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: tripleducke

UCI Transit

Description

UCI Transit quickly displays the nearest UCI shuttle stops and their arrival times. Quickly pull up the app, check the times, and head your way. Never miss class again!

Intended User

Primarily UCI students/residents that live on campus apartments (as they are the majority of bus users), though any other people utilizing the UCI shuttles can benefit.

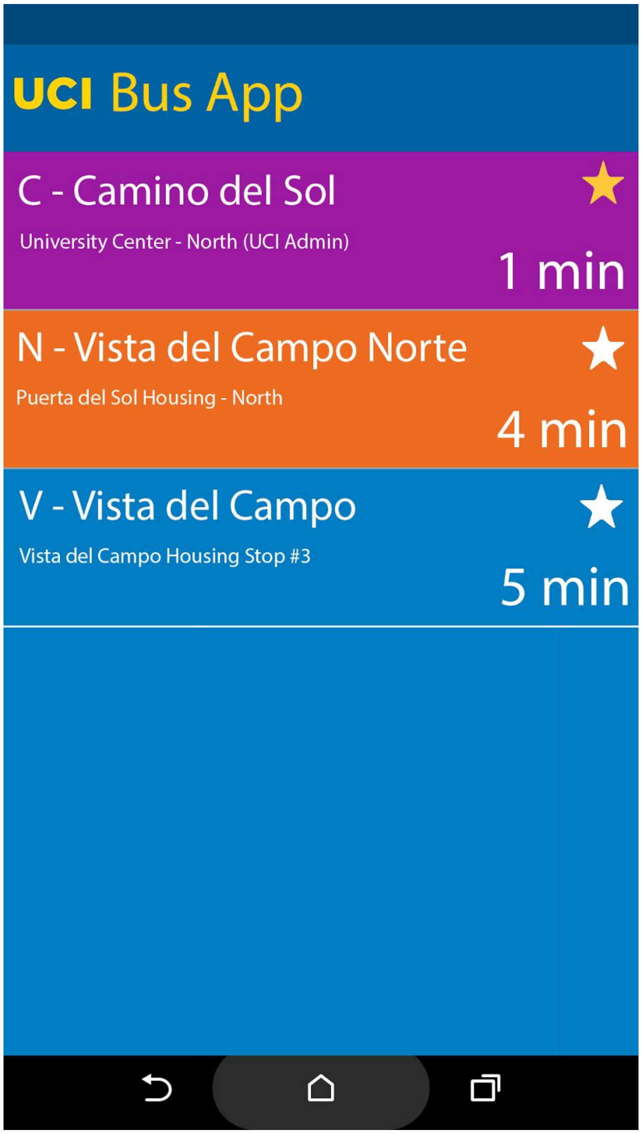
Features

- Nearby stops for quick access
- Save favorite stop + bus line
- Shows map of route

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.

Home page



UCI Bus App

C - Camino del Sol ★
University Center - North (UCI Admin)
1 min

N - Vista del Campo Norte ★
Puerta del Sol Housing - North
4 min

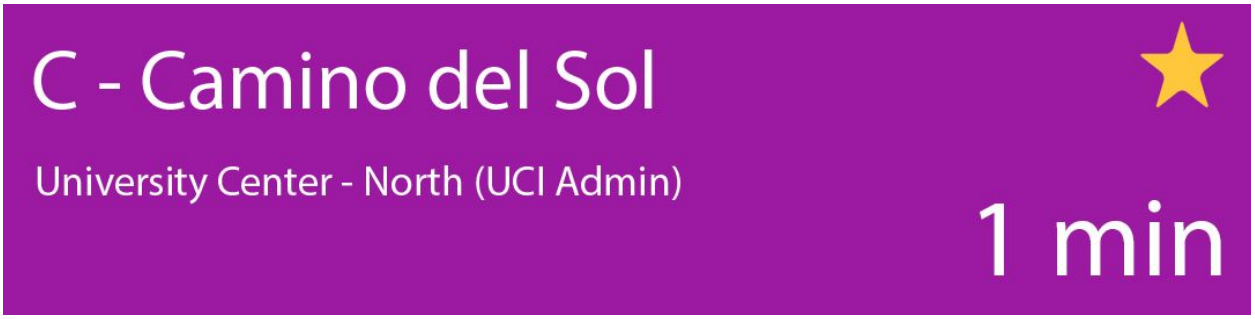
V - Vista del Campo ★
Vista del Campo Housing Stop #3
5 min

This is the main page, that displays nearby stops and their bus times. Users may scroll up or down to view the items. Logos and titles in the image were for mockup purposes only. Auto-updates every 20 seconds.

The layout is quite similar to Gmail: a list or recycler view with navigation drawer. I had thought to have items similar to Gmail's as well, but after looking at apps like Transit, I really enjoyed the full-bleed colors.

Order of the list would be favorites, nearest, then repeats. So using the above mock, Camino del Sol would have another item under the blue Vista del Campo (say at 10 minutes).

Detail Item



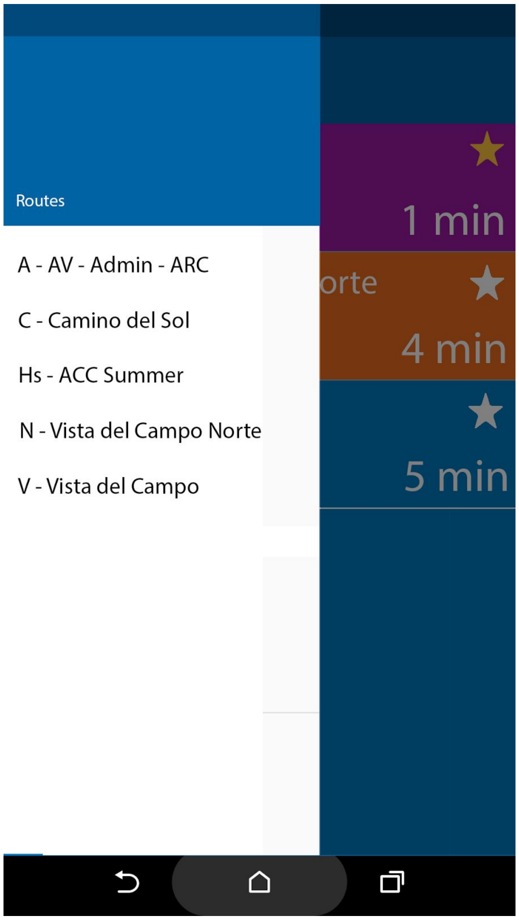
C - Camino del Sol ★

University Center - North (UCI Admin)

1 min

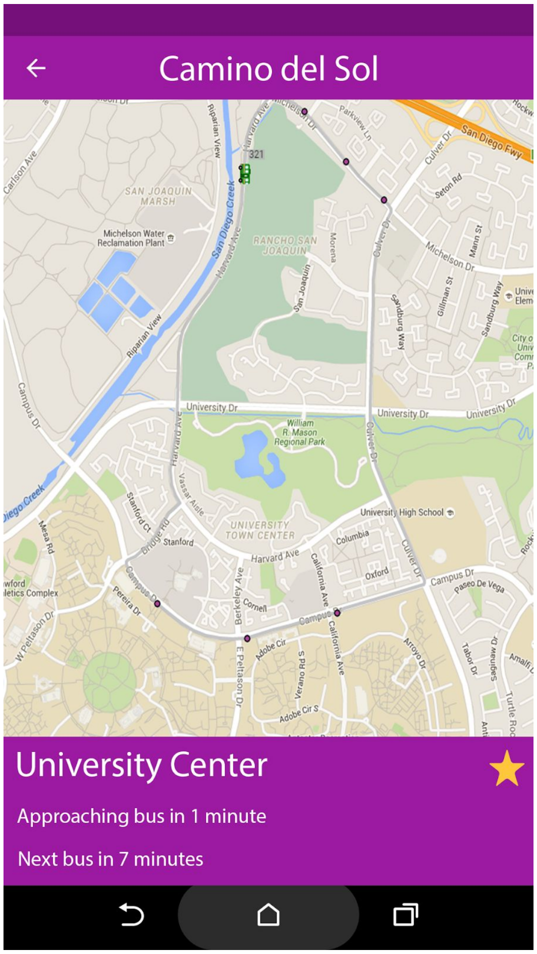
- Shows the bus line (C - Camino del Sol)
- Shows the bus stop (University Center - North UCI Admin)
- Shows the most immediate wait-time (1 min)
- Star button to favorite the stop/line, to prioritize in the display list.

Navigation Drawer



Navigation drawer will show all the routes. Clicking them will go to each route's map and details.

Route Detail Screen



Camino del Sol

University Center

Approaching bus in 1 minute

Next bus in 7 minutes

The route detail screen will show a map with bus route and stop. Clicking on a stop (which will be marked) will show a item displaying its info. The route name is on top with a back arrow to the main screen.

Key Considerations

How will your app handle data persistence?

Since there is no API for UCI shuttles, data will be pulled as html, parsed, and then stored into a content provider. The content provider will store and distribute data such as bus line, bus stops, and time until arrival. Since I want to include nearby service, initial data such as the stops and its location may have to be initialized.

Data pertaining to user favorites (bus stops) will be stored in preferences.

Describe any corner cases in the UX.

UX is pretty straight-forward. The main activity is scrollable and populated with detail items. Each detail item has a star button, which allows the users to favorite their stops. Navigation bar pulls up all the routes and clicking on a route takes the user to the detail activity, where a map is drawn. Both the nav-bar and app-bar back will take the user back to the main activity.

Network and Location

- If network is unavailable in the detail screen, no map will be drawn and a description will be provided to explain why.
- If network is unavailable in the main screen, an item saying so will be displayed instead.
- If data cannot be pulled, an item saying so will be displayed instead.
- If location is unavailable, an item saying so will be displayed on top, followed by favorites

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

- Will be using OkHttp to get bus times from the internet
- Maps API to render routes
- Various support libs for Material Design

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

- Configure libraries
- Familiarize and play with demos on Maps API + other libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for main activity
  - RecyclerView
  - Item layout
- Build UI for detail activity
  - Detail item layout
- Build UI for navigation bar

Task 3: Content Provider

(Inspired from the developer guide)

- Pull and parse html data
- Design data storage
- Design content URIs
- Implement Content Provider

Task 4: Service

- Build SyncAdapter
- Implement GCM

Task 5: Location

- Get user location
- Geofencing for stops

Task 6: Maps

- Get stop locations
- Build custom map
- Build custom marker and labels

Task 7: Corner Cases & Accessibility

- Implement messages in case of network and location failure
- Content descriptions
- RTL support

Task 8: Touch-ups

- Final touch-ups
- Final tests and clean-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**"