System and Unit Test Report
Yet Another Password Manager

Team: William Walker, William Connor Koch, Kyle Oda, Theodore Tefera, Shaurya Kapoor
Revision #1, 2018-11-30

System Test Scenarios

A. Sprint 1 & 2 User Story 1: As a user, I want to be able to create new passwords, so that I can keep track of all the details and don't have to remember them

B. Sprint 1 User Story 2: As a user, I want to store the passwords I create on my phone so that they are always with me.

C. Sprint 1 & 2 User Story 3: As a user, I want to be able to see my passwords in a list, so that they are all in one place, easy to see.

   Scenario:

   1. Start YAPM app; Tap on plus button in bottom right corner; type
      - Name = <Amazon>
      - Username = <BarackObama>
      - Password = <Super*Secure^Password$>
      - Confirm Password = <Super*Secure^Password$>
      - URL = <amazon.com>
      - Notes = <Blah Blah Blah>
   2. Press Save button
   3. User should see the name of their new password at the bottom of the screen in a list view, as well as any other passwords they created previously.

D. Sprint 2 User Story 3: As a user, I want to be able to manage and edit existing passwords, so that I can update my passwords when I change them.

   Scenario:

1. Start YAPM app
2. Tap on an existing account password
3. Click the “Edit” button on the pop-up dialog
4. Change each field in the view password fragment
5. Click save
6. Click on the updated account password to see if the changes were stored

E. Sprint 2 User Story 4: As a user, I want to be able to use a password generator to create a new randomized password, so that I know it is a secure, unique password.

Scenario:

1. Implant “passwordGen” and “characterGen” methods into a separate testing environment
2. Run the passwordGen method 1,000 times with special characters enabled and 1,000 times without special characters enabled. Output each line to test.txt
3. Run a script to ensure no out of place characters and to check for an equal distribution of characters
4. Check resulting log, keep in mind due to random number generation the percentage of use for each character will not be exactly even

F. Sprint 3 User Story 1: As a user, I want to be able to generate random passwords with a certain length, so my passwords can pass online password checkers.

Scenario:

1. Start YAPM app; Tap on plus button in the bottom right corner
2. Click Generate Random Password Button
3. Enter 10 for password and check Allow Special Characters
4. Click Generate
5. Click show password button
6. The user should see a matching 10 character password that contains special characters in both password and confirm password fields.

G. Sprint 3 User Story 2: As a user, I want to be able to delete old passwords so that my view isn't cluttered.

   Scenario:

   1. Start YAPM app; Tap on plus button in the bottom right corner
   2. Press Save to create a blank entry
   3. Click on the blank entry; click delete; click yes
   4. The user should see a toast at the bottom at the bottom of the screen that says Password Deleted and they should not see the blank entry in the list view anymore.

H. Sprint 4 User Story 1: As a user, I want my passwords to be secure whenever I'm not using the app, and I want to be the only person who can access them so that no one can steal them.

   Scenario:

   1. Start YAPM app
   2. The user should see a login screen where they have to input a password and scan their fingerprint.

J. Sprint 4 User Story 4: As a user, I want to be able to copy the password and go directly from the app to the website that the password is for so I don't have to click a lot.

   Scenario:

   1. Start YAPM app
   2. Create a password and save it
   3. Click on the password to get the pop-up dialog box
   4. Click on the copy icon
   5. Exit the app and paste on the the web browser
   6. Check to see if pasted data was the same string as the password.

<u>Unit Tests</u>

Basically, one idea is to mention all the edge cases which we tested out.
We can also mention we tested out the base case and also cases with randomized data.

OR we talk about the tasks for each user story and how we tested those individual tasks

- Possibly yeah, we can do that too.

- Sprint 1: User story 1
    - Task 1: Create a New Password Fragment (View)
    - Task 2: Create a New Password Fragment (Controller), connect to UI elements
        - Tested it by attempting to create a new password to view the new password screen
- Sprint 1: User Story 3
    - Task 1: Create Password List Fragment (View)
    - Task 2: Create Password List Fragment (Controller) - setup ListView adapter w/Pagination
        - Saved a password and checked to see if it was visible on the main screen
- Sprint 2: User Story 1
    - Task 1: Implement safe arguments so that view can be used to create or edit passwords (currently only creates)
        - Testing it by inputting random arguments.
- Sprint 2: User Story 2
    - Task 1: Attach RealmRecyclerView to the adapter so it displays entries (1.5 hours)
        - Testing it by making it creating multiple passwords and seeing if they displayed
- Sprint 2: User Story 3
    - Task 1: Enable editing to stored passwords.
        - Tested it by trying to edit stored password already in the database.
- Sprint 2: User Story 4

    - Task 1: Create guideline options for password generation.
        -
    - Task 2: Create a method to securely create random passwords with options passed as arguments
- Sprint 3: User Story 1
    - Task 1: Edit the existing random password generator button to prompt the user for inputs
        - Test if the edit button existed
- Sprint 3: User Story 2
    - Task 1: Enable deletion of passwords from database (2 hours)
        - Tested by creating and deleting passwords
- Sprint 3: User Story 3
    - Task 1: Implement the generation and storage of keys and android secure storage
        - Tested by logging in, closing the app, and then opening the app again to see if the information was saved
- Sprint 3: User Story 4
    - Task 1: Add the edit button to the popup view of the existing password
        - Tested by starting app, clicking on existing password, and checking to see if the edit button was there.
    - Task 2: Once button is pressed, pass the existing password arguments to populate the edit password view. (2 hours)
        - Tested by starting app, clicking on existing password, clicking the edit button, and checking to see if the existing password's fields are populating it
- Sprint 4: User Story 1
    - Task 1: Create a login view and set it as the start destination (1 hour)
        - Tested by starting the app, checking to see if the login view is the first thing is visible to the user
    - Task 2: Implement fingerprint authentication (2 hours)
        - Tested by starting the app, attempting to use the fingerprint authentication to login
- Sprint 4: User Story 3

    - Task 3: Show an error message to the user if the password and confirm password edit texts do not match
        - Tested by starting app, checking to see if an invalid password could be saved
    - Task 4: Show the website with each entry in the password list fragment
        - Tested by starting app, checking to see if an existing
- Sprint 4: User Story 4
    - Task 1: Make passwords able to be copied with one click
        - Tested by starting app, clicking on existing password, clicking the copy password button, exiting the app, and pasting on another app to see if it was copied correctly
    - Task 2: Make the website url clickable (2 hours)
        - Tested by starting app, clicking on existing password, and clicking on the url to see if it takes you to the right website