Volume Manager

Author: Chris Hare
GitHub Username: o4wcoder
Date: 2/10/2016

Contents

Description

Don’t you hate it when that one person has their phone loudly go off during a meeting or at the movies? You don’t want to be that guy, right? Well, with the Volume Manager App you won’t! Volume Manager lets you set up a time when your system volume settings will automatically change when you enter this time period. So say you have a recurring meeting from 3 pm to 4 pm every Wednesday at work; you can create a profile to repeatedly turn your volume settings to vibrate or off during that period every Wednesday so you won’t disturb you coworkers if you get a call or any other type of contact. How about having a volume profile where you phone volume turns off every night at a specific time and then comes back on when you wake up? No more annoying email buzzes or beeps in the middle of the night!

With Volume Manager you can also create a volume profile around a specific location. By selecting your location on a map, you can create a fence around your location, that when entered will change your volume settings to your desired settings when you enter that location. The volume settings will return to the previous setting when you leave the location. This feature can make sure your phone is quite every time you enter your church or school for instance. You can also search for any location on the map and create your volume settings at locations even if you are not currently there.

The convenient widget will allow you to fully control your volume profiles from your home screen, or simply see if you are currently under one of your pre-defined volume profile’s control.

Intended User

Volume Manager could be used by any one, but it greatly benefits people that routinely go to places where they should silence their phone like office workers and students.

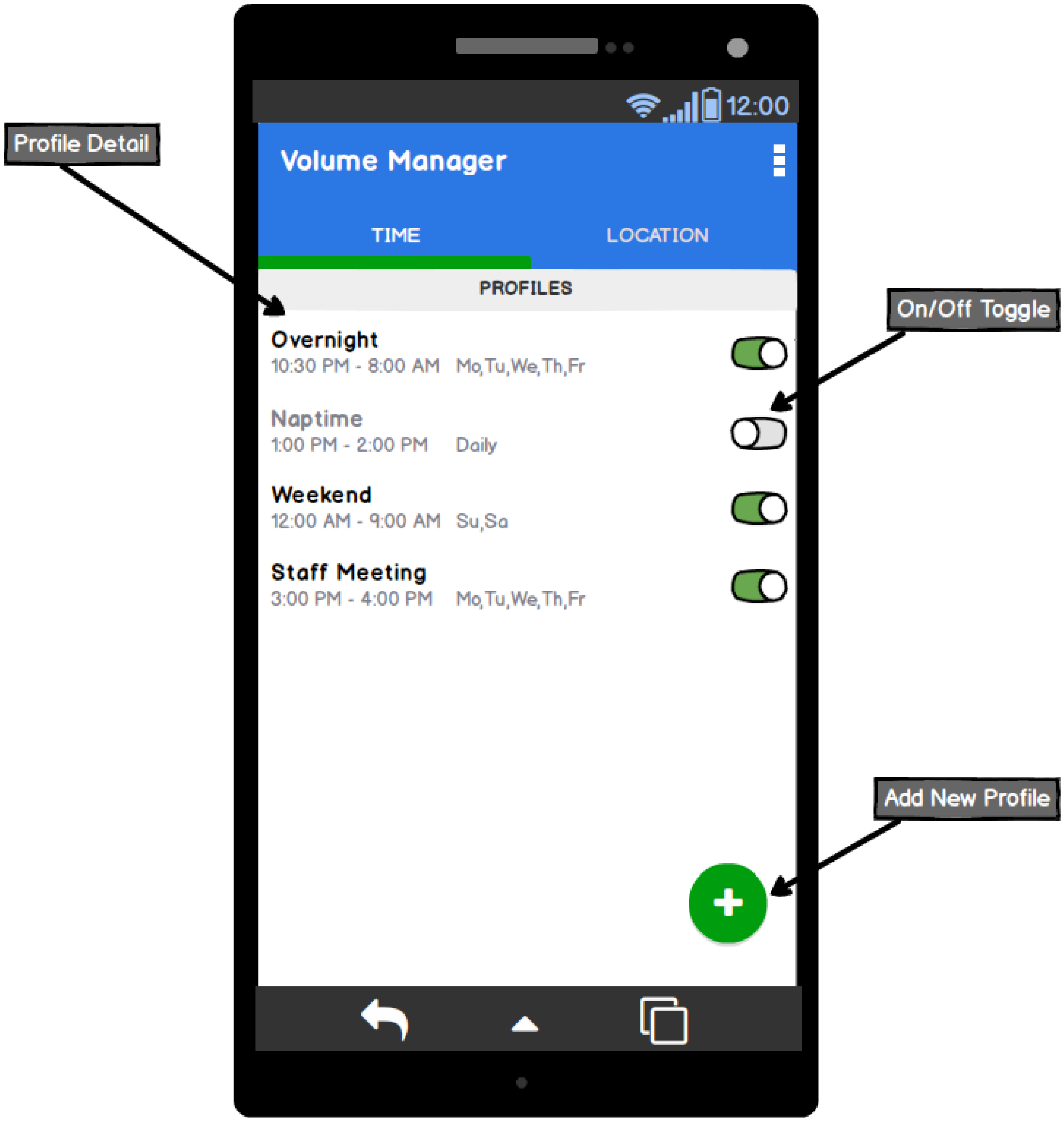
Features

- Creates volume controls for pre-defined time intervals, where the volume will change to defined settings when entering this time interval.
- Creates volume controls for pre-defined Locations, where the volume will change to defined settings when entering a specific radius of that location.
- All Volume Profiles created are saved locally and are persistent upon reboots.
- The volume control is for system level controls only. It does not handle individual volume settings such as Media and Notifications.

User Interface Mocks

Time Profile List

The main screen contains a TabView containing the two types of Volume controls profiles; Time and Location. Here is an example of the Time Profile List. It contains a list of all created Time Profiles which show the time range and days of the week it is activated. It also has a switch to turn the control on/off. Note, this switch component may be changed into an animated icon to toggle the state of the time profile. A FloatingActionButton at the bottom right is used to create a new profile. Also, a shared element transition will be used for the title of the profile between the list and the Time Profile Details seen in the next mockup.



Overnight
10:30 PM - 8:00 AM Mo,Tu,We,Th,Fr

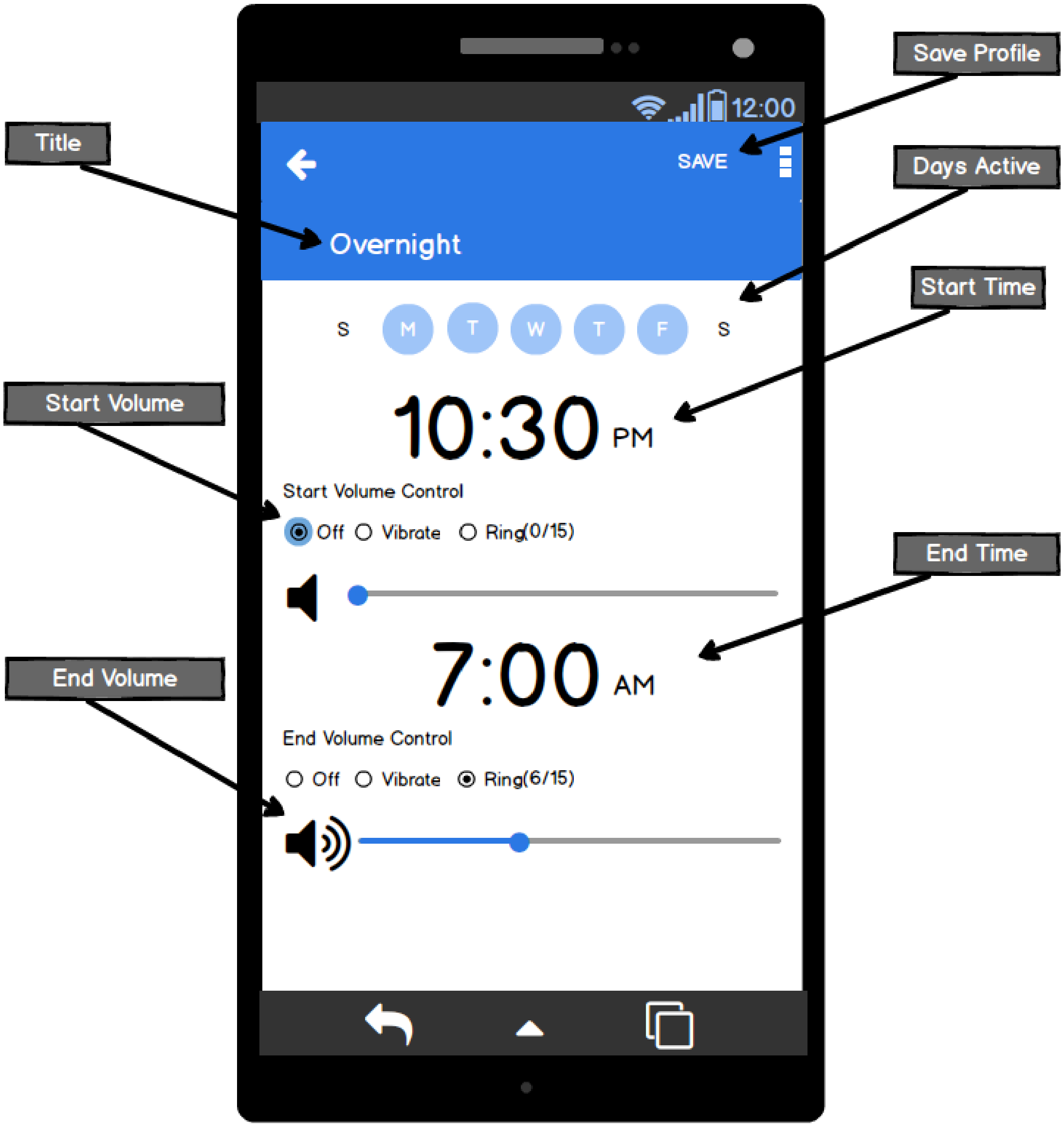
Naptime
1:00 PM - 2:00 PM Daily

Weekend
12:00 AM - 9:00 AM Su,Sa

Staff Meeting
3:00 PM - 4:00 PM Mo,Tu,We,Th,Fr

Time Profile Detail

The Profile Details screen is used when a new Time Profile is created or an existing one is selected. Here you create the settings for the volume controls of that profile. From the top going down, we start with the title of the profile. In this example it's "Overnight". It can be entered here in the text box. There must be a title before the "SAVE" button on the top of the Toolbar is activated. Next we have the days of the week, using custom on/off toggle buttons, the volume control will be triggered. Below that is the time the volume controls starts, followed by the type of volume control and then the ring volume (if ring is used). These controls are repeated next with the end of the volume control settings.



Overnight

S M T W T F S

10:30 PM

Start Volume Control

● Off ○ Vibrate ○ Ring(0/15)

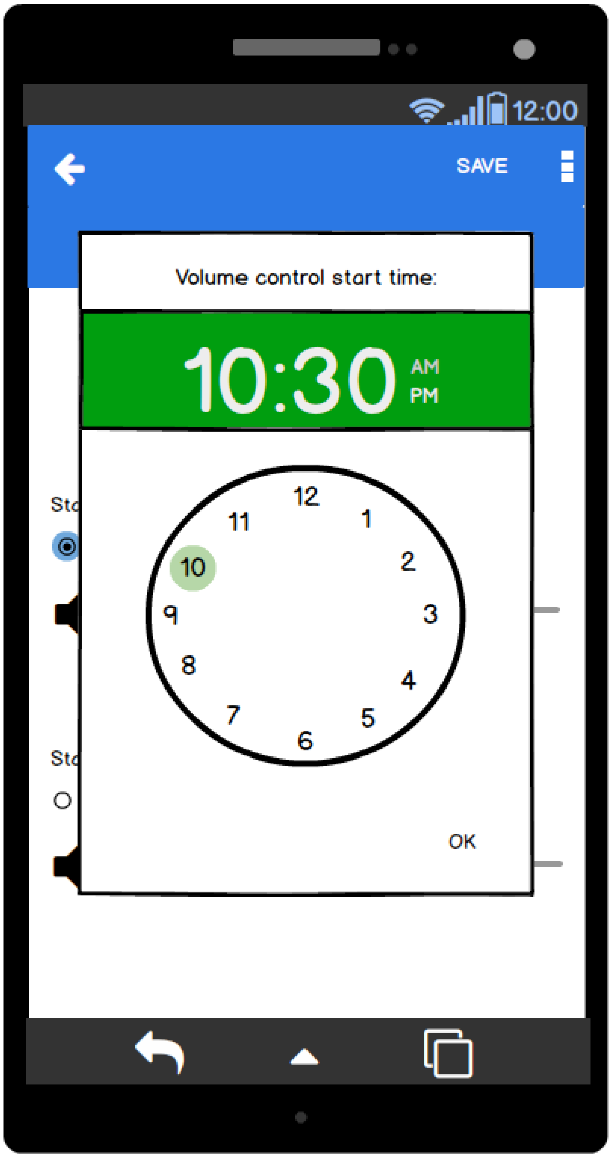
7:00 AM

End Volume Control

○ Off ○ Vibrate ● Ring(6/15)

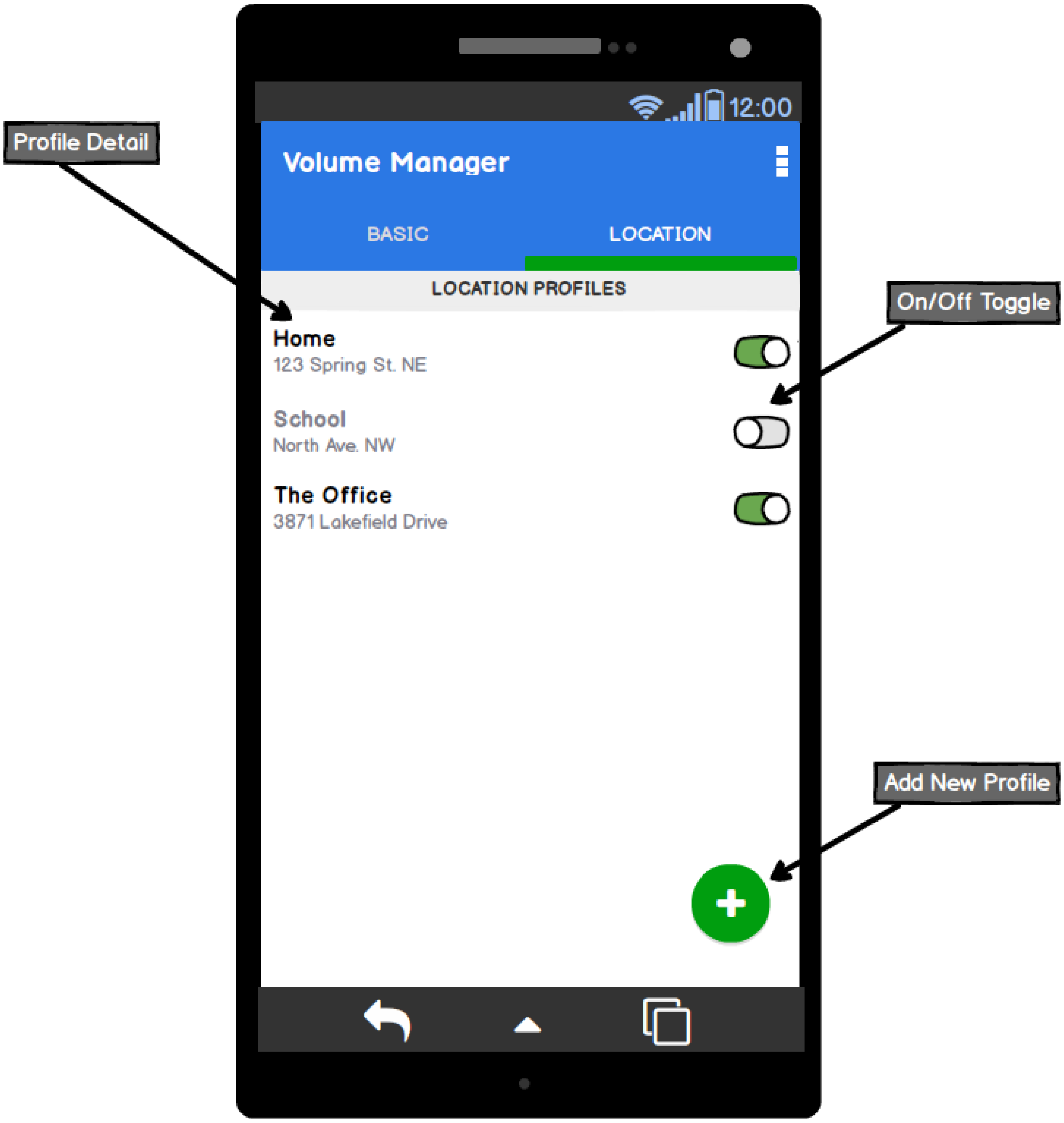
Edit Volume Control Time

After selecting the start or end times, a TimePicker Dialog is displayed to set the time.



Location Profile List

The second tab on the main screen is for Location Profiles. It contains a list of all created Location Profiles which show the address where the volume control will be triggered. It also has a switch to turn the control on/off. Note, this switch component may be changed into an animated icon to toggle the state of the time profile. A FloatingActionButton at the bottom right is used to create a new profile. Also, a shared element transition will be used for the title of the profile between the list and the Time Profile Details seen in the next mockup.



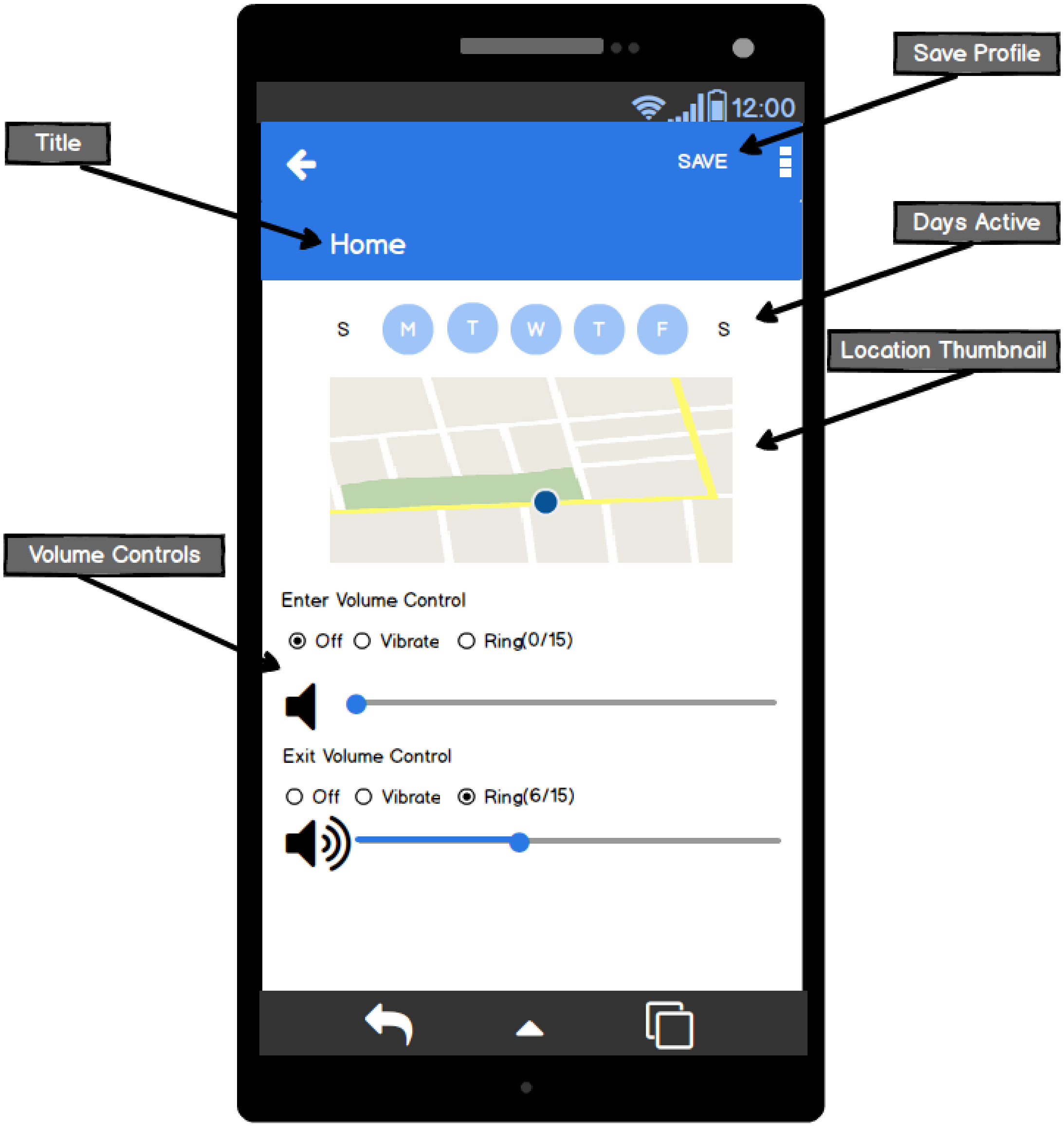
Home
123 Spring St. NE

School
North Ave. NW

The Office
3871 Lakefield Drive

Location Profile Detail

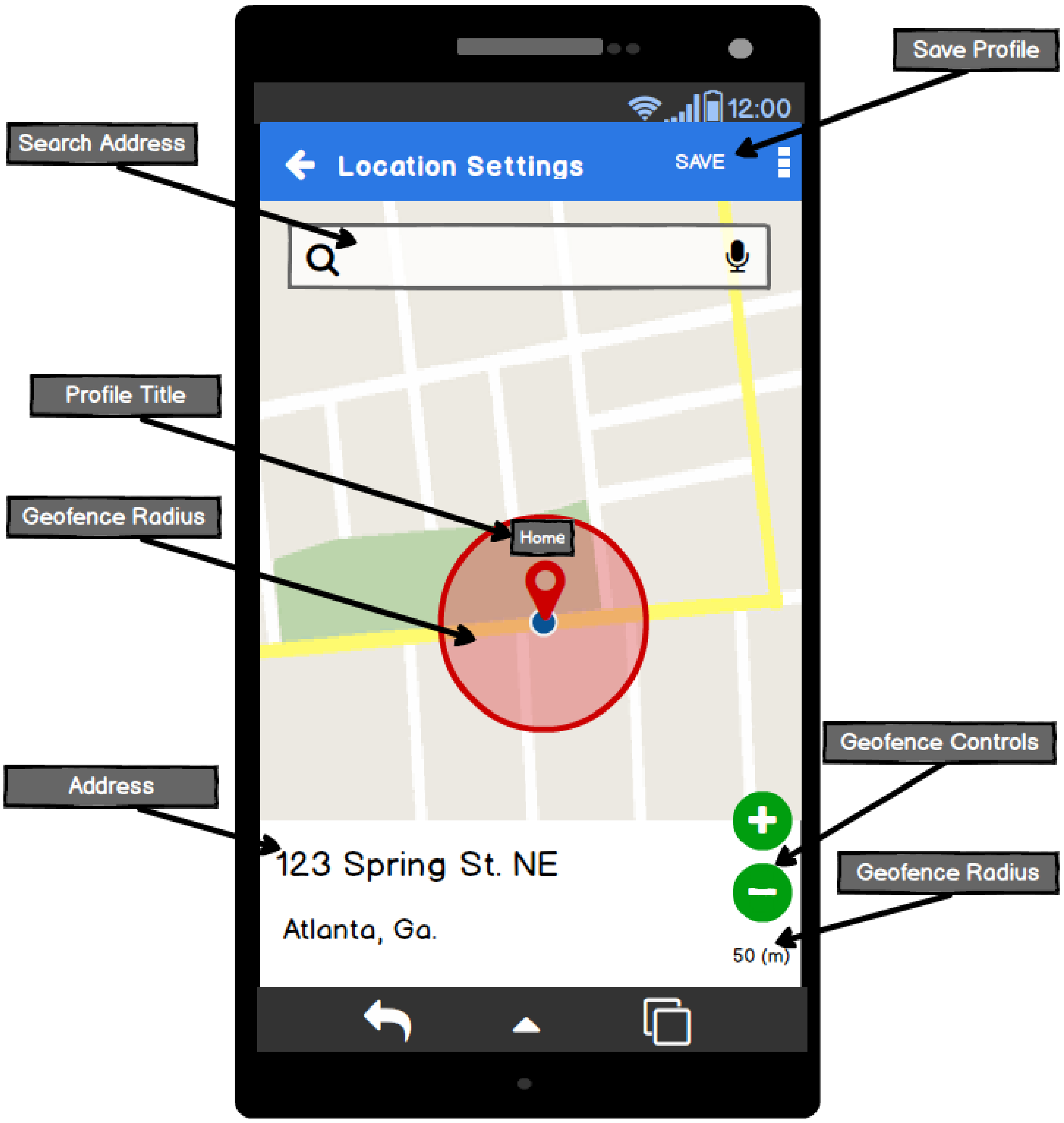
After adding a new location profile or selecting a location profile from the list, you will get the Location Profile Detail screen. Here you create the settings for the volume controls of that profile. From the top going down, we start with the title of the profile. In this example it’s “Home”. It can be entered here in the text box. There must be a title before the “SAVE” button on the top of the Toolbar is activated. Next we have the days of the week, using custom on/off toggle buttons, the volume control will be triggered. After that is a thumbnail map of the location in the profile. Selecting it will bring you to the Location Map Activity seen in the mockup after this. Below that are the location enter volume controls, followed by the type of volume control and then the ring volume (if ring is used). These controls are repeated next with the location exit volume control settings.



Location Map

The location map screen is a Google Map that lets you select the location to set a volume control. When the screen is shown after creating a new location profile, it will show your current location on the map. If the current location is not what you want, you can search for an address in the search bar at the top using the Google Places API. They map will then zoom to that location.

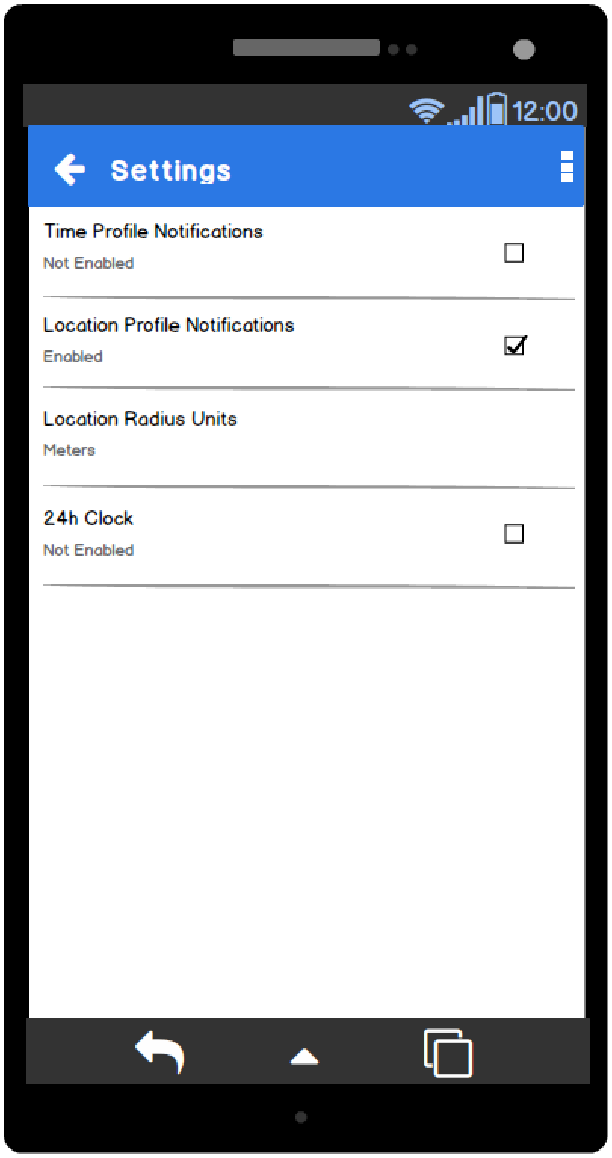
A marker is used to show the location. The title give to the location profile will be displayed above it. The user can press anywhere on the screen to have the marker move to a new location. A red circle around the location will show the radius of the Geofence around the location. The radius can be increased or decreased by the controls on the bottom right of the screen. The default radius is 50 meters.



Preference Settings

The Settings menu can be reached on any Screen from the overflow menu on the top right of the Toolbar. The values here will be stored in SharedPreferences and will consist of the following settings.

- *Time Profile Notifications*: Turn notifications on/off for Time Profile start and end times. When enabled, each profile will get two notifications for start and end times. Default is Disabled.
- *Location Profile Notifications*: Turn notifications on/off for Location Profile enter and exit Geofence boundary. When enabled, each profile will get two notifications for enter and exit boundary events. Default is Disabled
- *Location Radius Units*: Change the units of the Geofence radius, around a location from meters to feet. Default is meters.
- *24h Clock*: Use a 24 or 12 hour clock. Default is Disabled.



Time Profile Notifications
Not Enabled ☐

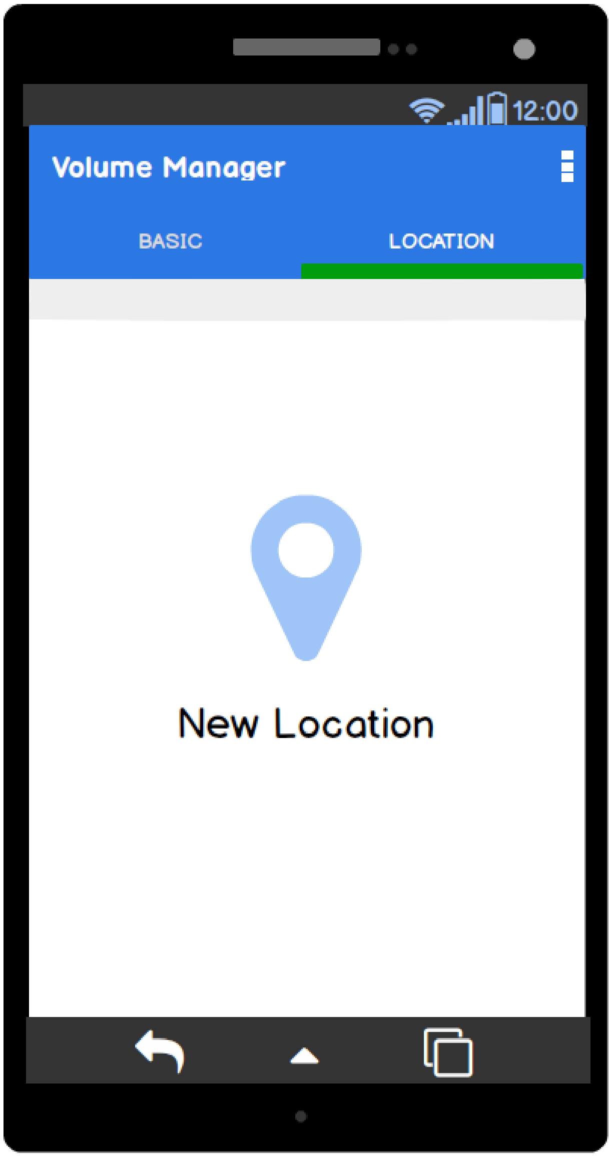
Location Profile Notifications
Enabled ☑

Location Radius Units
Meters

24h Clock
Not Enabled ☐

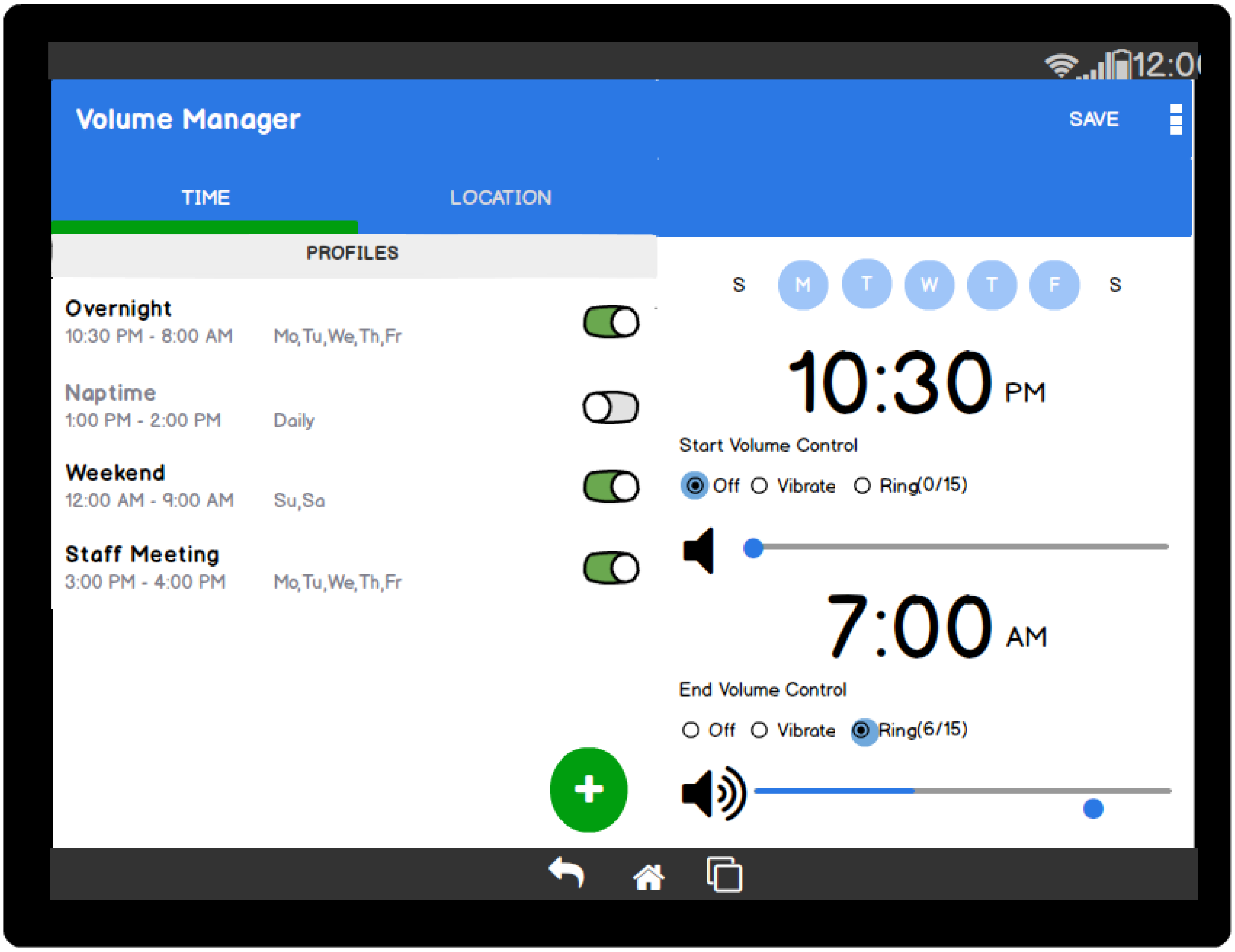
New Profile

When a profile list is empty the “New Profile” screen is displayed. This is taken care of by the library described in the “Libraries” section below. When selected it will bring you to the Profile Detail page. Below is a screen of the “New Location” profile screen



Tablet Layout

The tablet UI will follow the basic master/detail layout; with the list of profiles on the left and the detail of the selected profile on the right. The final look will be a bit different than this mockup as Material Design principles are applied.



Overnight
10:30 PM - 8:00 AM Mo,Tu,We,Th,Fr

Naptime
1:00 PM - 2:00 PM Daily

Weekend
12:00 AM - 9:00 AM Su,Sa

Staff Meeting
3:00 PM - 4:00 PM Mo,Tu,We,Th,Fr

S M T W T F S

10:30 PM

Start Volume Control

● Off ○ Vibrate ○ Ring(0/15)

7:00 AM

End Volume Control

○ Off ○ Vibrate ● Ring(6/15)

Key Considerations

How will your app handle data persistence?

Data for each volume profile will be handled with a Content Provider. There will be one table for the main volume control settings. And a second table containing Location information.

Preference Menu settings will be stored in shared preferences. These will be the required settings.

- Enable Notifications for Time and Location start/end volume settings.
- Geofence Unit Measurements (meters or feet).
- 24 or 12 hour clock.

Describe any corner cases in the UX.

Both the Time and Location Profile's require the user to add a title before the SAVE button becomes enabled. Profiles with no title will not be allowed to be saved in the Database.

Libraries

For this project I’ve decided to create my own library that contains a custom view. This will take care of both the Library requirement as well as the optional custom view requirement.

The purpose of this library is to create a custom view for the “Empty” state of both the Time and Location ListViews. When shown it will contain an image and a message that says to create a “New Profile”. See the section “New Profile” above.

For API 5.0 and higher, the clickable view will contain a Vector Drawable animation. When selected an animation will trigger and the Time or Location Profile Detail UI will be loaded (See Mockups above). The animation will be a quick fade and spiral. My goal is to create several different types of animations that the user can select when using the custom view in their XML Layouts. However for this project, and time requirements, only one type of animation will be done.

For APIs lower than 5.0, a simple image will be used as the clickable view. No animation.

Required Tasks

Task 1: Project Setup

- Install libraries for Google Play Services in build.gradle file and AndroidManifest.xml
- Get and setup required Google API Keys
- Create Profile Model class. This will contain all the settings for a single volume control profile.
- Create Location Model class. This will contain all the location details.
- Create Content Provider
- Create AndroidTest functions to test the Content Provider. Verify it’s working before proceeding.

Task 2: Implement UI for Main Activity and Time Profile Detail Fragment

The main Activity will consist of two tabs. One for Time profiles and the other for Location profiles. Each tab will contain a ListView that will display the list of profiles that have been created. A PagerView will connect the two tabs

Tasks:

- Create MainActivity UI that contains two tabs and a PagerView

- Add ListView for the Time and Location Profiles. They will share the same ListView, even though they are on different tabs. However, they will require separate adapters.
- Create the UI for the Profile Detail Fragment. This is where you will set the start and end times of a profile along with the volume settings at each point
- Link Content Provider to Main Activity and Detail Fragment. The Main Activity will have the ability to add a new Profile Row to the Content Provider and the Delete a Profile Row. From the Detail Fragment you can add, delete and update a Profile row.
- Adapt UI for Tablets. Create Main/Detail UI for Landscape Mode using Material Design Elements

Task 3: Create Volume Control Backend Services and Receivers

The Backend of the app will handle the changes in the volume settings. Each volume profile will have a start and stop event for changes in the volume. This will be handled by the AlarmManager class. Each start and stop times will be registered with the AlarmManager to be fired off when the times are reached. When they are fired off, they will be handled in the background by an IntentService. It is here in the IntentService's onHandleIntent() method that the volume settings will be changed. Also here is where notifications will be sent if enabled.

Tasks:

- Create IntentService to handle the start and stop times of the volume profile
- Create AlarmManager that will register all the volume profiles start and stop times and register with the IntentService.
- Change volume settings, using the AudidoManager when IntentService's onHandleIntent() is called.
- Send out notification of change in volume. This setting will always be on until the Preference Settings Menu is created to be able to turn it off.
- Use a BroadcastReciever to reset all the stored profile's start/end times in the AlarmManager.

Task 4: Implement Location UI

Tasks:

- Create Google Maps Activity
- Pull thumbnail of current/stored location to show on Location Detail UI.
- Adapt UI for Tablets. Create Main/Detail UI for Landscape Mode using Material Design Elements

Task 5: Implement Maps and Geofencing Backend Services

Tasks:

- Add Map Markers to Map when the user presses on a location. Have the title of the profile, given by the user, displayed above the marker.
- Create and IntentService to handle enter/exit of Geofence. Put Geofence code in it's on class (GeofenceManager) so it can be accessed from not only the Activity with the Map, but from the Activity with the list of locations.
- Use BroadcastReceiver to turn back on Geofences when the phone reboots.

Task 6: Implement Google Places API

The user will not only be able to set up Location Profiles by selecting a spot on a map, they will also be able to search for any address using Google Places API

Tasks:

- Create search bar at the top of the map for entering and address
- Implement backend of Google Places API

Task 7: Create Custom Library

Tasks:

- Create custom view library.
- Custom view contains a vector drawable animation for APIs 5.0 and greater
- Create static image for APIs lower than 5.0
- View is clickable and title on custom view programmable.

Task 8: Create Custom Widget

There will be two widgets for this app. One small widget to just show if the user is currently under volume control, and a large widget that will contain the list of time and location profiles.

Tasks:

- Create small widget to show if the user is currently under volume control.
- Create large widget that will contain the list of time and location profiles.

Task 9: Final Features and Miscellaneous Items

This section includes and final tasks to make the app production ready.

Tasks

- Create Preference Settings menu
- Verify all strings are located in the strings.xml file and uses RTL
- Create Content Descriptions on relative components on the UI

Task 10: Get App Ready for Release

All tasks that should be done in the final stage for release to the Google Play store.
Tasks:

- Create Free and Paid Versions. Add Ad to the bottom of the MainActivity under the Profile ListView
- Create App Icon
- Create Signed APK for App
- Create Free and Paid versions and release each to the Google Play Store.

Optional Features

These are features that are optional and may be implemented on this first phase if time permits. If not, they will be moved to Phase II development.

Tasks:

- Extend to Wearable
- Create Intro Tutorial Screens
- Create a third type of profile called “Calendar”. This will link to your Google Calendar and perhaps and Exchange Calendar, so you can set volume controls for your meetings and events.