**EECS 1022 Project: Phase 2**

**Team Name:** The Tridroid Developers

**Team Members:** Aijaisarma Sabaratnasarma, Andre G. Muyalde, Ethan Rei C. Afable

**Project Title:** Password Protected Diary App

**Project Description:**

Our app will load up with a screen welcoming the user and directing them to press a button and enter a password to access their notes. If the password is entered correctly, they will be able see a list of notes with the name of the note as a label, then clicking on it will reveal the note itself. There will also be 2 buttons to add new notes and delete notes that are not needed. If the user leaves the screen that shows all the notes, then to go back to that page they will have to re-enter the password. If we have time, we might add a feature to change your password or to edit currently saved notes.

**Functional Requirements for Project:**

1. Users can set a password for their diary and change it at any time.

* The first time the app is used, it will ask the user to set a password. Any subsequent uses of the app, the user must put in their password to access the diary.
* If the user forgets their password, there will be a “change password option”.
* Passwords can contain any type of characters on a keyboard but must be within 15 characters.
* All password features will be operated using buttons and methods in Android Studio.

1. In the main menu of the app, users will be able to scroll through their notes.

* Each note will have a numeric label with a title that the user sets. If a title is not specified, the default title given will be “Note [positive integer]”.
* Each note label will be clickable to access the note on a separate page controlled by Android Studio.

1. Users can create new notes by pressing a button on the main menu of the app.

* The create button will prompt the user to give a title for the new note, after which they click a confirm button.
* Adds a new note label to the main page, which then the user can click to edit their newly created note.
* Button will be located at the bottom right of the main menu screen

1. Users can delete existing notes by pressing a button then selecting the notes to delete and pressing the same button.

* When notes are deleted, the main menu should enumerate itself to make sure the numbers of each note label are in order.
* Button will be located at the bottom of the main menu screen, centered.

1. Each note page will have a button to save their current changes made to the note.

* If a user tries to leave the note’s page without saving, the user will be prompted if they would like to save before exiting.

1. Each page on the app will have a “back” button to make navigating the app easier by taking the user to the previous page.

* If the user is on the main menu, then the back button goes to the launch screen (note to get back to the main menu the password must be re-entered).
* If the user is within one of their notes, then the back button takes them to the main menu where all their notes can be found.

1. Ability to recover previous edits of notes.

* There will be a button on the page of each note the user creates that gives access to previous edits. (Through either another page or a slide down window)
* The app will store the past 5 “saves” of a note they had made.
* The user will be able to preview previous iterations of their notes on a separate page.
* Reverting will erase the saved edits that come after the chosen to save (user will be notified if this is ok with them)

1. Users will have access to a toolbar to format their text in their notes in different ways.

* Ability to bold, italic, underline, etc.
* Collapsible; Users can reveal the toolbar through a user input, so it is not always on the screen.

1. Users will be able to add text to their notes via the following “texting” based method.

* Users enter text through a textbox and push/send their additional text to their existing notes.
* Existing text within their notes may be removed by the user through a deletion button beside the notes.

1. Search bar for searching notes by note title on the main menu.

* Search for the name of the note by its filename.
* The app will
* bring up the file with the corresponding filename.
* bring up the files with the closest filenames if there is no file with the filename (e.g. searched for ‘blue’; returns ‘blue1.txt’ ‘blue2.txt’).

**Wireframes of User Interface:**

**Opening Screen: Menu of Notes:**

**Graphical user interface, application

Description automatically generatedText

Description automatically generated**

**Menu of Notes (Note Deletion View): Note Editor Screen:**

**Text, letter

Description automatically generatedTable

Description automatically generated**

**Note History Page: Note Editor Screen (Blank View):**

**Shape

Description automatically generatedGraphical user interface

Description automatically generated**

**Diagram

Description automatically generated**

**Here is a functional view of all the wireframes together with mobile interactions to access different wireframes (prototype):** [**https://www.figma.com/proto/yYpxm9uzGJSHYwdTuoGRsT/Password-Protected-Diary-App?node-id=1%3A2&scaling=scale-down&page-id=0%3A1&starting-point-node-id=1%3A2**](https://www.figma.com/proto/yYpxm9uzGJSHYwdTuoGRsT/Password-Protected-Diary-App?node-id=1%3A2&scaling=scale-down&page-id=0%3A1&starting-point-node-id=1%3A2)

**Figma Layouts showing all Wireframes with Arrows:** [**https://www.figma.com/file/yYpxm9uzGJSHYwdTuoGRsT/Password-Protected-Diary-App?node-id=0%3A1**](https://www.figma.com/file/yYpxm9uzGJSHYwdTuoGRsT/Password-Protected-Diary-App?node-id=0%3A1)

**GitHub Repository:** [**https://github.com/duyamn/Tridroid-Developers-Password-Protected-Diary-1022-Project**](https://github.com/duyamn/Tridroid-Developers-Password-Protected-Diary-1022-Project)