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: nabishop

RighTide

Description

RighTide provides users with up to date information on marine weather which is especially important for surfing. The app is usable in California and uses user’s location to find nearby beaches and give them details about the wave conditions.

The app also ranks the beaches for the ideal surfing conditions (I may add beginner/intermediate/experienced rankings later). The user can also look up beaches based on county and top spots and can even favorite certain places.

Intended User

My intended user is surfers.

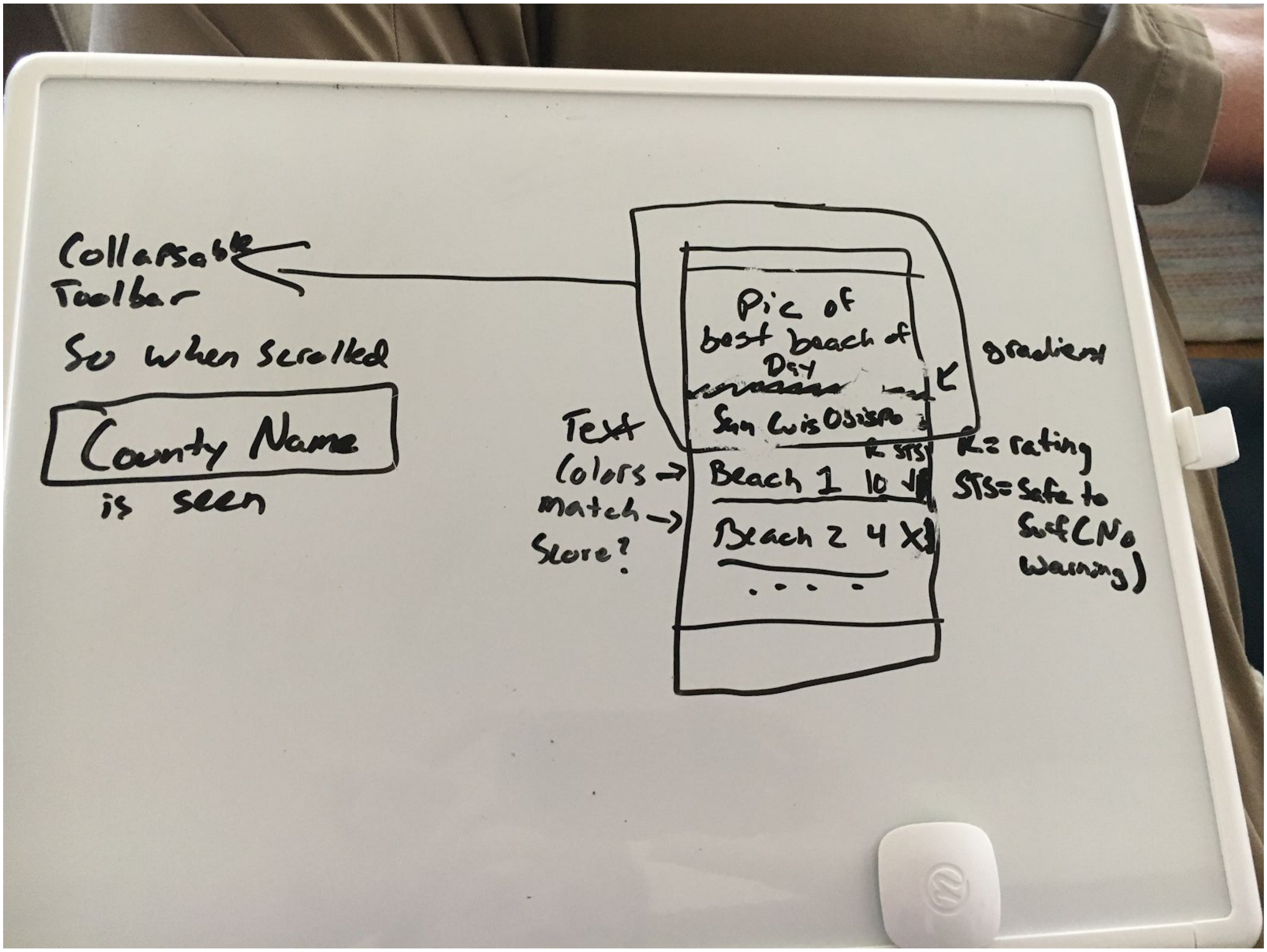
Features

- Shows weather information for nearby beaches
- Recommends beaches based on weather ranking for ideal conditions
- Allows for the user to favorite beaches and always be informed
- Sends the user optional notification each morning about how well certain beaches are doing
- Shows potential warnings for beach site
- Shows nearby counties and their beaches

User Interface Mocks

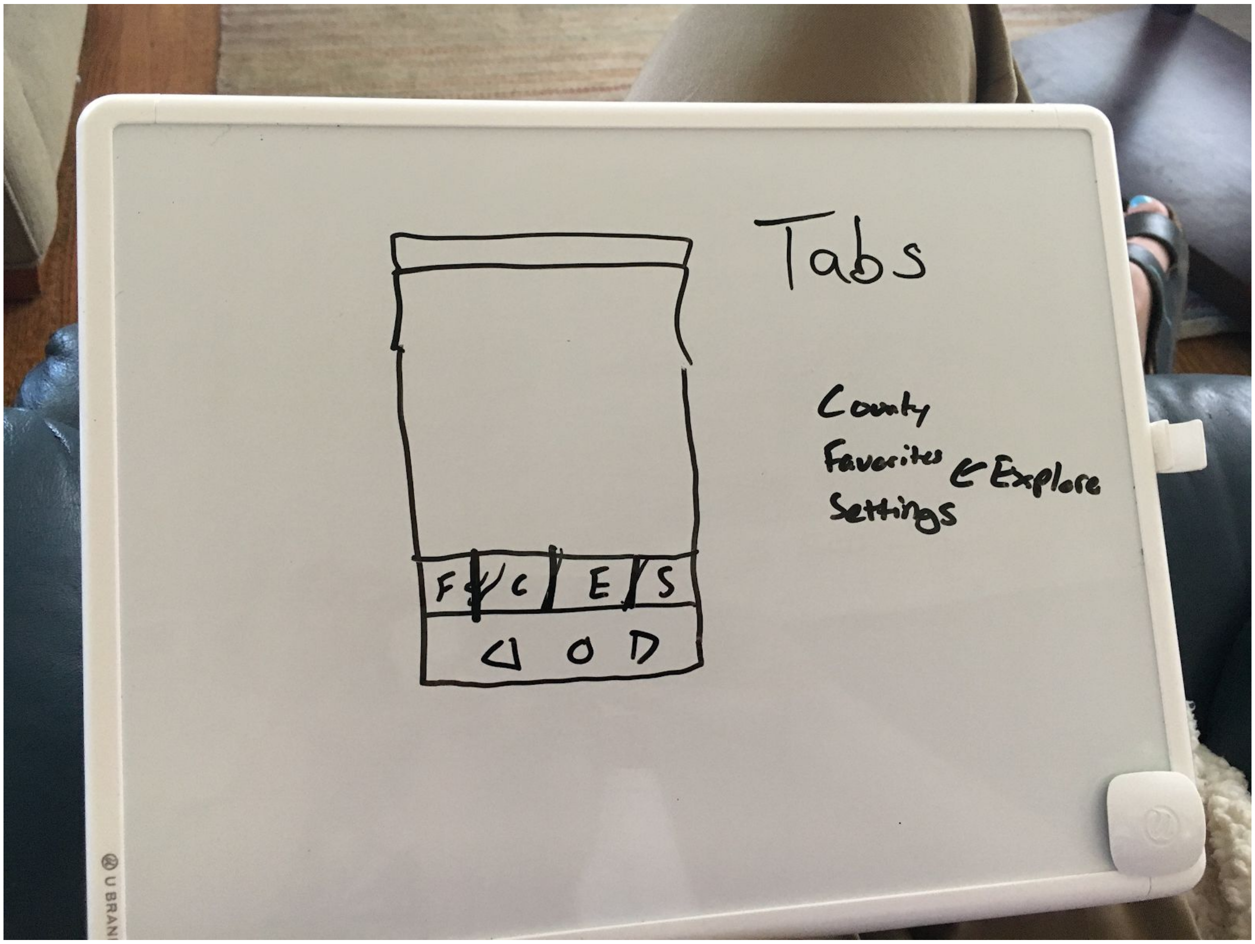
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1 - Beaches by County



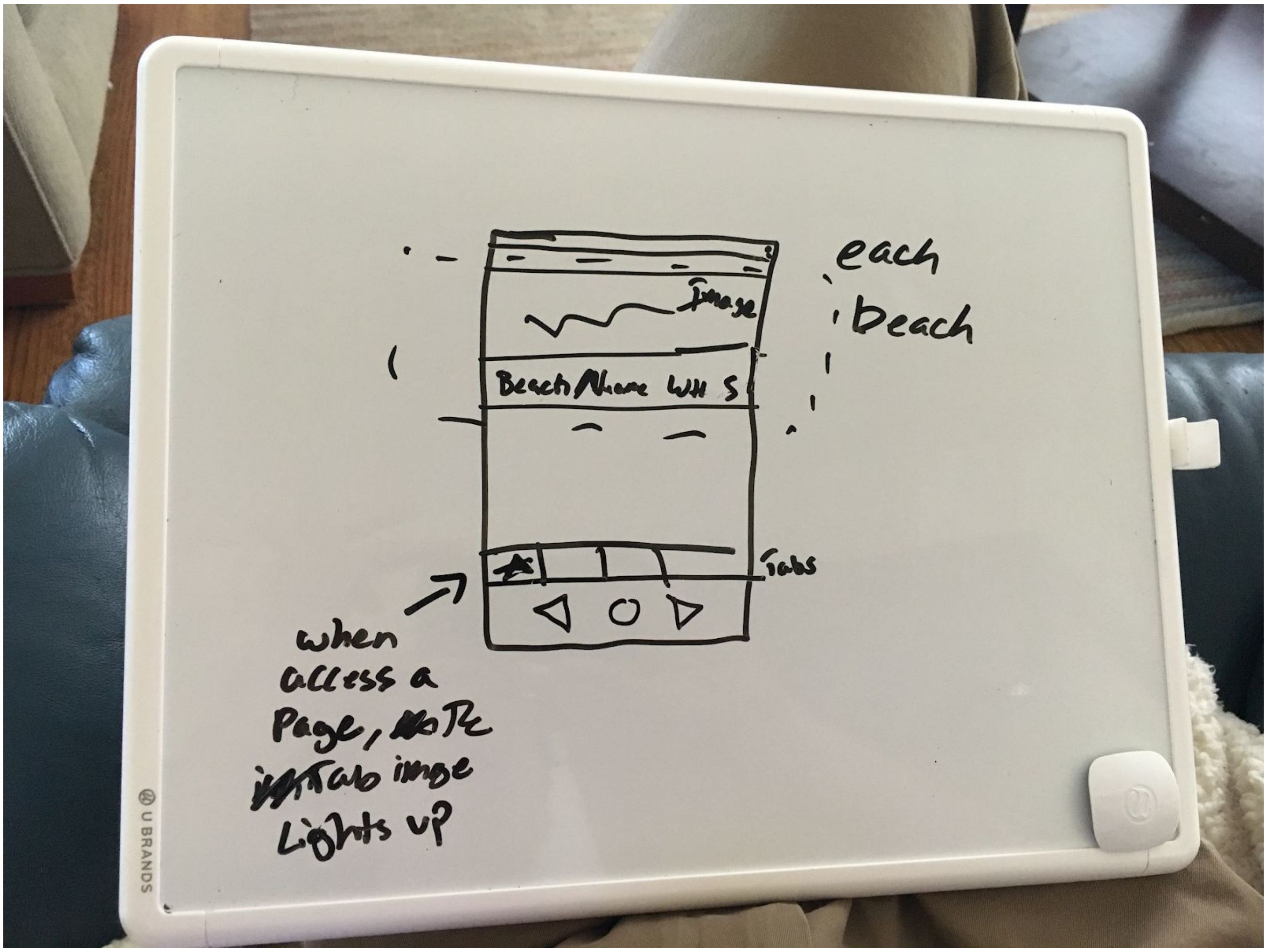
If the user does not have any saved favorites, this will be the first page that is shown. It shows all of the beaches in the users current county and gives them a little information about each. Clicking on a beach will launch a beach detail activity. May add in an option to favorite a beach in this list view.

1 Extra - Tabs for Navigation



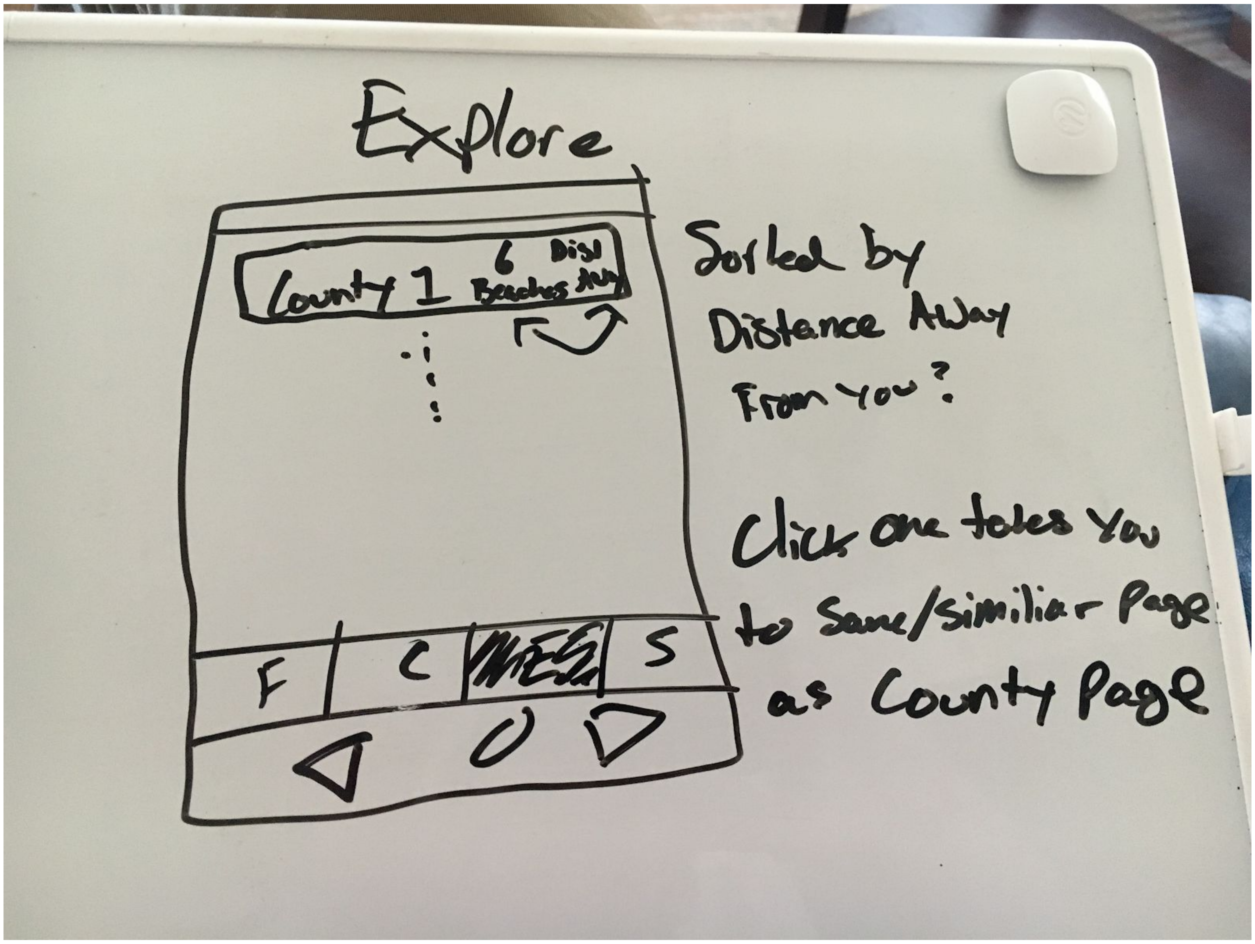
Not sure if this will be a top/bottom thing but there will be tabs for County, Favorites, Explore, and Settings.

Screen 2 - Favorites List



If the user has favorite(s) then this will be the first screen shown. All of this info will be saved via the content provider. Clicking on a beach will launch a beach detail activity.

Screen 3 - Explore

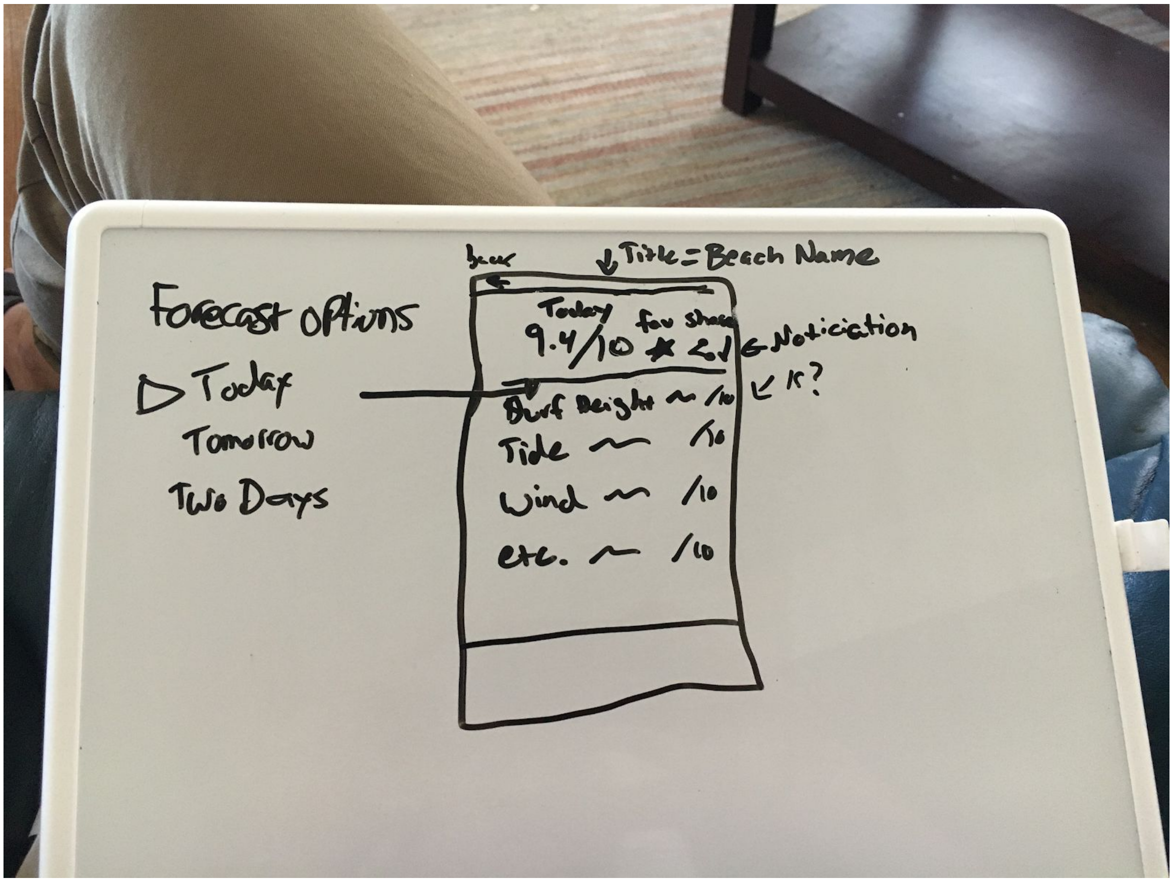


Sorted by Distance Away From you?

Click one takes you to Same/similiar Page as County Page

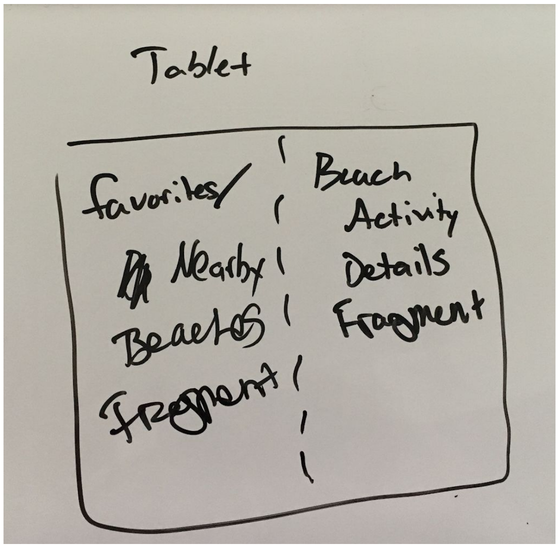
Will show nearby counties based on how close they are. Will also show how many beaches exist within each county. When a county is clicked a list like Screen 1 will pop up showing all the beaches.

Screen 4 - Beach Detail Screen

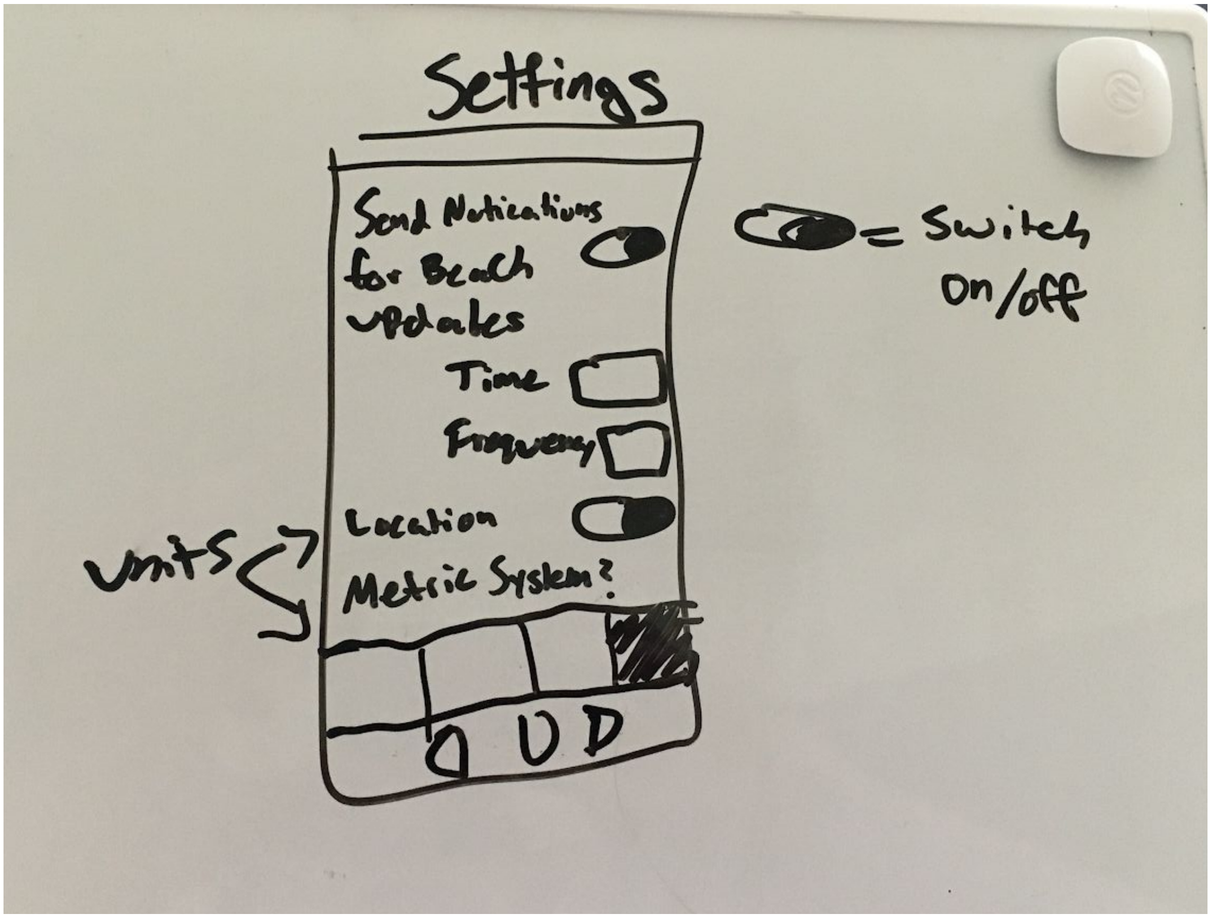


This is the screen for the beach details. Shows a bunch of weather information as well as a breakdown of how the score is calculated. The user can view the next few days too. Also here, the user can favorite a beach and turn on notification settings to receive alerts about how the beach is doing which can be edited in the settings.

Thinking about adding this in for tablets (view list and beach activity at same time)

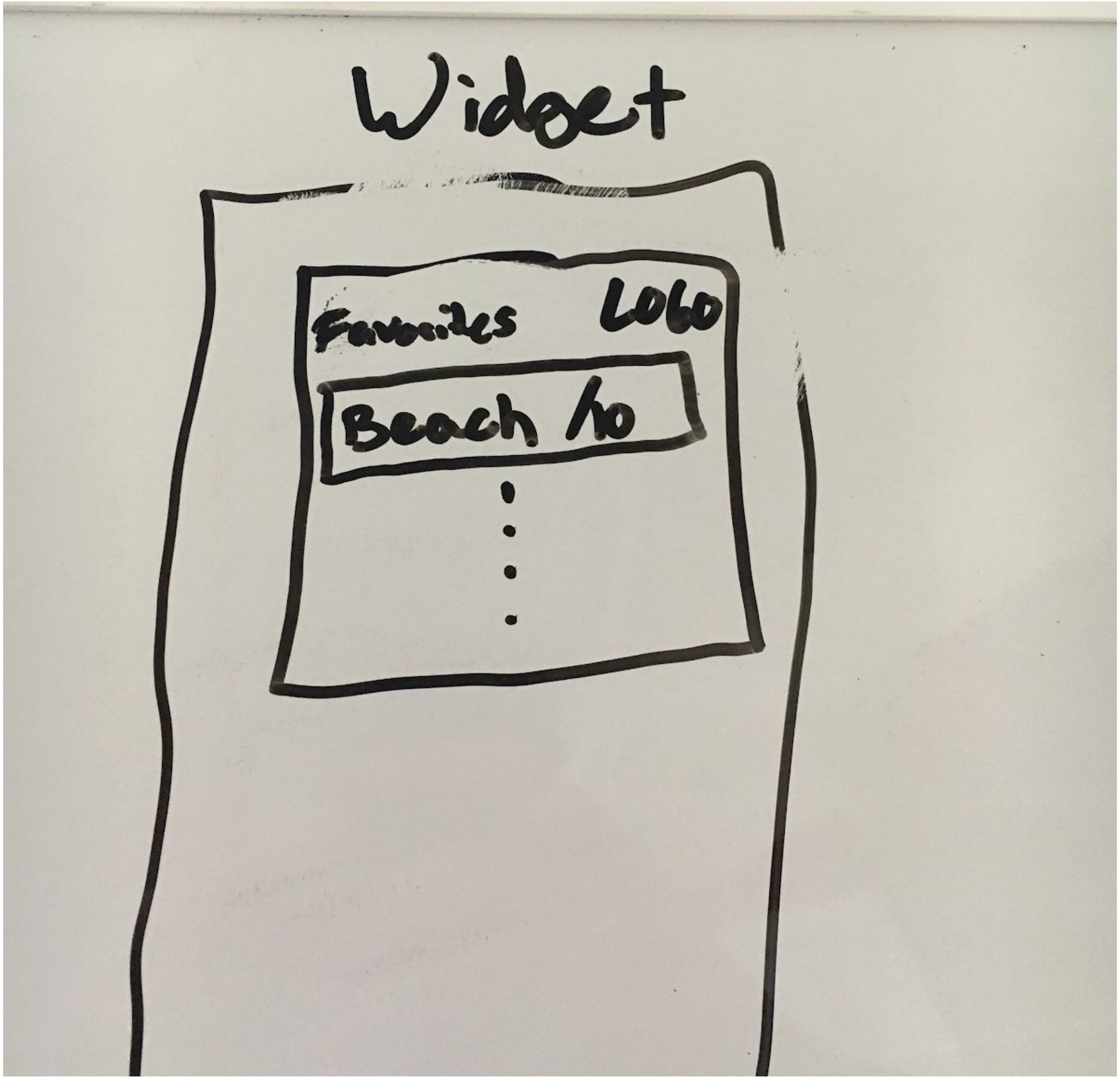

Tablet
favorites/ Nearby Beaches Fragment
Beach Activity Details Fragment

Screen 5 - Settings



This is the settings screen where you can update notification settings, location usage, and adjust your units of measurement.

Widget Screen



Shows the users favorites, LOGO in the top right corner of screen. For each beach it shows the days weather score (out of 10) and will open the detail activity if clicked. If there are no favorites then it will simply say to start favoriting some apps and will allow the user to click on the widget to open the app.

Key Considerations

How will your app handle data persistence?

I will use a Content Provider to store the users favorite beaches. I may add an option for data storage offline which could be stored in the database as well.

Describe any edge or corner cases in the UX.

The only edge case I can think of is in the activity where you can view both the beaches and the beach detail in one screen for the tablet. This is because of users trying to press back once they click on a beach but I think this will be fine because they will be easily able to maneuver around all the beaches since the list is always up in the tablet mode. Also, since the list will be half of the tablet screen, the images will not have to be very different from the phone version.

To consider a beach near it must be in the same county. If I want to rank how close they are to the person I can just use both of their coordinates.

If there is no internet connection then the app will not be able to populate the data until the user gets on wifi. I think that internally storing the data at incremental times throughout the day would put too much of a tax on the users device and the api.

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

I will use Picasso for getting images into the app for each beach and other included images. I will also use android testing libraries to ensure everything is working. I will also use RecyclerView to make custom layouts and use space efficiently.

Picasso Version - 2.71828
RecyclerView Version - v7:27.1.1
Espresso Version - 3.0.2
Android Test Version - 1.0.2
Appcompat Version - v7:27.1.1
Design Version - 27.1.1
compleSdkVersion 27
        Version name 1.0
classpath 'com.android.tools.build:gradle:3.2.0'
distributionUrl=https\://services.gradle.org/distributions/gradle-4.6-all.zip
Spitcast does not have a version number - http://www.spitcast.com/api/docs/
Unsplash does not have a version number - https://unsplash.com/developers

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

I'll use the google play service location api to show the user beaches in their county first. I will also use the spitcast api to get the surf data and then the unsplash api to get open source images for each beach. Finally, I will use the google play service analytics api to get information about my apps performance for the long term.

classpath 'com.google.gms:google-services:4.1.0'
Spitcast does not have a version number - http://www.spitcast.com/api/docs/
Unsplash does not have a version number - https://unsplash.com/developers

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

- Setup GitHub repository
- Configure libraries and implementations
- Lay out UI flow

Task 2: Obtain Weather Information

- Set up data structures (models) to hold beach / county information
    - County holds beaches which holds data for its beach
    - List of beaches
- Parse information into objects
    - Use manual http connection and read input stream with a buffered reader
- Test to ensure parsing correctly

Task 3: Set up Content Provider

- Set up Content Provider to store favorited beaches
- Test to ensure working
    - Use Loaders to fill the favorites screen with data stored in the database

Task 4: Implement UI for Each Activity and Fragment

- Create fragment UI for beach list
- Create fragment UI for beach details

- Create fragment UI for beach favorites
- Create activity UI for favorites
- Create activity UI for county beaches
- Create activity UI for explore
- Create activity UI for settings
- Create widget UI

Task 5: Implement Java code to make App functional

- Set up beach list
    - Then
        - Beach details
        - Beach favorites
- Set up activities for fragments
    - County beaches -> favorites -> explore -> settings
- Set up widget’
- **LIFT STRINGS INTO STRINGS.XML FILE FOR TRANSLATABILITY AND PORTABILITY**

Task 6: Test and add!

- Write tests for UI and other features of app
- Add new features as I see fit

Task 7: Submit to Udacity and Google Play Store

- Create a signkey and apply it to my release task
- Submit and continue polishing!

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**”