How to Use this Template
1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: adodge1

KidsSummer organizer

Description

KidsSummer organizer is an app that helps you keep track of your kids summer plans. It will help you organize their summer camp weeks per child.
Save-to-organizer option makes it easy to remember what's going on for each camp day.
It will keep track for you of the following information:
- If you must send child with lunch and snack or lunch.
- Where they are going that day of the week and the outfit they should wear.
- The camp's address and drop off and pick up hours.
- The camp's emergency contact person. Email and phone.
- The camp's paperwork has it been filled out or not.

Add-to-favorites option makes it easy to remember for next year if you child liked the camp and wants to be registered again next year.

Intended User

This is app is for parents.

Technical Features

This is app will be coded only in java programming language.
This app will pull and send data to/from firebase on a per request basis this app uses an IntentService to do this request.
This app also performs short duration, on-demand requests, app uses an AsyncTask for this.
This App will keep all strings in a strings.xml file and enables RTL layout switching on all layouts.

This App includes support for accessibility.

This app will use the following library versions:

Gradle	4.1 or later
Android Studio	3.4 or later
com.google.firebase:firebase-database	16.1.0 or later
com.google.firebase:firebase-core	16.0.8 or later
com.google.gms:google-services:	4.2.0 or later
com.android.support:support-v4	28.0.0 or later
com.android.support:appcompat-v7	28.0.0 or later
com.android.support:design	28.0.0 or later
com.android.support:palette-v7	28.0.0 or later
com.android.support:recyclerview-v7	28.0.0 or later

compileSdkVersion	28 or later
targetSdkVersion	28 or later

Features

- Saves information
- Integrates with Google Maps
- Will try to do reminder to drop off and pick up.

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 Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1

KidsSummer Organizer

Your Children   Add Child

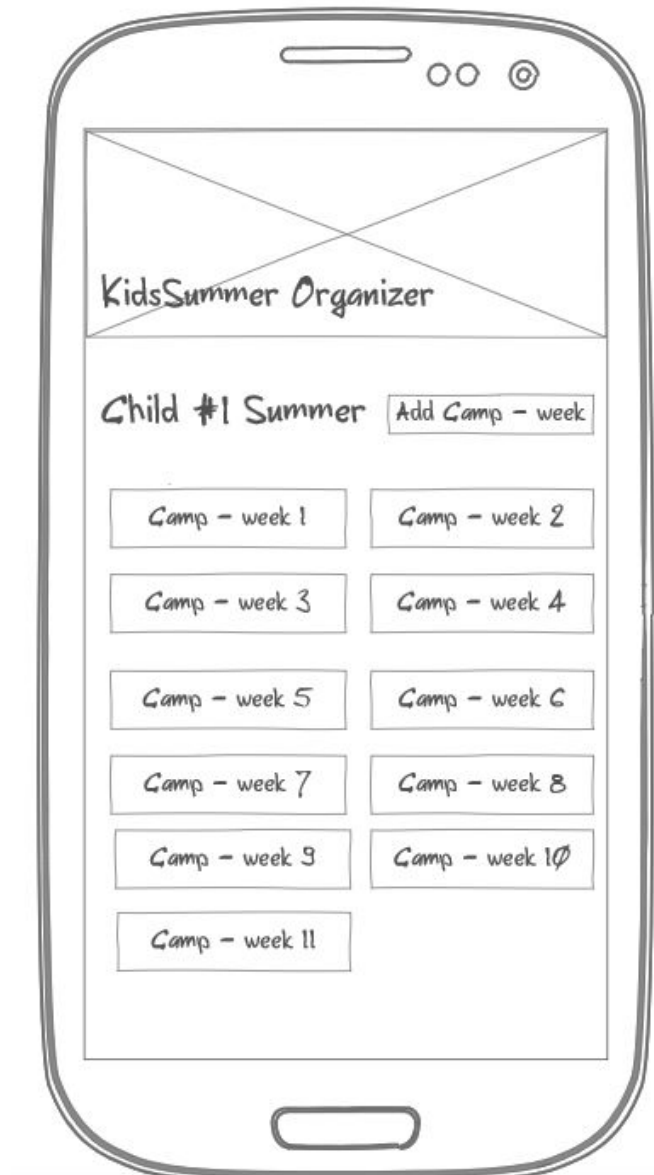✉ Child #1

✉ Child #2

✉ Child #3

Google Ad

This is the Main activity you can either Add a Child or Go to Child's camps page. If you click on Add Child it will take you to Screen #2. This page will have a google ad at the bottom.

Screen 2

This allows you to add a child to the database.

Screen 3

KidsSummer Organizer

Child #1 Summer [Add Camp – week]

Camp – week 1	Camp – week 2
Camp – week 3	Camp – week 4
Camp – week 5	Camp – week 6
Camp – week 7	Camp – week 8
Camp – week 9	Camp – week 10
Camp – week 11	

This allows you to view the list of camps for that child.

Screen 5

If you click on week 1 camp you can see the info you saved. You can also click on the map link that will open google maps for that location.

App's widget

Camp Current Week

Child # 1 : Name
Camp Location

Camp Hours

Bring Lunch: Yes

Contcat Person: Leah,
A073335555

Extra Notes : Do not forget
Wed pool day
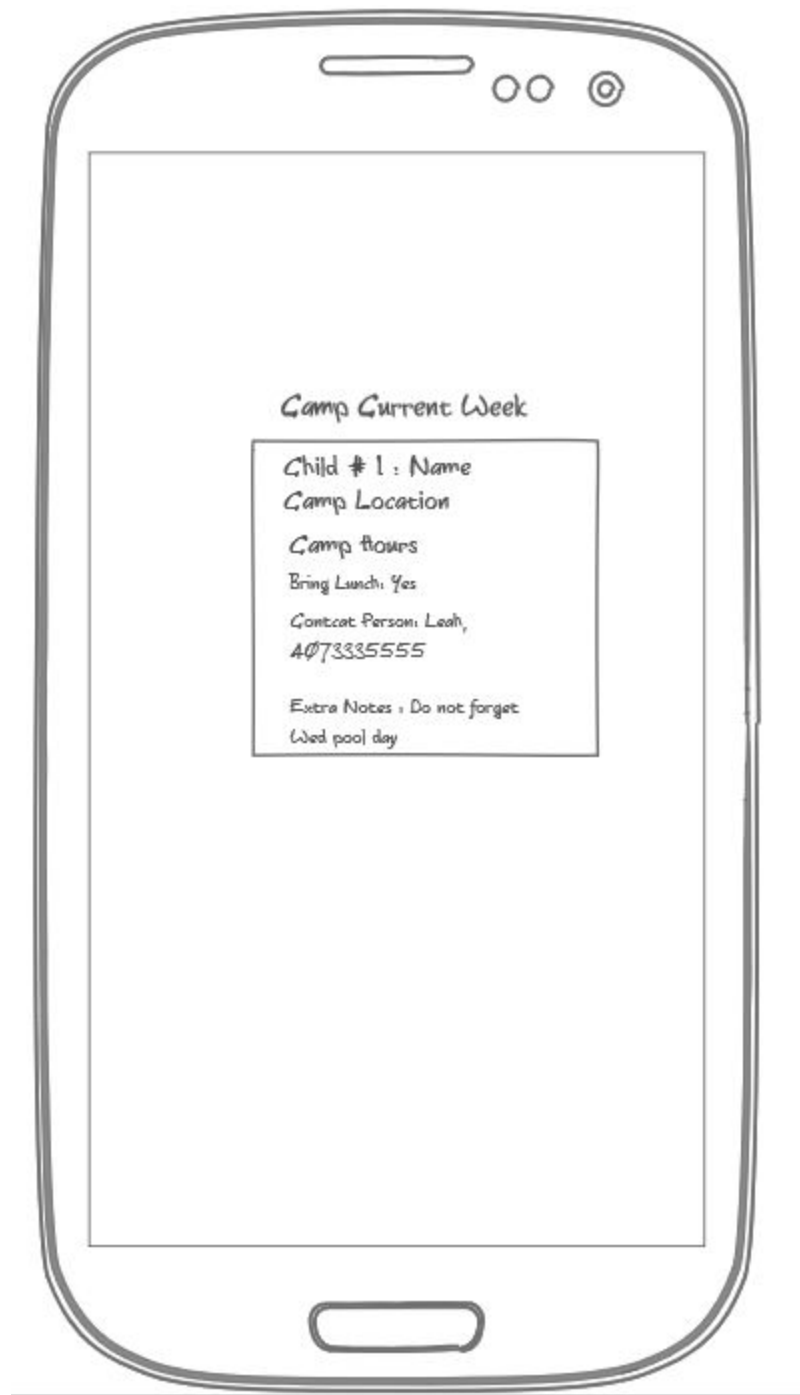
If you have any items in the database the app will show you the current week camp for each child on a scroll view . it will allow you to click on it and get to the camp screen.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Firebase database will be where I store my data.

Describe any edge or corner cases in the UX.

If no internet - Firebase apps automatically handle temporary network interruptions. Cached data is available while offline and Firebase resends any writes when network connectivity is restored.

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

implementation 'com.google.firebase:firebase-database:16.1.0' - to save my data.
third-party libraries:
1. https://jakewharton.github.io/butterknife/ - Butter Knife to find and automatically cast the corresponding view in your layout.
2. Library for Date Range picking: (not sure if I need it but putting it here) https://android-arsenal.com/details/1/6929 - Calendar Date Range Picker - to select the camp week date ranges.

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

Will be using google maps and google ads. Google maps will be a link to them with address to find. The ads will be a double click ads from google inserted in Main activity page.

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

Subtasks
- Implement UI for each activity
- Connect my app to firebase
- Create the data structure
- Enable disk persistence

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Add Child
- Build UI for view child
- Build UI for Add camp
- Build UI for view camp

Task 3: Connect the app to firebase

Describe the next task. List the subtasks. For example:
- Add realtime database to app
- Configure realtime database rules
- Structure my Database
- Enable offline capabilities

Task 4: Implement Google Play Services

Describe the next task. List the subtasks. For example:
- Add google ad to the main activity

Task 5: Error handling

Describe the next task. List the subtasks. For example:
- Test all error handling like when adding should add
- When viewing one child should be that child's camps
- When no internet firebase offline should work.

Add as many tasks as you need to complete your app.

Submission Instructions
- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
  - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:
- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"