Coding style guide:

We will be using elements from the Google Java Style Guide to develop our application, using only the elements important/relevant to us.

Naming:
Each code file in the project will end with the .java extension. Method names will always start with a lowercase letter and class names will always start with a capital letter.

Code blocks and braces:
K&R style, with opening braces followed directly with a new line, and a line break after the closing brace. For example:

```
while (condition()) {
  method();
}
```

Block indentation:
Each code block will be indented with 2 spaces.

Column limit:
No line will exceed 100 columns. This line is drawn in Android Studio.

Camel Case:

1. Convert the phrase to plain ASCII and remove any apostrophes. For example, "Müller's algorithm" might become "Muellers algorithm".
2. Divide this result into words, splitting on spaces and any remaining punctuation (typically hyphens).
   - Recommended: if any word already has a conventional camel-case appearance in common usage, split this into its constituent parts (e.g., "AdWords" becomes "ad words"). Note that a word such as "iOS" is not really in camel case per se; it defies any convention, so this recommendation does not apply.
3. Now lowercase everything (including acronyms), then uppercase only the first character of:
   - ... each word, to yield upper camel case, or
   - ... each word except the first, to yield lower camel case
4. Finally, join all the words into a single identifier.

@Override is always used
Whenever it's legal, @Override is always used.

Javadocs are used on all user-defined functions
Fine to not use them when overriding a system or library's function

Done Criteria

User story:

- Approved by all team members
- Integrated into main program
- Everything works together seamlessly
- Approved by project owner
- Acceptance criteria met (changes per task)
- Easily findable and recognizable
- Tested by team

Features:

- Properly tested and working
- Coded using the style guide
- Documented using Java docs
- Test case written
- Scrum board updated

Acceptance Criteria

Home Button
As a user, I need a home button to take me back to the home page so that I can return at all times.

Acceptance Criteria:
1. Home button takes user to home page
2. Home button is easily recognizable
3. Home button is available on all pages

Category Button
As a user I need to get around the app by using a categories button.

Acceptance Criteria:
1. Home button takes user to home page
2. Home button is easily recognizable
3. Home button is available on all pages

Search Bar
As a user, I need to be able to use the search function to take me to desired places.

Acceptance Criteria:
1. Search bar is easily found
2. Search bar allows user to search for places
3. Search bar takes user to the item they searched for

About Page
As a user, I need to be able to know who made the app.

Acceptance criteria:
1. Page is easily found
2. Page clearly states team members and their positions
3. Page looks clean and is easy to read

Rating System
As a user, I want to be able to rate places (and see ratings), to know which are good and bad

Acceptance criteria:
1. App allows user to rate places
2. App allows user to see ratings made by other people

App Icon
As a phone user, I want to be able to easily find the app among my list of apps so that I can use it.

Acceptance criteria:

1. Icon is easily recognizable and pleasant to look at
2. Icon is labeled with the name of the app

Final Presentation
As a TA/Peer/professor I need a presentation so I know about the app.

Acceptance criteria:

1. Presentation describes functions of app
2. Presentation describes intentions of app
3. Presentation shows architecture of app
4. Presentation provides a demonstration of app
5. Presentation is easy to understand (not heavily loaded on text)

Team Working Agreement
As a TA/professor, I need a Team Working Agreement with details on how the team agreed to work together so that I can evaluate how well the team worked together

Acceptance criteria:

1. Lists the acceptance criteria
2. Gives our agreed style guide
3. Shows the UML diagram of how our app works
4. Gives a user manual for the app
5. Shows the done criteria for both user stories and individual tasks

---

Descriptions
As a user, I want to be able to read descriptions of the places on the map

Acceptance criteria:

1. When a user presses the pin it should show a small description of the place
2. When a user presses the small description it should take the user to a larger description of the place
3. The larger description should have pictures to show what the place looks like

---

Implement own Pins
As a student I want my voice heard / want others to discover the places I like / I would like to share my stories

Acceptance criteria:

1. User can implement own pins on map
2. User must write name of pin (place)
3. User must write short description of pin (place description)
4. User must write location (coordinates)
5. User must be able to enter information prior to creation of pin

---

Spam Filter
As a student I want relevant content, as to not waste time on digging through "spam"

Acceptance criteria:

1. When a place is down-rated below specified number it should be deleted

---

App
As a developer I want to put something cool on the market

Acceptance criteria:

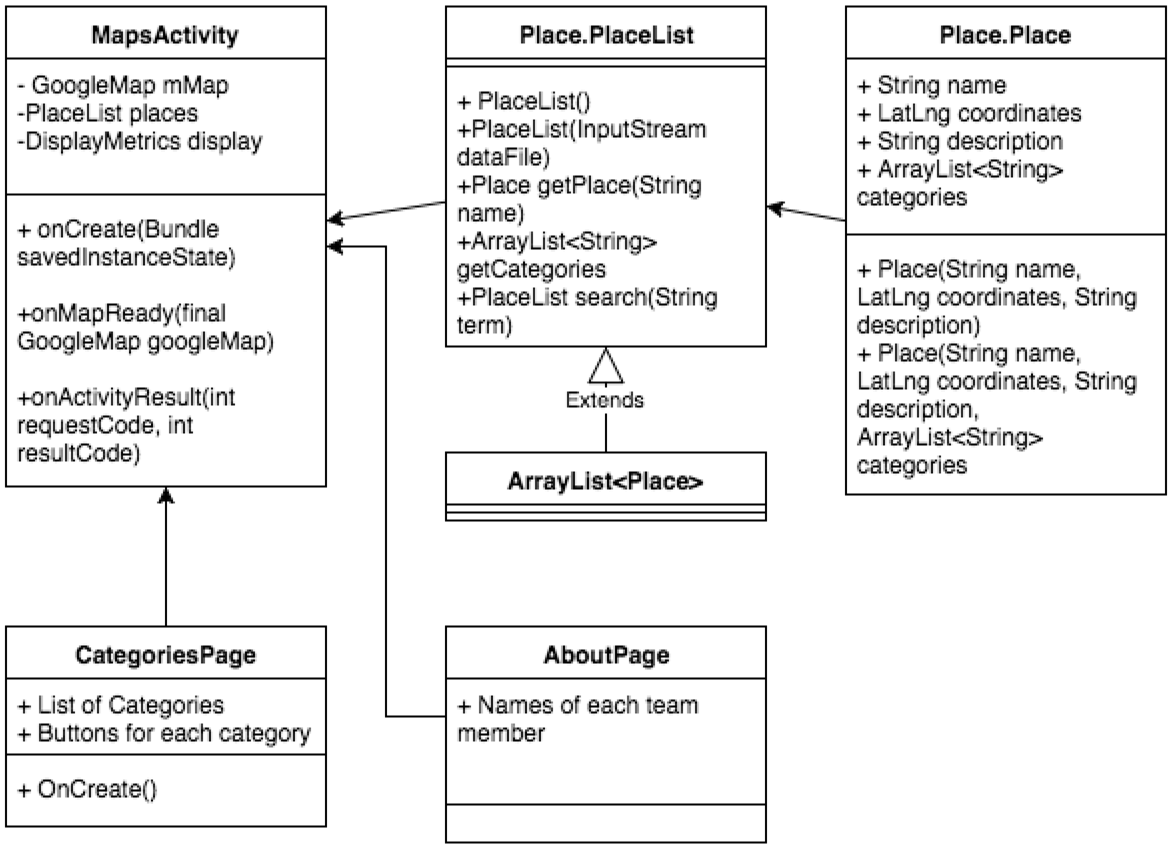
1. App should be made available through app-store
2. App should work without major bugs

App
As a new student, I would like to find points of interest on and around campus

Acceptance criteria:

1. App should be available to ucsc students
2. App should show pins with “points of interest” on and around campus
3. App should allow users to read description of places
4. App should show location of place

UML Architecture:


MapsActivity
- GoogleMap mMap
-PlaceList places
-DisplayMetrics display
+ onCreate(Bundle savedInstanceState)
+onMapReady(final GoogleMap googleMap)
+onActivityResult(int requestCode, int resultCode)
Place.PlaceList
+ PlaceList()
+PlaceList(InputStream dataFile)
+Place getPlace(String name)
+ArrayList<String> getCategories
+PlaceList search(String term)
Place.Place
+ String name
+ LatLng coordinates
+ String description
+ ArrayList<String> categories
+ Place(String name, LatLng coordinates, String description)
+ Place(String name, LatLng coordinates, String description, ArrayList<String> categories
Extends
ArrayList<Place>
CategoriesPage
+ List of Categories
+ Buttons for each category
+ OnCreate()
AboutPage
+ Names of each team member

User Manual:

To find the spots in a category:

1. Click “Go”
2. Click “Categories”
3. Select the category you’re interested in

To return to all places:

1. Click “Go”
2. Click “All”

To search for a place by description, category, or name:

1. Press on the search button on the top right corner
2. Type what you want to search for
3. Press “Search”