James Kennedy
Dr. Pulimood
CSC 415
4/6/18

OSS Analysis and Design

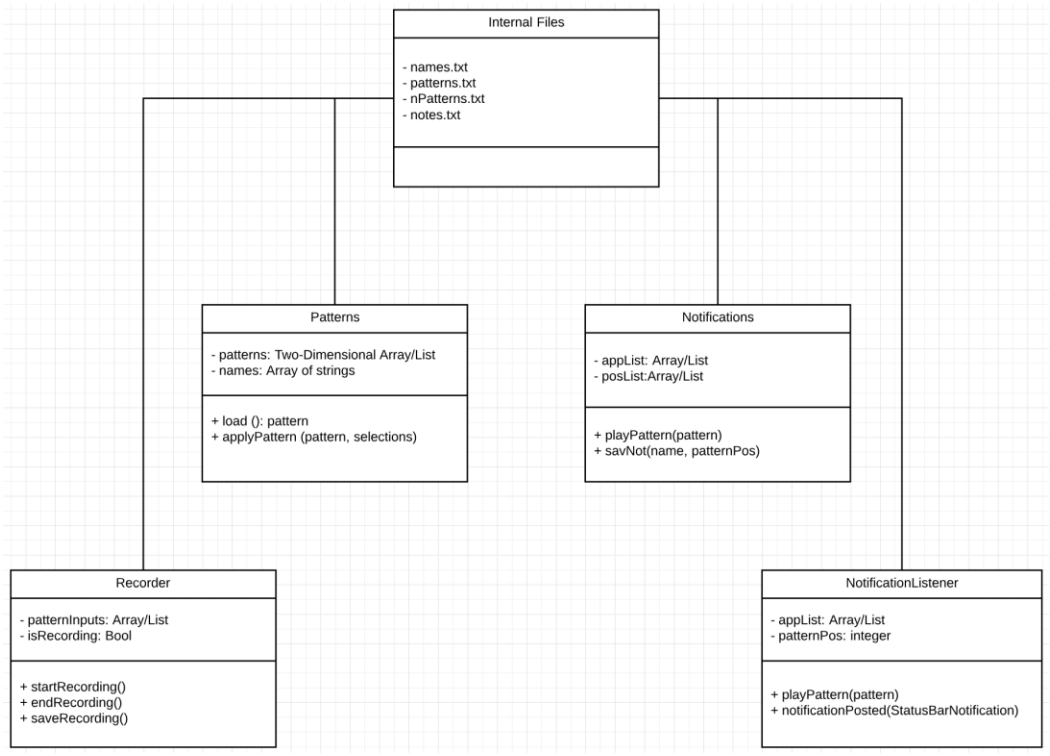
These designs are for my new app idea, an app that can record vibration patterns and apply them to different notifications. This app will help visually impaired individuals identify the notifications they receive much more easily, while only having to use touch to do it.

Use Case Descriptions:

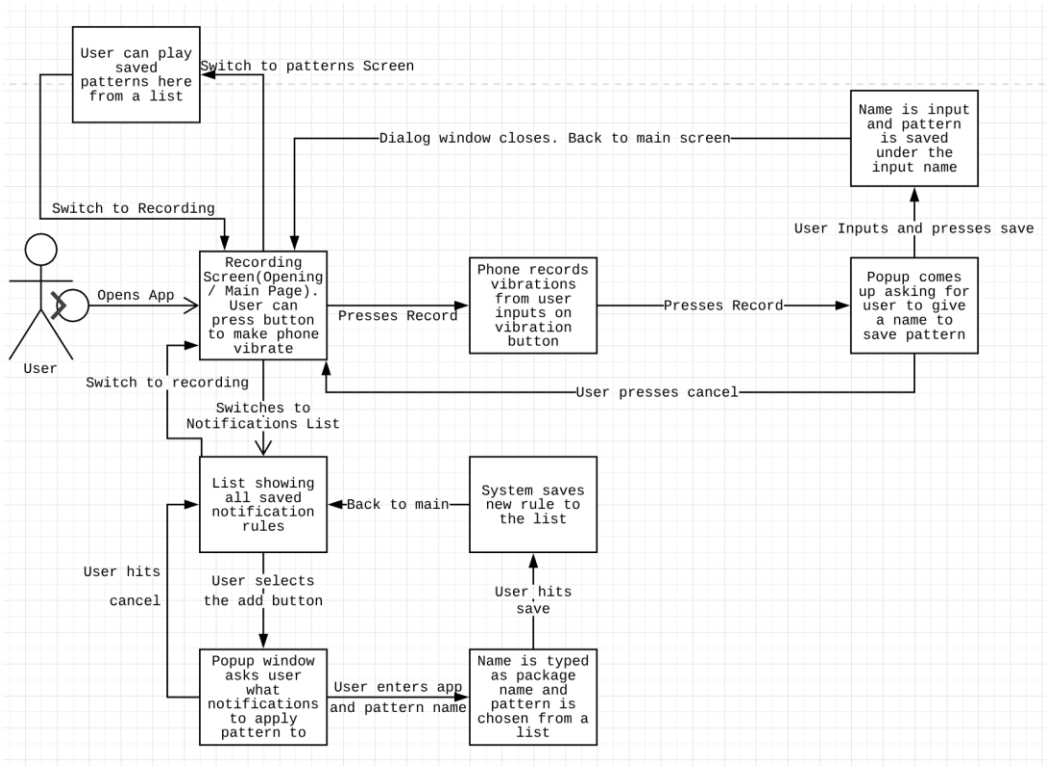
1) User wants to record a new pattern
    a) Primary actor - User
    b) Goal - create a new vibration pattern that can be saved and added to the list of available patterns.
    c) Preconditions - System must currently be on the recording window of the app
    d) Scenario
        i) User opens the app
        ii) User navigates to the recording tab
        iii) User can press or hold the vibration button on the page to experiment with patterns or practice a desired one
        iv) When ready, the user presses record to begin the recording
        v) The user presses and holds the vibrate button several times to create the desired pattern and timings.
        vi) Pop-up comes up and asks user to save the pattern
        vii) User inputs name and saves
    e) Exceptions
        i) User decides not to save pattern (hits cancel in popup)
        ii) User tries to stop recording without any vibration inputs (recording stops, no popup)
2) User wants to apply a pattern to a notification
    a) Primary Actor - User
    b) Goal - apply a vibration pattern to a certain notification or set of notifications
    c) Preconditions - System must be on the patterns list to be able to apply a vibration, and there must be at least one saved pattern to apply a vibration
    d) Scenario
        i) User opens app
        ii) User navigates to the saved patterns list
        iii) User selects a desired pattern
        iv) Popup window displays asking which notifications to apply this too
        v) User makes a selection and then presses ok to confirm
    e) Exceptions
        i) User decides not to apply the pattern (can press cancel in the popup to stop the operation)

ii) User tries to confirm pattern without selection a notification to apply it to (app displays message to make a notification selection)

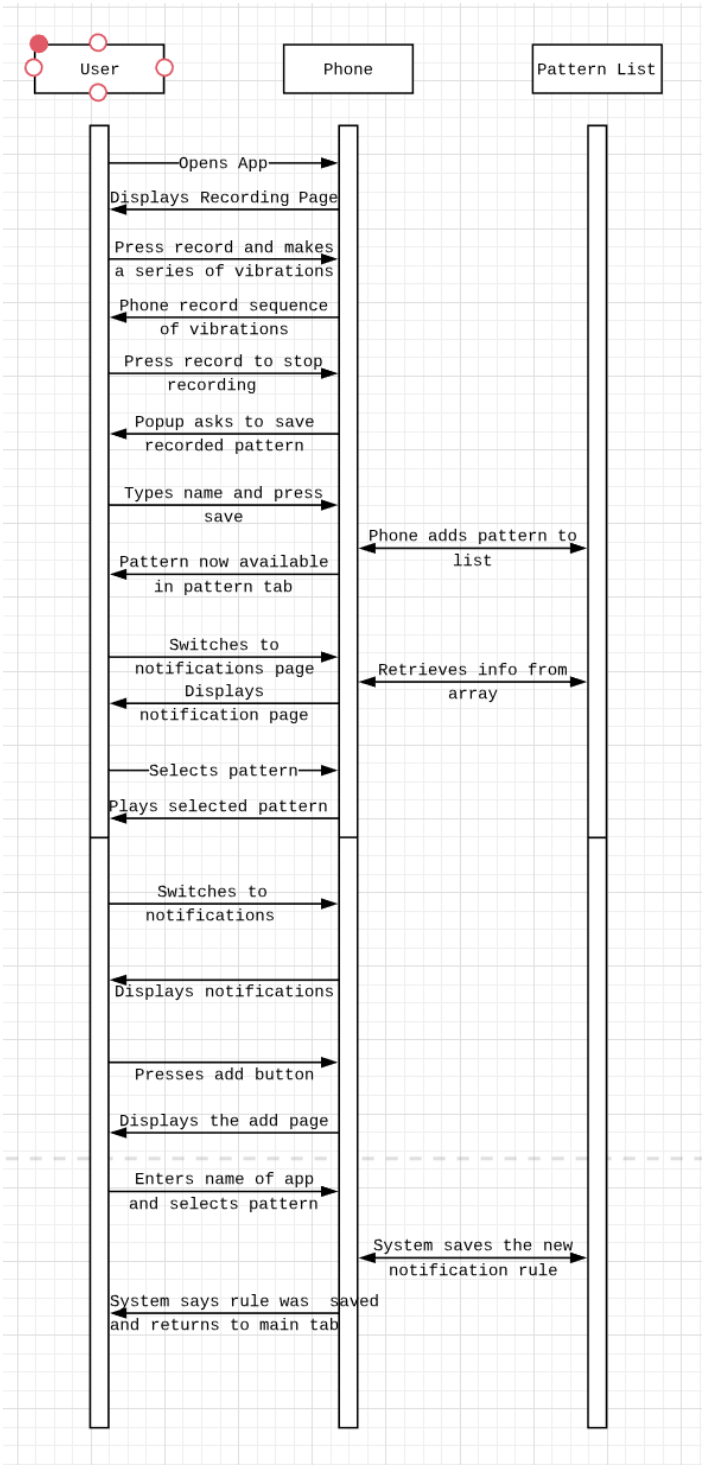
Class Diagram:



Statechart:

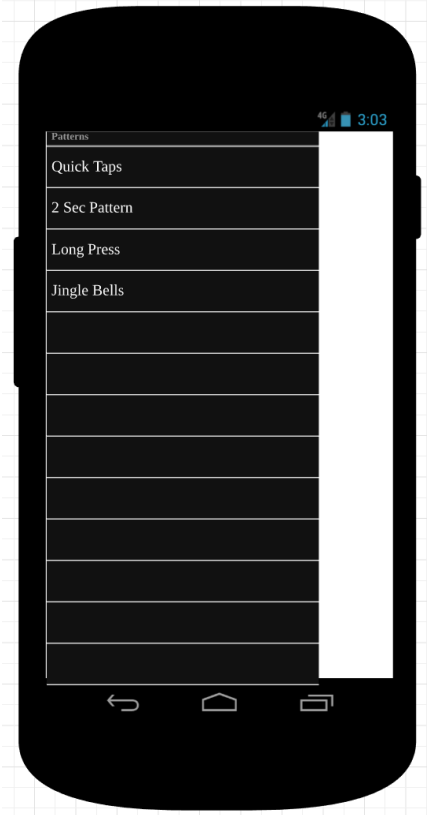


System Sequence Diagram:



UI Concept:


3:03
SixthSense
Vibrate
Record
ON
3:03
SixthSense
Please Name the Recorded Pattern
Name
Cancel
Save
Record
ON


3:03
Patterns
Quick Taps
2 Sec Pattern
Long Press
Jingle Bells

This UI abides by most of the 8 eight golden rules:
Strive for Consistency – Actions performed in the application remain consistent. The app will always receive input and then request for a name for the input and then save the input. With this system and the bottom navigation bar which is always present and allows for consistent navigation to the three main pages of the app.
Enable frequent users to use shortcuts – My app goes beyond this and allows for even non-frequent users to use shortcuts. The always present navigation bar is always present at the bottom of the window always for consistent instant navigation to the three main pages of the app no matter what window the user is currently looking at.
Offer informative feedback – Every save action includes a message that confirms that the input was saved with the input name. In addition to that, users are able to access pattern and notification rule lists and press them to test them and see if they do indeed play the correct vibration.
Design dialog to yield closure – All of my process are consistent in that they all end with the action of returning to the main page of the tab that they are working in. This means that every action comes full circle bring the user right back to where they were when they first came to the page.
Offer simple error handling – Because nearly all user input is handled through button pushed this limits the errors than can occur greatly because button presses are a very controlled input that has very little variation. All text input are stored in strings so there are never concerns for conversion issues on input numbers.
Permit easy reversal of action – This is a rule that I have not been able to implement yet. Internal storage files have strange behavior so they are a little difficult to work with, but it was what I thought to be the easiest way to allow my fragments to communicate. As of yet, removing saved inputs is not possible.
Support internal locus of control – Every action performed by the app is a response to a user input, therefore the user is always in control of the flow of the process.
Reduce short-term memory load – The application is keep consolidated by keeping 3 main tab that hold the three main functionalities of the app. There is wasted space and the layouts are kept simple and limited to a few buttons and text boxes.

This program uses encapsulation to organize the functionality into the three primary groups. Main/UI functionality, Pattern recording and Notification Management to help organize functions and architecture. This program maintains elegance by maintaining simplicity in design and implementation and approaches data by using arrays to organize information into well sorted groups.

Test Cases:

Functionality Tested	**Inputs**	**Expected output**
Recording Patterns	Given sequence of vibrations After toggling on recording	System should save and be able to playback pattern
Apply patterns to notifications	Generate various notifications	Proper vibrations with corresponding notifications
Notification tab displays proper list of saved patterns	Generated pattern list	Notification page displays complete and accurate list of patterns
Press button to make vibration	User presses large button on recording page	Phone vibrates while button is held
Save recorded pattern	Toggle off recording	Window asking for name to save under
Play vibration pattern	A vibration pattern is input to the vibrator	Phone produces correct vibration pattern