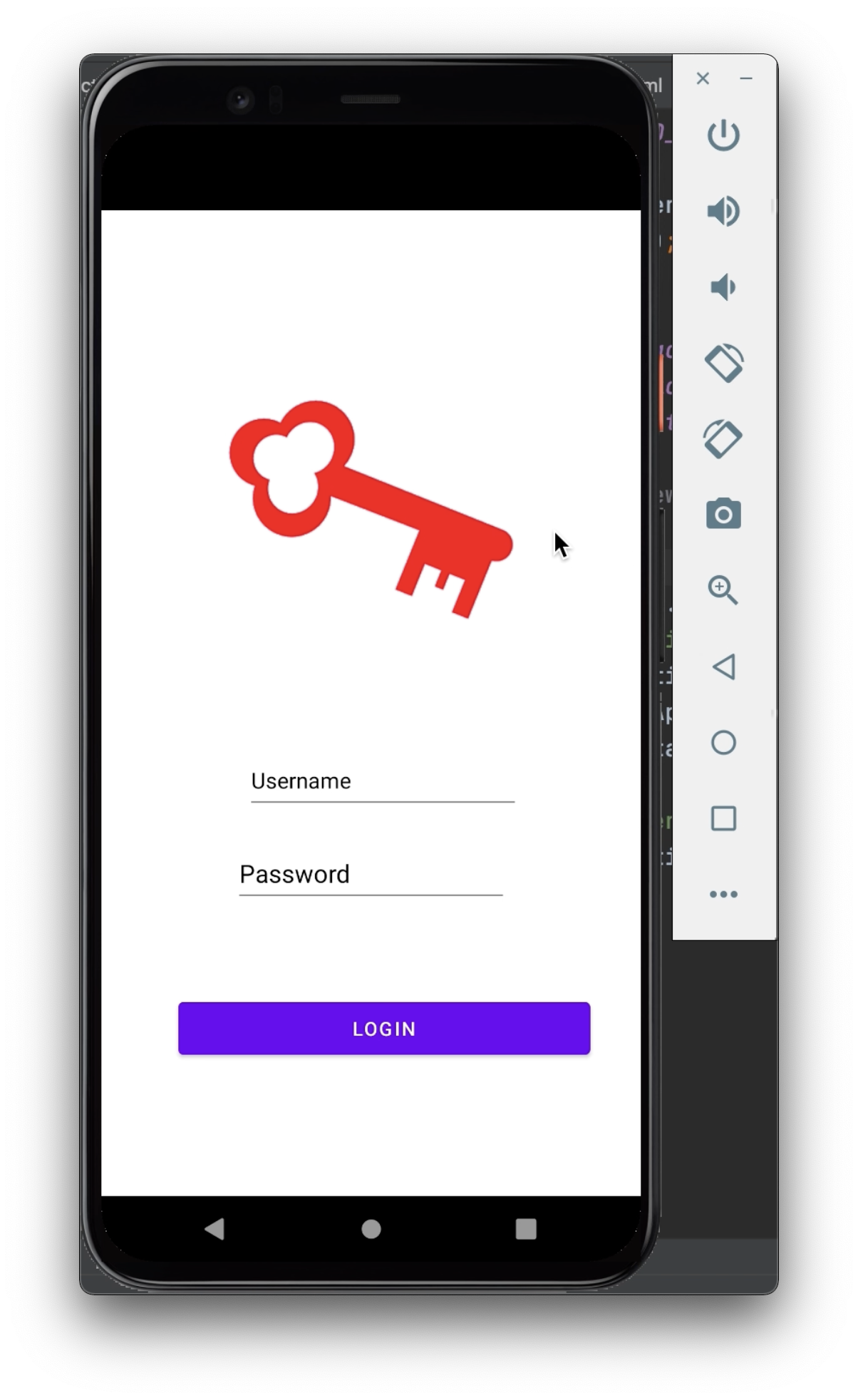
**RazKey**

By: Nicholas Hardy, Kaede Kawata,

Yen Vo, James Sullivan

**Introduction →**

For our final project, the four of us all collaborated in the design and creation of our product, *RazKey*, a secure passwords and sensitive information app with connectivity to a mobile Raspberry Pi. The project was split into two main aspects. The front end app design and code, and the backend raspberry pi component. The primary focus for us was to create a fully functional and robust backend program that stores and passes information safely upon request. 

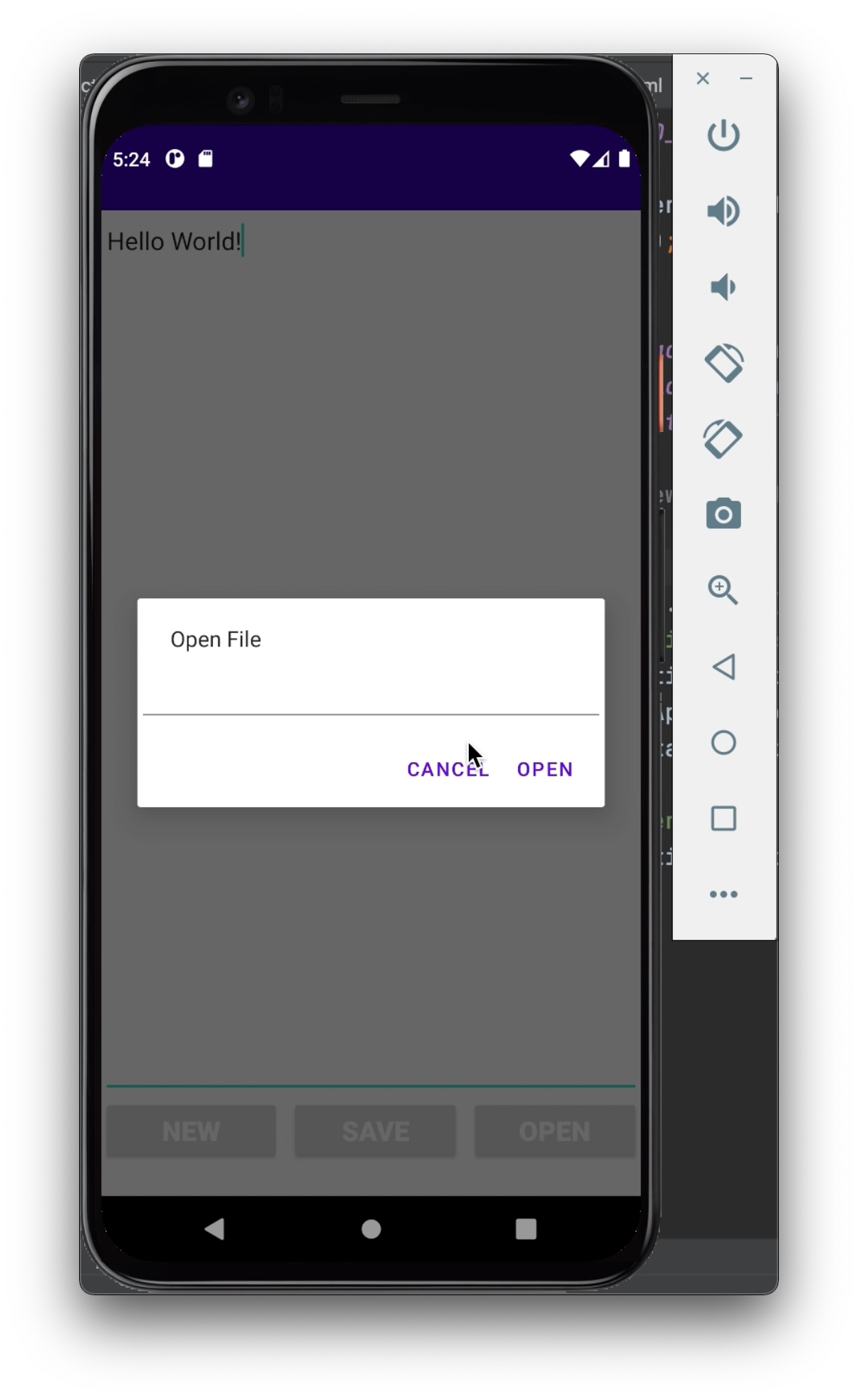
Our initial concept came to fruition during our first team meeting when we all agreed to find a way to include a hardware aspect to our final project, thus including the Raspberry Pi as our wireless storage device. The aspect of including a physical device such as the arduino through which one can easily and safely access data is appealing because only one person has access to that data at a time. It is applicable for any person regardless of their technical knowledge or their age.

As college students in the digital age we often all carry around backpacks filled with tablets and laptops all of which store information on the cloud. Although most services are exceptionally safe with their encryption, the RazKey offers the next level of security. Only the user has access. The device is small and portable and can easily be thrown into a backpack along with other devices.

**Technical Brief →**

The project was split into two main aspects. The front end app design and code using Android Studio, and the backend raspberry pi component. We use Android Studio to build and design the appearance of the app. The Raspberry Pi functions as our host for the back end program and is written in Java. The primary focus for us was to create a fully functional and robust backend program that stores and passes information safely upon request.

**App Functionality →**

When the user opens the app they will see a login screen and are prompted to enter their username and password in order to log in. Once logged in successfully, the user will be able to see the menu screen that shows different folders such as passwords, birthdays, and recipes, to access. By clicking on the folders (button), the user will be directed to a new screen, which shows the information stored in that folder. In the ‘Passwords’ folder, the user can if the user pressed the ‘New Password’ button they are redirected to a keyboard and new password entry. Pressing the save button returns the user back to the passwords screen while saving the password. The user can also return to the menu screen from the folder by clicking the ‘Menu’ button. The user can also add a new folder by clicking the ‘New Folder’ button.

**Raspberry Pi Functionality →**

The Raspberry Pi acts as the host for our back end program. The back end is written in Java and contains a number of functions. The program begins by asking the user to enter a new password which is then stored for the first time as the login password. The program then again prompts the user to log in and enter their newly created password.

If the user ever enters the wrong password then the program loops again printing out the error message until the correct answer is entered.

Once the correct answer is entered the program then prompts the user to choose which folder, or in this case file, to view the contents written. The program accepts three character inputs for the three different files, ‘p’ for “Passwords.txt”,’r’ for “Recipes.txt”, and ‘b’ for “Birthdays.txt”. The program then reads and returns the data written in the chosen .txt file while also asking the user if they would like to enter any new data and write it to the file. If the user enters ‘no’ or ‘No’ then an empty string is stored. If there is an entry then the data is appended to the end of the file and then returned to the user where all the data is visible. including the new user input.

The device communicates to the app via the ability to use ssh commands on the raspberry pi and also the functionality of java on both Android studio and the raspberry pi.

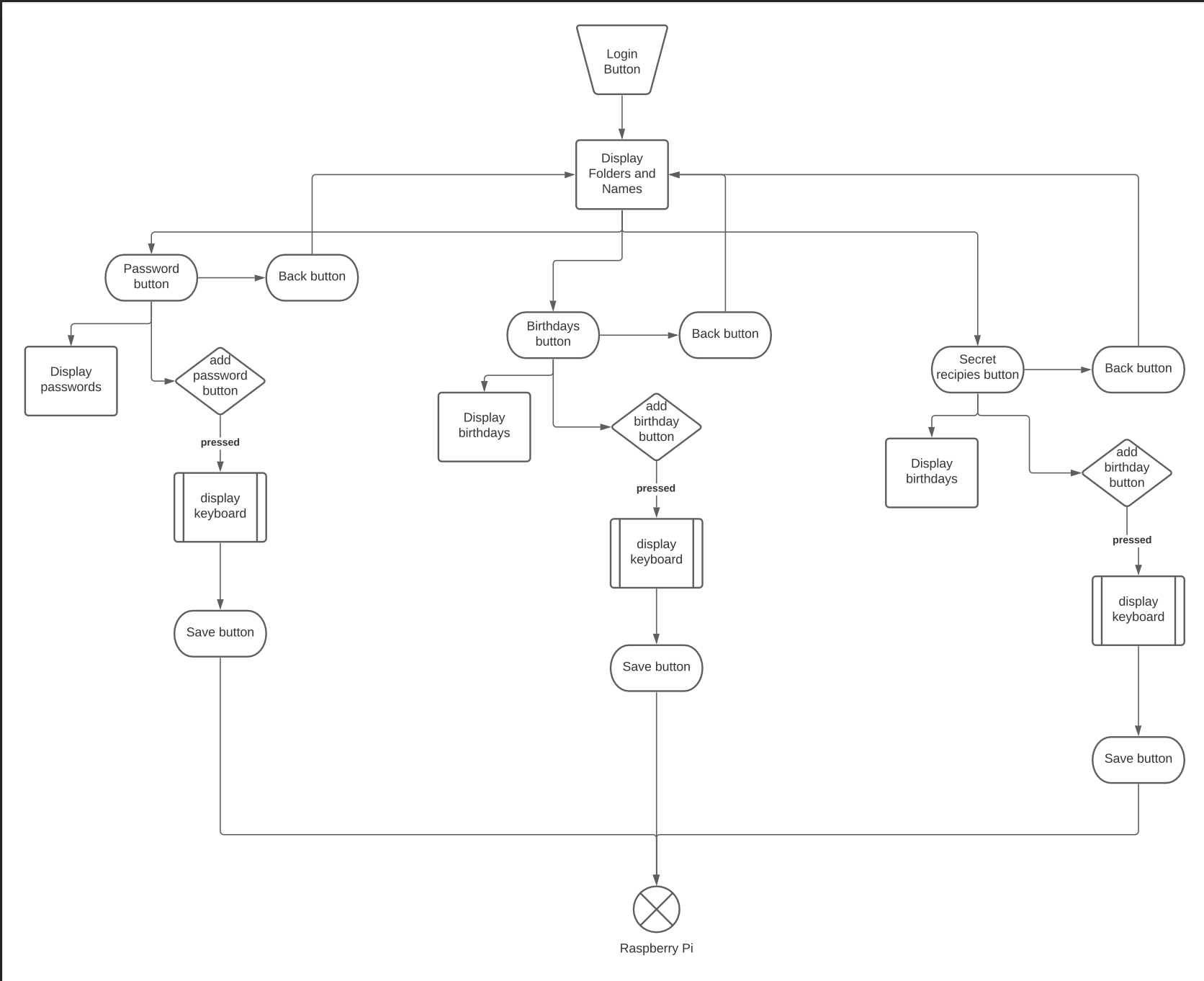
**Challenges →**

Throughout the process we ran into obstacles along the way which forced us to modify our designs, which created different challenges for us. The first challenge that we faced was that we decided to use Android Studio to design the app and it required us to convert our code, which was originally written in C++, to Java. This was challenging as for us the last exposure we had to any form of Java was taking a high school AP class.

Along with the conversion of C++ to Java, we also had to do extensive research on the SSH functionality and how to implement it to our design. The initial connection to the Raspberry Pi was simple through our computers’ terminal command line, but then making the input from the app was where we hit a wall.

**Summary →**

Creating a secure passwords and sensitive information app using Android Studio and Raspberry Pi was a good chance for us to challenge ourselves and to learn new skills, which might help us in the future. We learned how to code our backend in C++ and Java. We also learned how to create and write a program as a team. We found out that sharing our code through github is much easier and more efficient than sharing it through email, which most of us used to do before this class. Although unsuccessful with our initial goals, it was the first time that we created an Android app and we all had a good time working on it together

**Figure 1: 1st design of front end architecture**

**Figure 2: 1st design of back end architecture**

