GitHub Username: dconz13

Strong Password Creator

Description

Tired of worrying about password strength? Worried about keeping your accounts secure? Look no further! Strong Password Creator will do all the heavy lifting for you by generating unique and virtually uncrackable passwords!

Features:

- Generate easy to remember passwords
- Securely save passwords using built in encryption
- Organize passwords in your password locker
- Backup encrypted passwords to the cloud

- Password protected locker with fingerprint support
- Supports 15 different languages for password generation (Catalan, Dutch, Esperanto, Finnish, French, German, Italian, Japanese, Maori, Norwegian, Polish, Russian, Spanish, Swedish and Turkish)

Warning:

- If you lose the key to your locker your passwords will be lost forever

Intended User

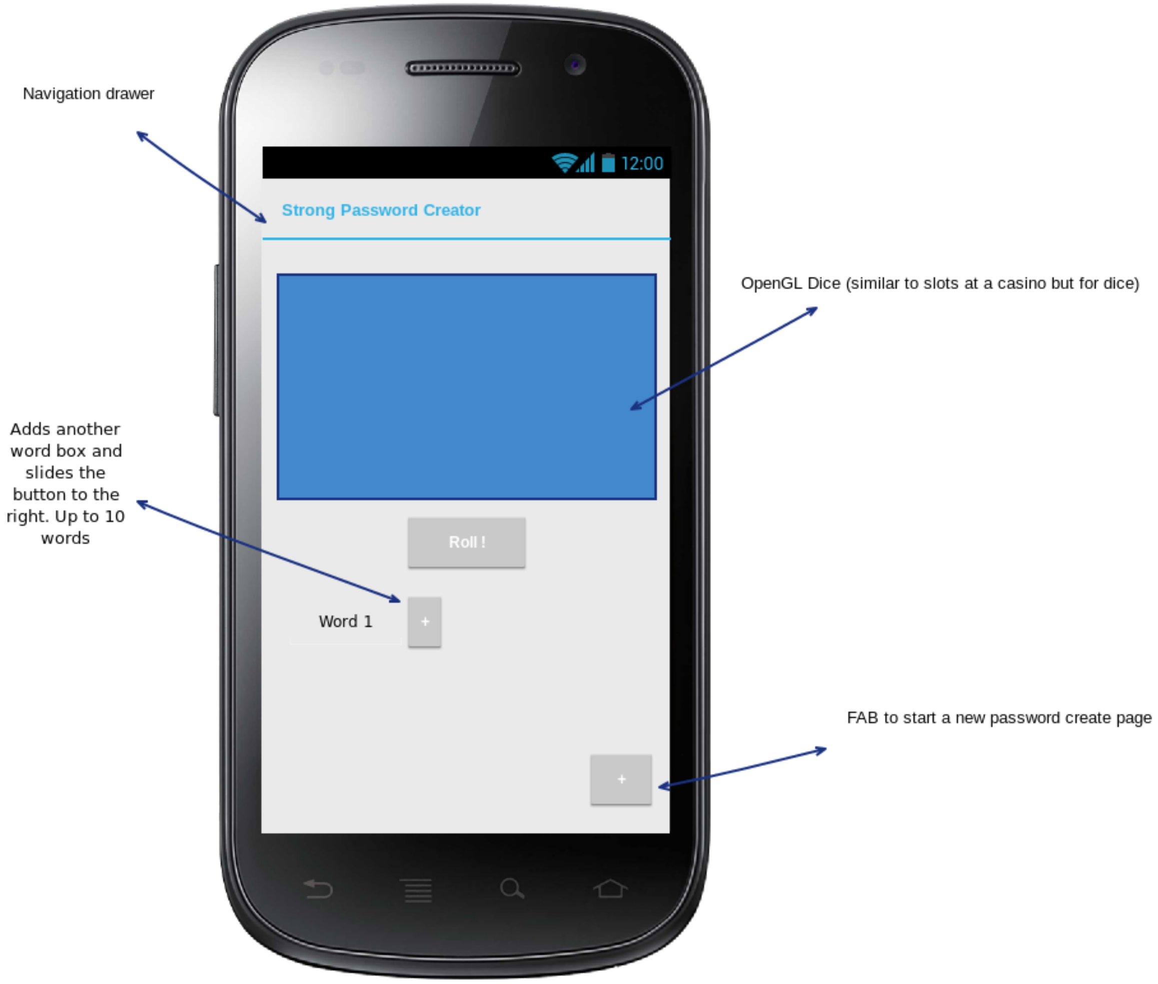
Users who wish to have strong passwords or wish to simply organize and store their passwords securely.

Features

- Generates secure passwords using the Diceware passphrase list
- Has a first time run tutorial of how to use the app and tips on password length
- Ability to sync encrypted locker file to a cloud of user choice
- Ability to lock the locker file with a password to protect passwords from snoopers
- Support for 15 different languages using the different language diceware passphrase lists
- Organize the passwords within the locker in a meaningful way
- Animated user interface to show dice being rolled and the words appearing
- Allow users to swap out words that they don’t like or that are hard to remember for them
- Google Analytics to track user usage
- Allow for free usage with banner ads or donate to disable ads

User Interface Mocks

Main Activity



Navigation drawer

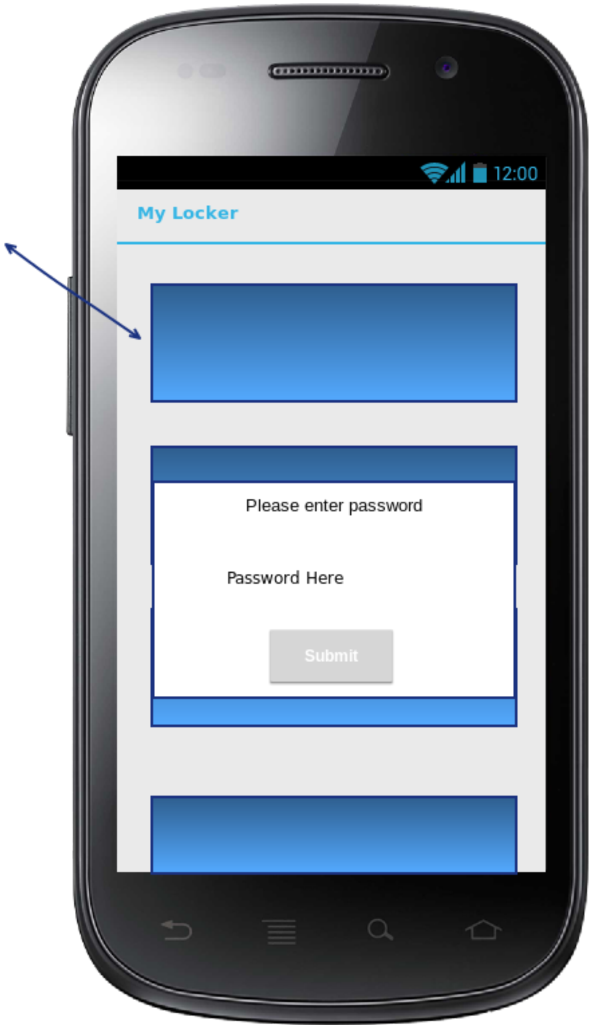
OpenGL Dice (similar to slots at a casino but for dice)

Adds another word box and slides the button to the right. Up to 10 words

FAB to start a new password create page

The OpenGL design is not finalized yet but it will be similar to slots at a casino except instead of numbers there will be dice. Pressing “Roll” will activate the animation and fill in the numbers that were obtained. The word that is associated with the rolled dice will appear where “Word 1” is written. The “ + “ button next to “Word 1” will add more word text boxes up to 10. Words will populate a ListView. Long pressing the word will allow for deletion. Re-pressing roll will re-roll the last filled in word. If the user clicks on a word then presses the Roll button it will re-roll that word. The fab button will popup a dialog to save the password in the same format as the locker entries. After saving or discarding, the screen will be refreshed to a new create screen.

My Locker (Locked)

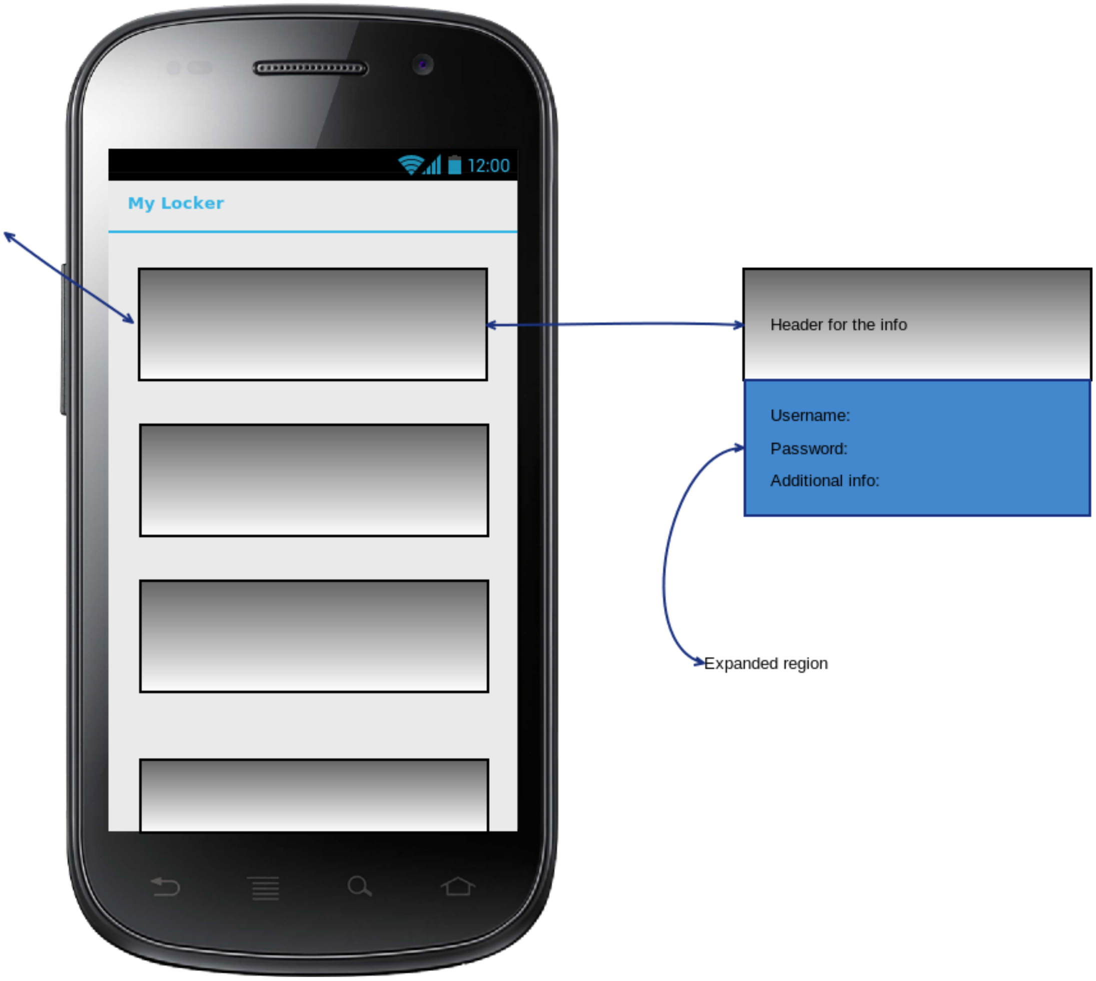


Please enter password

Shaded out regions that don't actually reveal any information. May show accurate number of blocks for the visual effect

Upon entering the “My Locker” Activity, users will be prompted to enter their password to unlock the locker. The blue boxes represent items in a ListView that contain the user’s info.

My Locker (Unlocked)



Clickable boxes that expand with additional info. The box itself will just show the header that the user can set

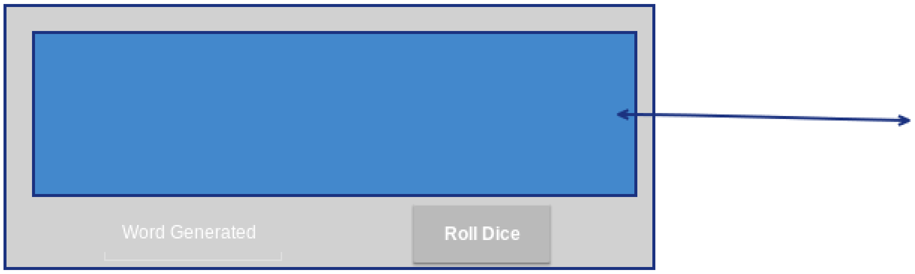
List view items will contain Header info that the user inputs when saving. OnClick will expand the list item as seen in the diagram. An edit icon will be present on the top right of the header info section when the view is expanded. Clicking it will open the original editor for that item.

Settings

Standard settings screen which will include options for:

- Setting up password for user locker (include fingerprint as option if supported)
- Select language for diceware words
- Upload / Download encrypted data to / from cloud
- Disable animations (Use png)
- About diceware (talk about password strength and cryptography)

Widget



PNG Picture format of open gl dice

OpenGL does not work with Remote view so this will be a PNG representation. It will just generate 1 word at a time using app settings for word language.

Key Considerations

Data Persistence

Data will be encrypted and stored in a database and accessed using a Content Provider. The password to decrypt the data in the database will be stored in sharedpreferences. User settings will also be stored in sharedpreferences.

Corner Cases in the UX

Pressing the back button will return to the previous view. It’s very important that the back button functions correctly or there could be some vulnerabilities with users accessing the password locker without the password. When onPause or onDestroy are called, the app will first encrypt the passwords and then return the user to either the home page or the enter password screen of the locker. The navigation bar on the top will also allow the user to swiftly navigate to any activity in the app.

Libraries Included:

- OpenGL ES for the animation of the dice because it will be cool for the user to see the dice rolling and OpenGL is natively supported by android.
- Android support library for support with older versions of Android. This will allow support for a wider variety of devices.
- Retrofit for handling api calls between the app and cloud services.
- Google play services for adMob and google analytics

Required Tasks

Task 1: Project Setup

- Set up libraries in gradle
- Set up manifest permissions

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
    - OpenGL dice roller
    - ListView for words
- Build UI for Settings
    - Details for settings options
- Build UI for Navigation drawer
- Build UI for locker
    - Pop up password entry to decrypt locker
    - Locker items
- Build UI for widget

Task 3: Implement Main Activity

- Create dice roller code
    - Do this last. Focus on functionality before aesthetics
    - Randomly generate the dice rolls and then fill in the value into the animation
- Create code to populate ListView with words
- Create add more words code
- Create save functionality
    - Pop up with the same UI as a locker item
    - Implement database and content provider for storing
        - Encrypt data with user password and then store data
- Create new password option from fab button

Task 4: Implement Navigation drawer

- Create options for:
    - Home
    - My Locker
    - Settings
    - (additional options if they are useful here such as export and upload)

Task 5: Implement Locker Activity

- Create fab to populate the ListView manually
- Populate ListView from database on initial load
- Implement edit ListView item functionality
- Create the password dialog popup with a password check. Have it lock users out of main ui if password is wrong with x amount of attempts.

Task 6: Create Settings Activity

- Create option for changing the current password on the locker
- Create option for changing language of the diceware phrases used
- Create retrofit code for api calls for uploading to various user cloud services
- Create code for disabling animations. Swap with same code the widget uses.
- Create description for about diceware and reference diceware website

Task 7: Create widget

- Widget uses already implemented roll code
- Reference sharedpreferences to know which language to use
- Use the disabled animations mode

Task 8: Set up Google Analytics

- Setup google analytics for each activity
- Track how many phrases people use for passwords
- Track which screens people use most
- Track how many people use the widget

Task 9: Set up AdMob

- Decide which activities to show a banner ad in and configure it

Task 10: Create first time use tutorial

- Explain the importance and strength of diceware in a Google app style
  - Side scroll with images
  - Say 15 languages are available with diceware passphrases
- Have users configure password for their locker
  - Explain that locker is encrypted and note if app data is wiped password data will be lost so make sure to backup your passwords!