GitHub Username: regmoraes

Closer

Description

Sometimes we forget of tasks that we've to do on our way. Have you ever missed the chance to buy or do something yet you were in the right place at the right time? Yeah, sometimes we just forget things. But the Closer app is here to help you.

With the Closer app you can create reminders for tasks that you have to do on a location. Do you need to go to the market on your way home? Do you need to buy some drinks on your way to the party? No problem, just create a reminder, add a description and mark the the desired location and when you're close to it, the app will notificate you!

Intended User

This app if for any type of user.

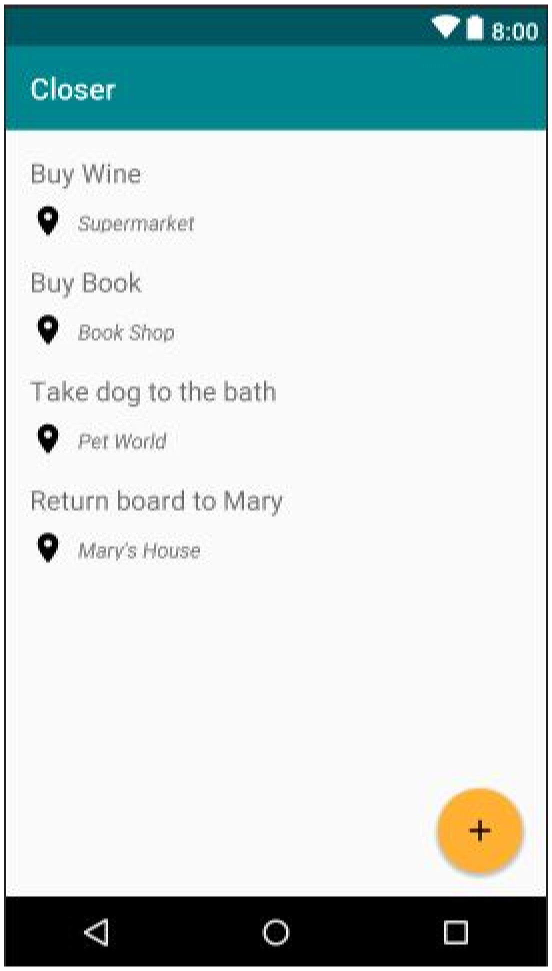
Features

List the main features of your app. For example:

- Create and edit reminders
- Search places for a reminder
- Add a place to a reminder
- Notificate the user as it passes nearby the desired location
- Option to start navigation to a place when available

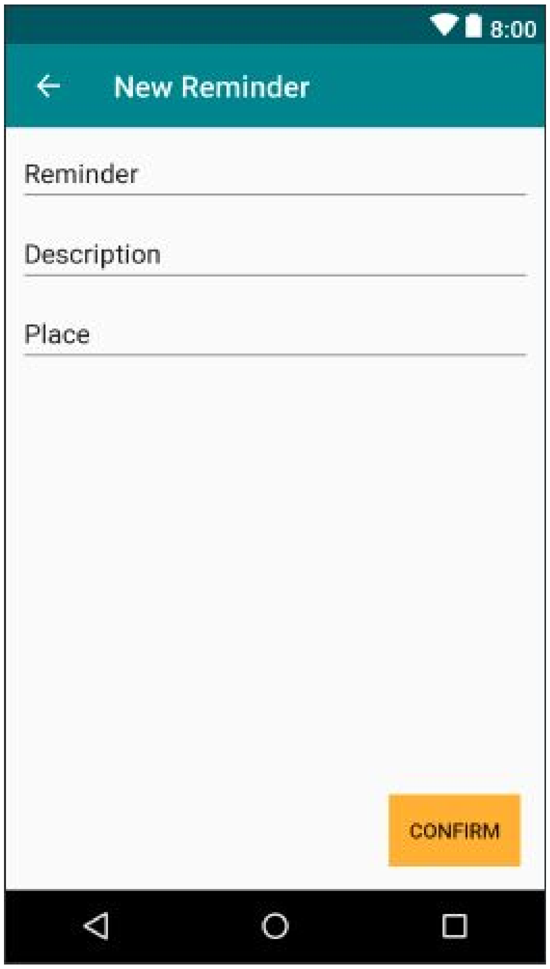
User Interface Mocks

Screen 1 - Reminders List (Main Activity)



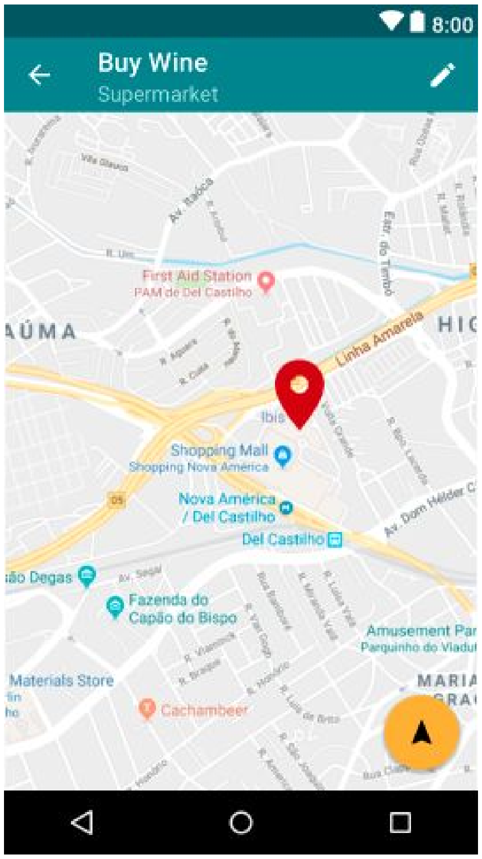
This screen shows all reminders. For each reminder item it shows a title and the location associated with the reminder.

Screen 2 - Add/Edit Reminder



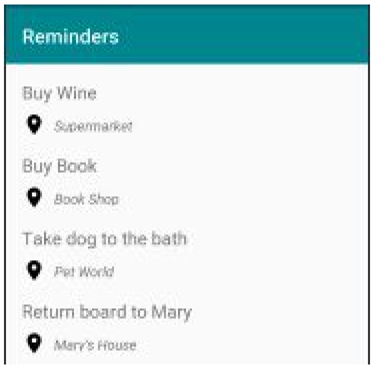
In this screen the user can add or edit a reminder. The places field offers is an autocomplete edit text that shows places provided by the Google Places API.

Screen 3 - Reminder Details



In this screen the user can see the reminder details. The screen shows the reminder detail and location title in the toolbar and a marker on the map. Also, it shows a FAB button that allows users to start navigation to the location (when available).

Screen 4 - Reminders Widget


Reminders

Buy Wine
Supermarket

Buy Book
Book Shop

Take dog to the bath
Pet World

Return board to Mary
Mary's House

The widget will show a list of current set reminders.

Key Considerations

The app will be developed using the Java programming language

How will your app handle data persistence?

All data will be accessed and stored in a SQLite database and exposed via a Content Provider.

Describe any edge or corner cases in the UX.

- [Screen 1] When the user clicks in a reminder, it will show the reminder detail (Screen 3)
- [Screen 1] When the user clicks in the '+' ***FAB*** it will show the add/edit reminder screen (Screen 2)

- [Screen 2] When the user clicks in the ***arrow back icon*** the app will return to the reminders list screen (Screen 1).
- [Screen 2] When the user clicks in the ***confirm button,*** if all field are filled, it will create a reminder and return to the Screen 1. If any field is empty it will show messages for each empty field.

- [Screen 3] When the user clicks in the ***edit icon*** it will open the add/edit screen (Screen 2) with all field filled with the reminders info.
- [Screen 3] When the user clicks in the ***navigation FAB*** the app will try to open the a navigation app to show the route from the current user location to the location set in the reminder.

- [Screen 4] When the user clicks on a reminder from the widget, it will open the reminders detail (Screen 3).

Describe any libraries you'll be using and share your reasoning for including them.

- The app will use the latest stable version (as May 2018) of the Android Studio (3.1.2) ,Gradle Plugin (3.1.2), Gradle (4.4-all) and Google Services plugin (3.3.0). Also, it will use the following libraries:

Library	Version	Purpose
Dagger 2	2.16	Manage dependency injection

Android Databinding	Automatically picked by
	
	`dataBinding {`
	`    enabled = true`
	`}`
	
	based on the gradle plugin (3.1.2)
		Help the definition and access to UI elements

Google Play Maps Library	15.0.1	Show a map and track location
Google Play Location Library	15.0.1	Geofencing
Google Play Places Library	15.0.1	Offer a list of available places to add in the reminder

Describe how you will implement Google Play Services or other external services.

- The Maps library will be used to show a MapView fragment in the ***Screen 3***
- The Places library will be used to offer a list of available places to add in the reminder ***Screen 2***
- The Location library will be used to create and track Geofencings for reminders location and correct notification based on location. This feature will be implemented using a **PendingIntents** and **IntentServices** to create Geofencings for reminders in the background and to listen for Geofencing transitions and creation of notifications.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Configure Google Play Services APIs
- Configure libraries dependencies
- Configure release keys and password

Task 2: Project Core Development

- Configure package structure
- Create core components and "managers"
- Create repository (database and content provider)
- Create Unit Tests for the core features

Task 3: Implement UI for Each Activity and Fragment

- Create Layout and build UI for RemindersActivity
- Create Layout and build UI for RemiderDetailActivity
- Create Layout and build UI for CreateEditReminderActivity

Task 4: Glue All Things

- Create dependency providers, components and modules

Task 5: Create Instrumentation Tests

- Create Android Instrumentation Tests for RemindersActivity
- Create Android Instrumentation Tests for ReminderDetailActivity
- Create Android Instrumentation Tests for CreateEditReminderActivity

Task 6: QA Tests

- Test the overall application in real cases