How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

GitHub Username: mcnewbk

Remindr

Description

Remindr allows users to create fast and easy to do lists. These lists will allow the user to attach images and text to the list items to spark the user’s memory and will utilize the phone’s internal notification system to do so. You can set a due date, which will then fire a notification for the user to see that the item is due or is coming up.

Intended User

The intended user is any user that can benefit from a reminder app in the format of a to-do list. Examples range from stay at home moms to college students and even to CEO's of a company.

Features

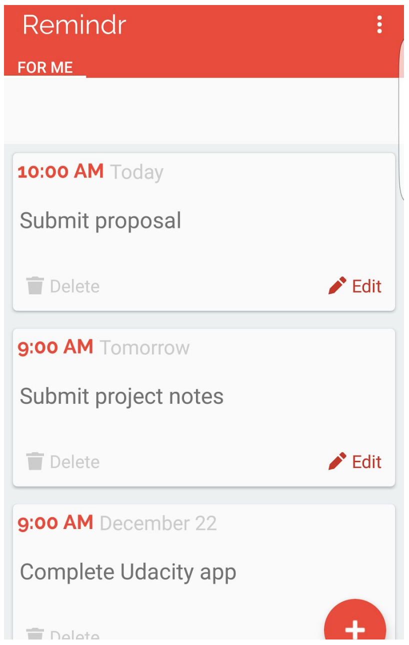
List the main features of your app. For example:

- Saves list of reminders for the user
- Allows users to attach text and images for the notifications
- Gives a clean and easy way to add items to the list
- Allows users to change settings for how the app should handle notifications.

User Interface Mocks

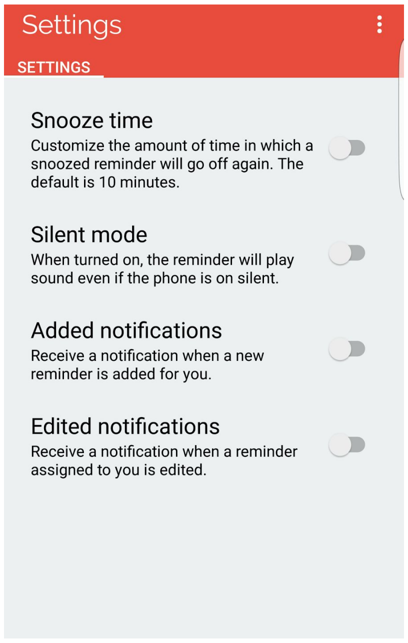
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



This is the home or main screen. This will have a CardView of all the items in your list with the image, text, dates, and times for the reminder to-do card. The option to add a new reminder is represented by a FAB on the bottom right hand of the screen. Settings access via the icon in the top right hand of the screen as well.

Screen 2


Settings

SETTINGS

Snooze time
Customize the amount of time in which a snoozed reminder will go off again. The default is 10 minutes.

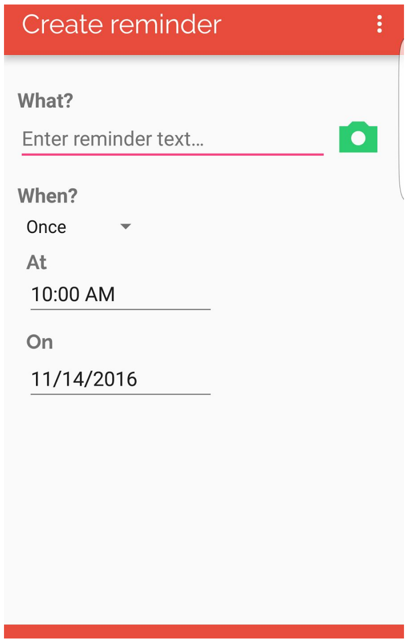
Silent mode
When turned on, the reminder will play sound even if the phone is on silent.

Added notifications
Receive a notification when a new reminder is added for you.

Edited notifications
Receive a notification when a reminder assigned to you is edited.

This is the settings screen that dictates how notifications will work throughout the application. There are few settings already programmed into my application already!

Screen 3


Create reminder

What?

Enter reminder text...

When?

Once

At

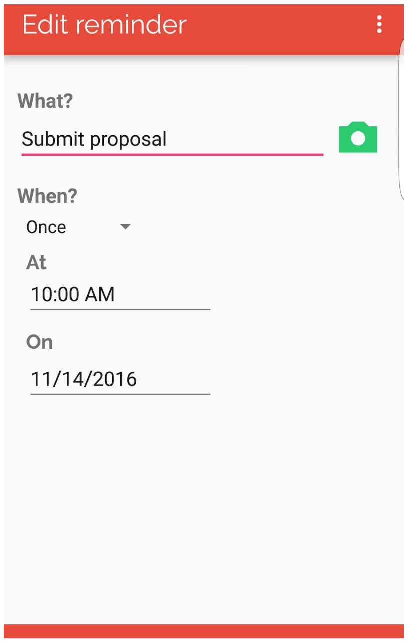
10:00 AM

On

11/14/2016

This is the create reminder screen. Here we will be determining what the reminder is, whether we would like to attach an image, and when it should go off. Pressing save at the bottom will take you back to the main page with the new reminder.

Screen 4



Edit reminder

What?

Submit proposal

When?

Once

At

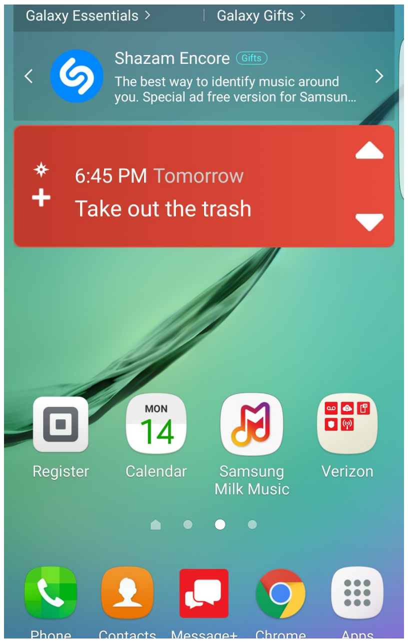
10:00 AM

On

11/14/2016

This is the edit reminder screen. This will allow you to change the fields of the reminder to best suit your needs. Saving or back will take you to the main screen.

Screen 5



6:45 PM Tomorrow

Take out the trash

This is the Remindr widget. It will show the top five reminders inside your list. If you click the plus button, it will open the app to the Create Reminder screen.

Key Considerations

How will your app handle data persistence?

My app will be using local storage, Sugar DB, and will be using the hardware's internal system to schedule the reminders.

Describe any corner cases in the UX.

I can not think of any current corner cases at the moment. Perhaps how to add to the list if a user inserts a new reminder between existing ones and how to elegantly handle that addition.

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

For example, squareup Picasso, several github repos for certain elements of the app(BindingCollectionAdapter, PhotoView, and FloatingActionButton), Raygun for error handling, Joda time for help with times and dates, and many others that will be outlined in the project.

Describe how you will implement Google Play Services.

I am currently importing Google Play Servcies 8.3.0 in my application, but not using it at the moment. I really don't see a use for it with the app that I envision. If I do end up using it, I can update this section of the document.

Next Steps: Required Tasks

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

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Configure libraries
- Setup db
- Declare models
- Setup UI and draw it out

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for new reminder activity
- Build UI for the reminder that shows on the screen
- Build UI for the settings page
- Build UI for the create and edit reminder page.

Task 3: Implement the Notification Service.

Describe the next task. List the subtasks. For example:

- Create the notification service
- Implement methods that will save reminders locally for the time given

Task 4: Testing app and timed reminder

Describe the next task. List the subtasks. For example:

- Go through the app and test
- Fix any issues that come up

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”