free/pay version user.
- All users are able to access features and note management tools.
- Systems does not ask for subscription fee.

Description on modes of operation;

- System requests users to create an account before starting to use app.
- Registered users can change their current password by simply clicking on change my password button.
- Previously created notes are visible in users home interface and can be sorted in different aspects; based on the data it is created, based on its category, and alphabetically with note titles.
- Users can create various text note with a title; any newly created note without a title is not accepted.
- Created notes can be labeled based on specific categories such as class notes, recipe, grocery, to-do list, address, etc.
- Created notes can only be deleted, changed, or modified by users themselves.
- Registered users are not allowed to see or engage other users' notes.
- Different class of users are able to take full advantage of product in same level.

The Proposed System: Operational Scenarios

Private Notes systems is designed to organize and search various text notes with easy and straightforward way.

- New users can create an account by clicking on 'create account' button and system will ask for a username and a password.
- Registered users can access their account by simply entering their username and password.
- There will be an authentication process, If user needs to change their password, there is a provided 'change my password' button in homepage and system will ask for current password and new password.
- In case of an incorrect input scenario for both username and password, there will be a generated warning message. More details are given in following parts.
- In the application user interface, all previously created notes are visible, and user can access them by clicking on the name of the notes.
- 'Create new note' button will be used to generate a note and requires entering a title before saving.
- In case of entering already used title or saving notes without specifying title, there will be a generated warning message. This allows system to sort notes without a system crash.

- If user no longer need or desires to modify existing notes, they can take an action by clicking on related options.
- Private Notes system contributes user to organize their notes by providing label option; they can label any notes based on specific categories and those categories will be named by the user. Each category can contain multiple notes and category of notes can be changed any time.
- If user desired to search his/her notes based on date, title, or category, they can select from sorting options in the application. If there is no previous entry, then they will see 'empty folder' message.
- Incorrect inputs and system crash are handled by the system; however, system is not handling loss of internet connections.
- *Input Sanitization* is handled by the framework.

The Proposed System: Operational Features

Must Have:

- Ability to create account and change password
- Visible Notes in application user interface
- Create, delete, and modify notes
- Assigning and changing title
- Selection for Sorting Options
- Restriction on accessing others account

Would Like to Have:

- Supporting system with artificial intelligence like Siri
- Ability to share notes and create social environment between other users
- Ability to store graphs, images, and drawings
- Sending reminder message based on date user preferred

The Proposed System: Analysis

Private Notes App will be developed by using JavaScript with React.js library/web framework and the backend database will be handled within JavaScript using Node.js environment and MySQL will also be used for database. Moreover, I will use Visual Studio Code, which is IDE-like features and open-source, to develop this system product. The reason I chose JavaScript, React.js, and Node.js is these languages are currently in high demand and mostly preferred languages in industry; moreover, it is beneficial for my career to learn these tools and techniques to be in better position in industry.

There are many advantages comes with both React.js and Node.js and both are widely used in web applications. React.js is open-source tool which is developed by Facebook and has many features to improve performance. On the other side, Node.js is a runtime environment based on Chrome's V8 JavaScript engine and used by companies like Twitter, Netflix, LinkedIn, etc. Both React.js and Node.js are open-source tools, so I will have free access; moreover, they can be executed client-side and server-

side and React.js code can be executed directly in the Node.js environment. By using this framework and runtime environment, I can access many tools and libraries, have better efficiency and productivity, have better development experience.

Because I am relatively new to React.js and Node.js., there will be a learning period for me to adjust and well perform in environment. System product is mobile and can run in any location with an active internet access. Because Private Notes App is a web application, it requires a stable internet connection to perform in desired way. In this case, system product cannot be used in a location where user cannot access to internet connection, such as in subways and underground.

Frameworks and databases used in this software product are hosted online; thus, the system is entirely internet dependent, which creates a limitation to usage location of product. There are other alternatives to develop Private Note's app; for instance, I could use Python with Django/Flask in backend development process instead of Node.js (JavaScript programming language), which is also in high demand and provides fast and efficient experience.

Project Management Plan

Esin Sari

Project 3: Private Notes

COP4331, Fall, 2020

Contents of this Document

Project Overview: Private Notes is a note-taking web application which enables users to store ideas, inspirations, tasks, daily lists, reminders, or any type of information. It provides ability to create, organize and search notes while ensuring distraction-free not-taking environment with its clear and easy access interface.

Applicable Standards

- Coding Standard
    - Appropriate and meaningful variable declaration
    - Using globally accepted coding styles
    - Commenting the code to ensure intelligibility
    - Following naming conventions, comment conventions, line length conventions, etc.
    - Reduction of complexity of methods and codes
    - Handling errors with meaningful error message
    - Applying other programming rules and principles.
    - See below links to get more information on Coding Standards which will be usen in this system product.
    - https://en.wikipedia.org/wiki/Coding_conventions
    - https://ace.apache.org/docs/coding-standards.html

- Artifact Size Metric Standard
    - Predicted number of classes: ∓10
    - Predicted number of lines of code: ∓2,000
    - Predicted number of input source: ∓ 10
    - API: ∓5
    - Frontend: Login page, page for changing password, buttons, user homepage, notes page, title and category entry tabs, sorting lists
    - Backend: storing user credentials, storing components of notes, database setting

Deliverables

Artifact	**Due Dates**
Individual Weekly Progress Reports	Weekly (Fridays) submission throughout the semester through Webcourses
Concept of Operations	September 18, 2020
Software Project Management Plan (SPMP)	September 18, 2020
Software Requirements Specification (SRS)	September 18, 2020
Test Plan	October 30, 2020
High-Level Design	October 30, 2020
Detailed Design	October 30, 2020
Test Plan	November 30, 2020

Test Results	November 30, 2020
Source, Executable, Build Instructions	November 30, 2020

Software Life Cycle Process: Phased Development: Incremental development will be used as software life cycle process. Developing process will start with focusing on small functional subsystems and additional functionalities will be inserted with each release. Reason for choosing this software life cycle process is because it enables developer to focus on different areas and it is easier to test and debug.

Tools and Computing Environment:

Operating System: Windows

Programming Languages: JavaScript, HTML, CSS, SQL

- Frontend: React.js, JavaScript, HTML,CSS
- Backend: Node.js, JavaScript, SQL
- Database: MySQL

Compilers: Visual Studio Code

Tools: Computer, Web Browser

Integrated Environment: Visual Studio Code

Configuration Management: GitHub will be used for configuration management. GitHub is known as well suited to handle version control and change control; it allows developer to create repository and those repositories could be accessible to public, so it generates easy collaboration between users, which is beneficial for teamworking. Moreover, keeping track of versions of product and maintaining consistency of product's performance and functionality could be accomplish easily by using GitHub. GitHub will be used frequently at every small feature is released and its adaptation with other features will also be controlled by the developer.

Quality Assurance: Plans will be established to determine processes and actions that are required to maintain and improve product quality. Results will be reported after each feature is completed by conducting code review process. Some QA activities that will be handled by developer;

- Testing
- Documenting frequently
- Analyzing performance of product
- Tracking schedule and development plan
- Addressing all requirements and different criteria
- Discussing external and internal characteristics of product
- Researching similar products and their innovations
- Releasing new versions with new functionalities
- Checking all criteria and goals of products are accomplished

Esin Sari is responsible for quality assurance.

Risk Management:

Human/Strategies Risks	**Impact Level**	**Strategy**
Misunderstanding requirements	Medium	Checking requirements; changing related features and functionalities; meeting with client if necessary
Personal shortfalls	High	Learning all necessary tools, programming languages, and ideas
Procrastinations	High	Keeping track of each process; not falling behind of schedule and plan
Illness	Low	Accepting outcomes/ricks and controlling; not falling behind of schedule and plan
Developing wrong function	High	Reconducting/editing function
Developing wrong user interface	High	Controlling requirements and design framework

Technical Risks	**Impact Level**	**Strategy**
Database crash	Medium	Informing user about technical issue with an error message and let them know the error will be fixed shortly.
GitHub crash	High	I will keep GitHub folder to prevent against GitHub crash
Input file problem	High	System will store copies of files
Library get deprecated	Low	Updating React.js/Node.js
IDE crashes	High	Updating Visual Studio Code

Business Risks	**Impact Level**	**Strategy**
A competitor is developing a similar product	Medium	Researching statistics and new versions of similar products in industry; analyzing their features, interface, and capabilities; releasing new versions with additional and improved features
Client dissatisfaction	High	Meet with client and make changes on requirements, functionalities, and features based on client's feedback

Table of Work Packages, Time Estimates, and Assignments:

Research: All necessary researches on appropriate programming languages, tools, similar note-taking apps, ideas, design, and other requirements and resources for web app development.

Define Requirements and Specification: Deciding the most relevant programming languages, data structures, tools, and environments. Define necessary actions and features to provide user and the boundaries.

Design Layout/Framework Design: Focusing on overall design of website and layout of each page to establish distraction-free not-taking environment for user.

Database Structure Design and Programming: Creating appropriate database system to store username, password (encrypted format), text notes, titles, categories, dates, and all other necessary variables.
*This work package may include the learning process of new language/tools.

Frontend Development: Creating 'client-side' where user have access to all features and functionalities of product and interact with framework. Establishing environment for user to interact with text, buttons, links, etc. and focusing on design of web application based on the determination from design layout work package.
*This work package may include the learning process of new language/tools.

Backend Development: Creating 'server-side' where developer can handle data in a way of storing, organizing, and answering requests of client. Backend development is required to make web application dynamic.
*This work package may include the learning process of new language/tools.

Testing: Testing program with different scenarios and inputs to catch faults; if ant errors occurred during testing, it is necessary to control both system design and program. There might be necessary error fixing, re-designing, or additional programming.

**All work packages will be handled by Esin Sari.

**Subject to change as more specific more goals are set.

Activity	**Time Estimate (in days)**
Research	4 days
Define Requirements and Specification	5 days
Design Layout/Framework Design	3 days
Frontend Development	10 days
Database Structure Design and Programming	10 days
Backend Development	10 days
Testing	5 days

Activity	September				October		November	
Research								
Define Requirements and Specification								
Design Layout/Framework Design								
Frontend Development								
Database Structure Design and Programming								
Backend Development								
Testing								

Technical Progress Metrics:

To track technical progress and plan the project, appropriate metric which focus on completed and uncompleted requirements will be used for different phases.

Requirements Phase: Metric will track the progress of the total number of requirements and the number of requirements changes.

Design Phase: Metric will track of UML diagrams, number of classes, packages, and functions completed.

Implementation Phase: Metric will track complexity of code and control if the code addresses all the requirements or not.

Testing Phase: Metric will track total number of errors and possible solutions to handle bugs. For system completion, project has to fix all possible errors and generate necessary actions (such as error message).

** For each phase above metrics will be used.

$$system\ completion = completed\ requirements\ /\ total\ requirement$$

$$requirement\ analysis = (analyzed/\text{gathered}\ requirements)\ /\ total\ requirement$$

Plan for tracking, control, and reporting of progress

I will collect all necessary data to build system product, for instance, learning new language, researching appropriate tools, researching statistics and similar web application to collect idea and information about content, interface layout, features, etc.

I will post my progress report weekly through Webcourses: the weekly report will include time spent in each activity, completed task(s), tasks in-progress, tasks planned for the following week, and individual issues and problems.

Each week, I read and analyze the logs; examine the technical content of the work done to date; examine the technical progress metrics; consider the QA results; reassess the potential project risks; and take corrective action if necessary.

Software Requirements Specification

Esin Sari

Project 3: Private Notes

COP4331, Fall, 2020

Contents of this Document

Section 1: Introduction

Software to be Produced:

- Software Product that I am producing for note-taking web application is one that is going to satisfy all the requirements listed in Software Requirements Specifications document.

Applicable Standards

- All standards that I have outlined in Project Management Plan are relevant for Software Requirements Specifications.

Definitions, Acronyms, and Abbreviations

- COTS: Commercially available off-the-shelf
- Self-explanatory/no acronyms, or abbreviations used

Section 2: Product Overview

Assumptions:

- Project scope will not be changed after the client and software development have agreed upon required features and functionalities.
- Software product will be created with JavaScript using React.js and JavaScript code will be executed during the operation of product.
- Software product will be fit into COTS category.
- There will be no budget for this software product.
- This software product will operate either on web-browser or mobile-phone with internet connection.

Stakeholders:

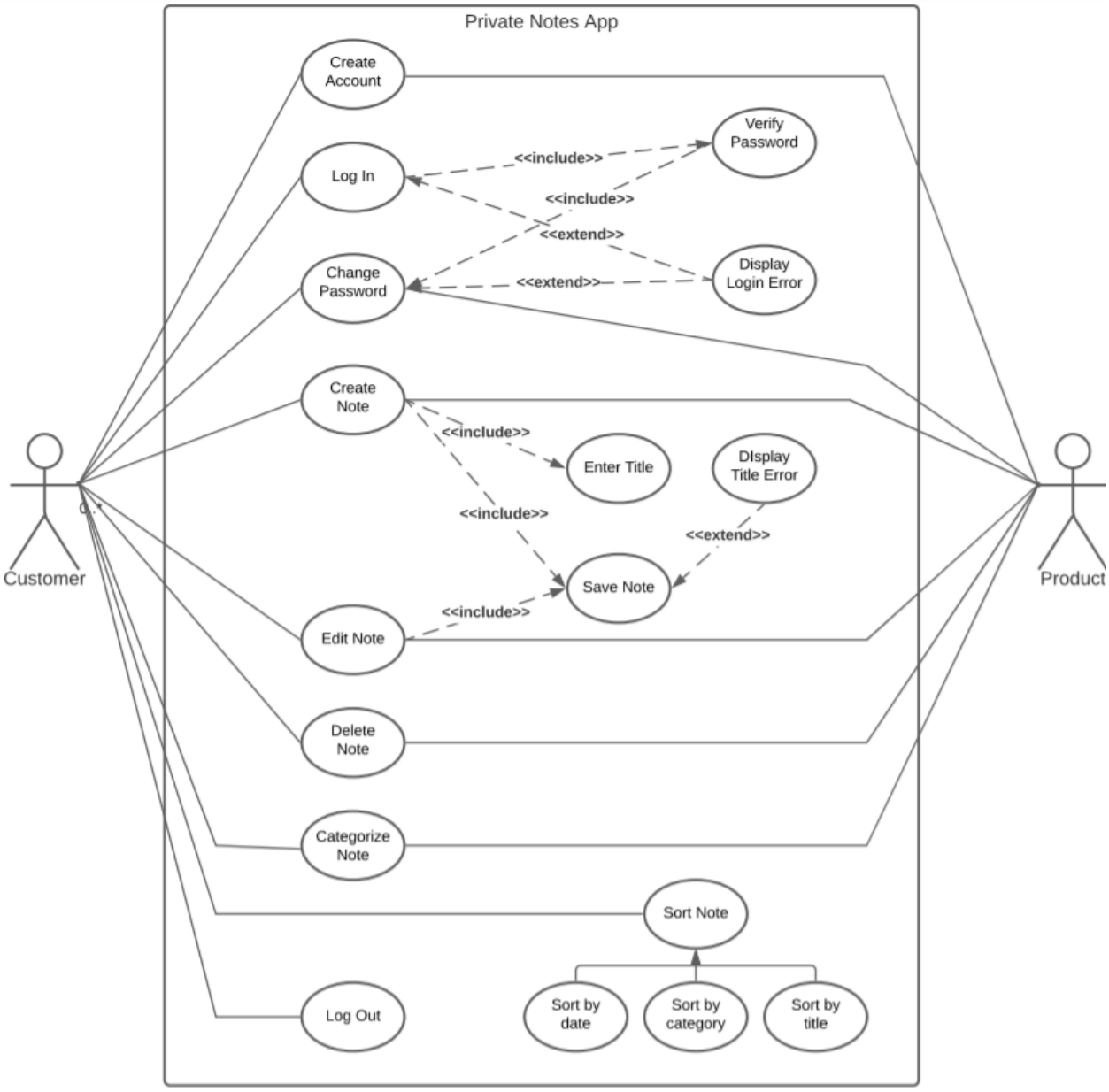
- Software Developer (Esin Sari): As a developer of this software product, I interact with the customer to ensure all requirements and features are implemented and functioning in desired ways.
- Customer: Nafisa will be the customer who I will communicate with during the completion process of product and discuss all requirements and features.

Event Table:

Event Name	External Stimuli	External Responses	Internal data and state
Log In	Enter credentials and click 'Login' button	Loading Page	○ Process Get request ○ Verify credentials ○ Generate authentication token for user ○ Send user to User Home Page in HTTP response. ○ In case of incorrect credentials, send landing page in HTTP response. ○ Display login error message
Change Password	Click 'Change Password' button and enter current and new passwords	Loading Page	○ Process Get request ○ Verify credentials ○ Generate authentication token for user ○ Send User to Loading Page and inform password successfully changed message. ○ In case of incorrect credentials, send landing page in HTTP response. ○ Display login error message
Create Note	Click 'Create Note' button and enter text information	Text Fields / Note Page	○ Process Get request ○ Send user to Note Page in HTTP response. ○ Store inputs (text and title) in database ○ In case of note without title, display title error message
Save Note	Click 'Save Note' button	User Home Page	○ Process Get request ○ Store notes in database ○ In case of note without title, display login error message ○ If no error, send user to User Home Page in HTTP response.
Edit Note	Click 'Edit Note' button	Text Fields / Note Page	○ Process Get request ○ Send user to Note Page in HTTP response. ○ Store new inputs (text or title)

			in database ○ In case of note without title, display title error message
Delete Note	Click ‘Delete Note’ button	User Home Page	○ Process Get request ○ Delete specific note from database ○ Inform user about deleting action by displaying message ○ Send user to User Home Page in HTTP response.
Categorization	Click ‘Categorize Note’ button	Text Fields / Note Page	○ Process Get request ○ Send user to Note Page in HTTP response. ○ Store category input in database ○ When activity is completed, send user to User Home Page in HTTP response.
Sort Note	Click ‘Sort Note’ button	Tags or menu with sort options / User Home Page	○ Process Get request ○ Get user input for option selection ○ Display notes in sorted way based on selected option
Log out	Click ’Logout’ button	Landing page	Process HTTP request

Use Case Diagram



Use Case Descriptions:

- Create Account: Customer first creates an account with selected username and password. Entered information will be taking by the system and stored in database; encrypted version of passwords will be stored.
- Log In: User enters username/passwords to access his/her homepage. Case must conduct relationship with Verify Password case and in case of error, Display Login Error case will conduct relationship with case.

- Change Password: User needs to enter current and new passwords to complete action. Case must conduct relationship with Verify Password case and in case of error, Display Login Error case will conduct relationship with case.
- Verify Password: All entered information must go into 'verify password case' to verify there is no mismatched error with password.
- Display Login Error: If any mismatched error is detected, Display Login Error will conduct relationship with relevant case and desired action will be failed.
- Create Note: User requests to create a new note in text form and must give a title to note; system will take both entry and store in database. Case must conduct relationship with Enter Title and Save Note case.
- Edit Note: User requests to edit existed note; system will apply changes in database. Case must conduct relationship with Save Note case.
- Save Note: Case saves inputs and will conduct relationship with Display Title Error case if note does not a title. In case error, system cannot save input in database; case must conduct relationship with Create Note and Edit Note case.
- Enter Title: User must give title to note; case must conduct relationship with Create Note case.
- Display Title Error: If there is a not without, Display Title Error will conduct relationship with relevant case and desired action will be failed.
- Delete Note: User requests to delete specific note; system will see the action and delete that note from its memory.
- Categorize Note: User requests to give category on specific note; system will take category information and store in its database.
- Sorting Note: User requests to search his/her notes in system; there are three options served to sort notes: sort by date, sort by category, sort by title
- Log Out: User requests to exit from his/her account.

Section 3: Specific Requirements

3.1 Functional Requirements

No: Web Application
Statement: System shall be web application
Source: Client and Developer
Dependency: Physical environment requirements
Conflicts: None
Supporting Materials: None
Evaluation Method: System product runs on web browser with an active internet connection
Revision History: Esin Sari, September 19, N/A

No: Authentication Process / Validity Checks of Inputs
Statement: System shall accept credentials from user and have capability of processing the authentication process and validity checks of inputs
Source: Client
Dependency: • Interface Requirements: Data Encryption, Accuracy of Data • User and Human Factors Requirements: Misuse of System • Data Requirements: Calculation for Password Encryption • Security Requirements: Security of User Account • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System access user information in database and control user inputs for validity checks
Revision History: Esin Sari, September 19, N/A

No: Responses to abnormal situations / Generate Error Message
Statement: System shall handle all possible errors caused by user; for instance, entering wrong username/password or trying to save a note without title; system shall also generates warning message in case of abnormal situations
Source: Client
Dependency: • Functional Requirements: Editing Credentials, Editing Notes and Related Information • Interface Requirements: Data Type, Accuracy of Data • Security Requirements: Security of User Account • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None

Evaluation Method: System displays error messages to UI in case of abnormal situation
Revision History: Esin Sari, September 19, N/A

No: Editing Credentials
Statement: System shall handle changing password action and authentication process
Source: Client
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Input • Interface Requirements: Data Items as Input, Data Type, Data Encryption, Accuracy of Data • User and Human Factors Requirements: Misuse of System • Data Requirements: Input for Credentials, Calculation for Password Encryption • Security Requirements: Security of User Account, Backup • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System edits/updates database with new credentials inputs
Revision History: Esin Sari, September 19, N/A

No: Editing Notes and Related Information
Statement: System shall allow user to make changes on inputs for notes, title, and category.
Source: Client
Dependency: • Functional Requirements: Sorting Notes • Interface Requirements: Data Items as Input, Data Type, Accuracy of Data, how often will each data item be received or sent • Data Requirements: Input for Notes, Input for Search/Organize Options

• Security Requirements: Backup • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System edits/updates database with new note inputs
Revision History: Esin Sari, September 19, N/A

No: Sorting Notes
Statement: System shall allow user to sort notes based on its title, category, and date.
Source: Client
Dependency: • Interface Requirements: Data Items as Output, how often will each data item be received or sent • Data Requirements: Input for Search/Organize Options
Conflicts: None
Supporting Materials: None
Evaluation Method: Systems displays notes based on selected sorting option
Revision History: Esin Sari, September 19, N/A

No: Relationship of Outputs to Inputs
Statement: System shall respond to users' requests; for instance, sorting the notes based on selected category or displaying selected notes to user interface
Source: Client

Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, Editing Credentials, Editing Notes/Related Information, Sorting Notes • Interface Requirements: Data Items as Input/Output, Data Type, Accuracy of Data, how often will each data item be received or sent • Data Requirements: Input for Credentials, Input for Notes, Input for Search/Organize Options • Security Requirements: Backup • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System is successfully responding to user request
Revision History: Esin Sari, September 19, N/A

3.2 Interface Requirements

No: What Data Items Are Input
Statement: System shall accept data items like username, password, notes as text form, title, date, and category information as input
Source: Client and Developer
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, Editing Credentials, Editing Notes/Related Information, Sorting Notes • Interface Requirements: Data Type, Accuracy of Data • Data Requirements: Input for Credentials, Input for Notes, Input for Search/Organize Options • Security Requirements: Backup • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System accept inputs for username, password, notes as text form, title, etc. and

perform expected actions
Revision History: Esin Sari, September 19, N/A

No: What is the Data Type
Statement: System shall different data types for different inputs. • String Data Type: Username, notes(might be text document), date, titles, and category • Integer Data Type: Password
Source: Client and Developer
Dependency: • Interface Requirements: Accuracy of Data • Data Requirements: Input for Credentials, Input for Notes, Input for Search/Organize Options • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System accept inputs and does not perform action if data type is not valid
Revision History: Esin Sari, September 19, N/A

No: What Data Items Are Output
Statement: System shall output error message in case of authentication issue or saving notes without title, each created notes with its title, category, and date.
Source: Client and Developer
Dependency: • Functional Requirements: Sorting Notes • Interface Requirements: Data Items as Input, Accuracy of Data, how often will each data item

be received or sent • Data Requirements: Input for Search/Organize Options • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System outputs either responds for user request or error message
Revision History: Esin Sari, September 19, N/A

No: Data Encryption
Statement: System shall encrypt user password with appropriate hashing methods to ensure security of information
Source: Client
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, • Interface Requirements: Accuracy of Data • Data Requirements: Input for Credential • Security Requirements: Security of User Account • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System encrypt passwords before storing on database
Revision History: Esin Sari, September 19, N/A

No: How Often Will Each Data Item Be Received or Sent
Statement: System shall receive and sent data items based on the user request. System shall display created notes with their titles in user interface every time user logs in to system.

Source: Client and Developer
Dependency: • Functional Requirements: Editing Credentials, Editing Notes/Related Information, Sorting Notes • Data Requirements: Input for Credentials, Input for Notes, Input for Search/Organize Options
Conflicts: None
Supporting Materials: None
Evaluation Method: System is successfully receiving and sending data items based on the user request; system will be tested with different inputs
Revision History: Esin Sari, September 19, N/A

No: Accuracy of Data
Statement: System shall handle authentication process to verify user password and check whether input is relevant with data type or not.
Source: Client and Developer
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, Responses to Abnormal Situation/Generate Error Message, Editing Credentials • Interface Requirements: Data Type, Data Encryption • Data Requirements: Input for Credentials • Security Requirements: Security of User Account • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System allows user login if credentials are met, otherwise, displays error message; system will be tested with different inputs
Revision History: Esin Sari, September 19, N/A

3.3 Physical Environment Requirements

No: Active Internet Connection
Statement: System requires an active internet connection to operate
Source: Client and Developer
Dependency: • Functional Requirements: Web Application • Physical Environment Requirements: Environmental Consideration, Types of Equipment • Resource Requirements: Hardware/Software/Tools
Conflicts: None
Supporting Materials: None
Evaluation Method: System runs on web browser with an active internet connection
Revision History: Esin Sari, September 19, N/A

No: Environmental Consideration
Statement: System shall operate in any location where user can access to internet connection; system product cannot perform from locations like subway and underground.
Source: Client
Dependency: • Functional Requirements: Web Application • Physical Environment Requirements: Active Internet Connection • Resource Requirements: Hardware/Software/Tools
Conflicts: None
Supporting Materials: None
Evaluation Method: System runs on web browser with an active internet connection
Revision History: Esin Sari, September 19, N/A

No: Types of Equipment
Statement: System shall run on equipment such laptop, mobile phone, tablet, etc. on web browser with an internet connection
Source: Client
Dependency: • Functional Requirements: Web Application • Physical Environment Requirements: Active Internet Connection • Resource Requirements: Hardware/Software/Tools
Conflicts: None
Supporting Materials: None
Evaluation Method: System runs on any equipment which has access to internet connection
Revision History: Esin Sari, September 19, N/A

3.4 User and Human Factors Requirements

No: Types of Users
Statement: System shall be accessible to anyone regardless of skill level, profession, age, etc.
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Anyone regardless of skill level, profession, age, etc. can use product
Revision History: Esin Sari, September 19, N/A

No: Additional Training and Documentation
Statement: System is designed in purpose of being easy to use and understandable; thus, system does not require any skill level, training, or documentation
Source: Client
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Anyone can use system product without having need of training or documentation
Revision History: Esin Sari, September 19, N/A

No: Misuse of System
Statement: System shall detect and prevent misuse with authentication.
Source: Client
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, Responses to Abnormal Situation/Generate Error Message • Interface Requirements: Data Encryption, Accuracy of Data • Data Requirements: Input for Credential • Security Requirements: Security of User Account • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System displays error message; system will be tested with different inputs
Revision History: Esin Sari, September 19, N/A

3.5 Documentation Requirements

No: Documentation
Statement: Because the system is accessible to anyone and easy to use, it does not require any documentation
Source: Client and Developer
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: None
Revision History: Esin Sari, September 19, N/A

3.6 Data Requirements

No: Input for Credentials
Statement: System shall store username/password to manage login actions
Source: Client
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, Responses to Abnormal Situation/Generate Error Message, Editing Credentials • Interface Requirements: Data Type, Data Encryption • Security Requirements: Security of User Account • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System load inputs into database
Revision History: Esin Sari, September 19, N/A

No: Calculation for Password Encryption
Statement: System shall perform hash function calculation to encrypt user password to ensure security
Source: Client
Dependency: • Interface Requirements: Data Encryption • User and Human Factors Requirement: Misuse of System • Data Requirements: Input for Credentials • Security Requirements: Security of User Account
Conflicts: None
Supporting Materials: None
Evaluation Method: System perform hashing methods to password before loading into database
Revision History: Esin Sari, September 19, N/A

No: Inputs for Notes
Statement: System shall store each information related with specific note; text document, title, category, date, etc.
Source: Client
Dependency: • Functional Requirements: Editing Notes/Related Information, Sorting Notes, Responses to Abnormal Situation/Generate Error Message • Interface Requirements: What Data Items are Input • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System load inputs into database

Revision History: Esin Sari, September 19, N/A

No: Input for Search/Organize Options
Statement: • System shall accept user input on search/organize options • system shall display notes sequentially based on title, category, or date options.
Source: Client
Dependency: • Functional Requirements: Sorting Notes • Interface Requirements: What Data Items are Output?
Conflicts: None
Supporting Materials: None
Evaluation Method: System get sorting request and displays notes in UI based on selection
Revision History: Esin Sari, September 19, N/A

3.7 Resource Requirements

No: Skilled Personnel to Build, Use, and Maintain System
Statement: Developer (Esin Sari) will be handle the design, planning, development, and testing process
Source: Client and Developer
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: System product is completed by Esin Sari

Revision History: Esin Sari, September 19, N/A

No: Schedule
Statement: System design and development steps must be scheduled detailly by the developer and schedule must ensure to address the due dates
Source: Client and Software Developer
Dependency: Skilled Personnel to Build, Use, and Maintain System
Conflicts: None
Supporting Materials: None
Evaluation Method: System product and each sub features and functionalities are completed by due date
Revision History: Esin Sari, September 19, N/A

No: Hardware/Software/Tools
Statement: • System shall utilize from determined software tools such as React.js, JavaScript programming language, or MySQL to create the product • System shall utilize system design and system schedule/plan • System shall work in any web browser with reliable internet connection
Source: Client and Software Developer
Dependency: • Functional Requirements: Web Application • Physical Environment Requirements: Active Internet Connection, Environmental Consideration, Types of Equipment
Conflicts: None

Supporting Materials: None
Evaluation Method: System product is successfully running, performing all expected features, and addressing all requirements.
Revision History: Esin Sari, September 19, N/A

3.8 Security Requirements

No: Security of User Account
Statement: • System must handle encryption operations for user password and handle authentication • System must isolate one user's date from others • System must control one's access to other systems and information
Source: Client and Developer
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, Responses to Abnormal Situation/Generate Error Message • Interface Requirements: Data Encryption, Accuracy of Data • User and Human Factors Requirements: Misuse of System • Data Requirements: Input for Credential, Calculation for Password Encryption • Security Requirements: Security of User Account, Backup • Quality Assurance: Detect and Isolate Faults
Conflicts: None
Supporting Materials: None
Evaluation Method: System displays error message; system will be tested with different inputs
Revision History: Esin Sari, September 19, N/A

No: Backup

Statement: System shall backup every time user add/delete/edit information and store backup copies in different location for recovery
Source: Developer
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: In case database crash, there will be a backup process to save user information
Revision History: Esin Sari, September 19, N/A

3.9 Quality Assurance Requirements

No: Detect and Isolate Faults
Statement: System must detect mismatch password and output error message; System must also detect notes without title and output error message. In both cases, system will restrict user to complete action.
Source: Client and Developer
Dependency: • Functional Requirements: Authentication Process/Validity Checks of Inputs, Responses to Abnormal Situation/Generate Error Message • Interface Requirements: Accuracy of Data • User and Human Factors Requirements: Misuse of System • Security Requirements: Security of User Account
Conflicts: None
Supporting Materials: None
Evaluation Method: System displays error message; system will be tested with different inputs
Revision History: Esin Sari, September 19, N/A

No: Availability of System
Statement: System should be available 24 hours a day
Source: Client and Developer
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: User have access to application any time
Revision History: Esin Sari, September 19, N/A

No: Prescribed Time for User to Adjust Product
Statement: Any user should be able to complete following tasks in less than 10 minutes; create account, create note, edit note, delete note, categorize and sort note
Source: Client and Developer
Dependency: • User and Human Factors Requirements: Types of Users
Conflicts: None
Supporting Materials: None
Evaluation Method: User can work with web application without having need to any training or documentation
Revision History: Esin Sari, September 19, N/A