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

Description....................2

Intended User....................2

Features....................2

User Interface Mocks....................3

Screen 1: Map / Main....................3

Screen 2: Search....................3

Screen 3: Directions....................4

Screen 4: Trip details....................5

Screen 5: Stop details....................6

Screen 7: Line planning....................6

Screen 8: Menu....................7

Screen 9: Newsfeed screen....................7

Screen 10: List Screen....................8

Screen 11: Log in Screen....................8

Screen 12: Widget Screen....................9

Key Considerations....................10

Programming Language....................10

Android Studio and Gradle....................10

How will your app handle data persistence?....................10

API Temps Reel Tisseo – Real Time Tisseo API....................11

Describe any libraries you’ll be using and share your reasoning for including them....................11

Describe how you will implement Google Play Services or other external services....................11

Resources....................11

Accessibility....................11

Next Steps: Required Tasks....................11

Task 1: Project Setup....................12

Task 2: Create the Java Model for the API....................12

Task 3: Design the Database to save the information....................12

Task 4: Networking utilities....12
Task 5: Create the Main Activity....13
Task 6: Create the Login Activity....13
Task 7: Create the Map Activity....13
Task 8: Create the Search Activity....13
Task 9: Create the Directions Activity....13
Task 10: Create the Directions Details Activity....14
Task 11: Create the Stop Details Activity....14
Task 12: Create the Network News Activity....14
Task 13: Create the Favorites List Activity....14
Task 14: Create the Favorites Widget Activity....15

GitHub Username: electromedico

On y Va (French for Lets Go)

Description

Tired of waiting for the bus to pass by? You are not sure which is the best way to your destination?

On y Va helps plan your plan your daily trips with real time information from public transportation in Toulouse. We will guide you thru the entire trip with notifications so you don’t miss the next bus and we will tell you when to get off. Also our special feature start walking alarm. You lets know at what time you want to start your trip, and we will send you a notification when you need to start walking to the stop. This way you don’t have to worry about checking the time every five minutes (we will do that for you)

Intended User

This app is intended to anyone using the public transportation in Toulouse.

Features

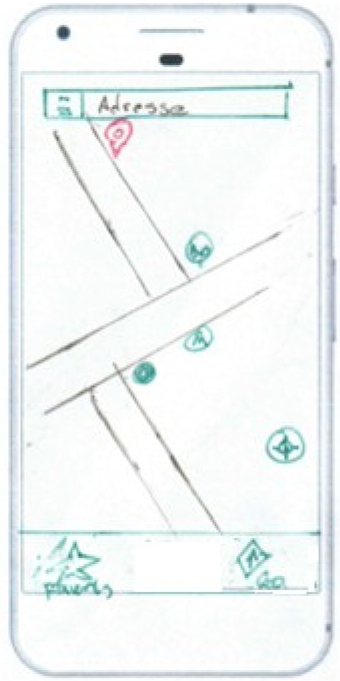
- Plan your trip and get directions
- Nearby stations
- Line information

- Favorites
- Daily trips alarms
- Traffic news

User Interface Mocks

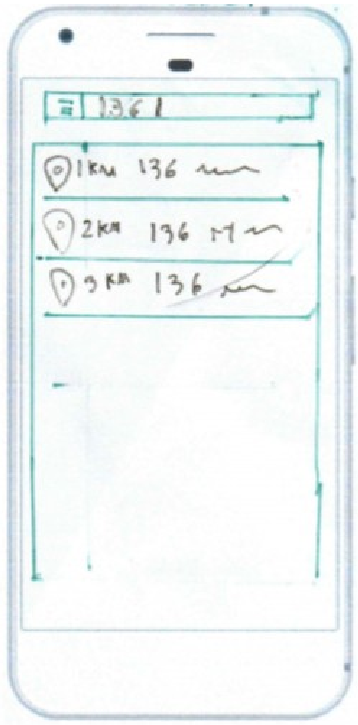
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1: Map / Main



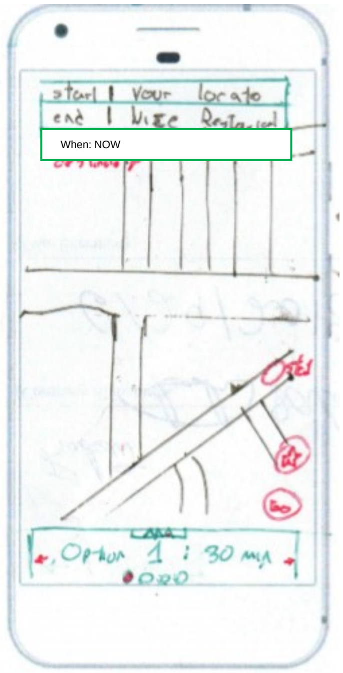
Main screen for the app. Here we will have the map with all the nearby stations, the toolbar with a search field and a buttons for the menu.
A FAB for finding your current location in the map
At the bottom we will have the buttons to access the favorites and start the trip.
If you click once in the map everything except the FAB will hide.

Screen 2: Search



Here the user can search for an address or points of interests. It will automatically update as the user continues to tap the information
Also the App will propose recants trips

Screen 3: Directions

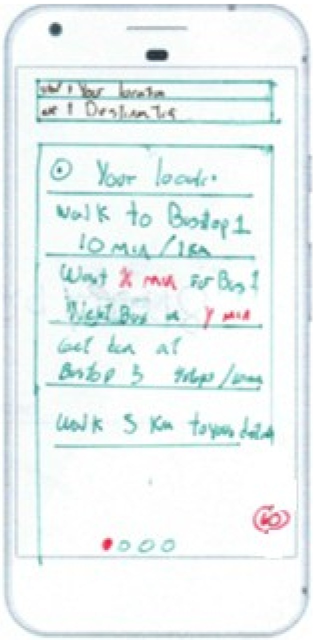


Here the User can see the starting and ending point of his trip. He also can chose at what time to start the trip

Also at the bottom if any, the user can chose from several options.

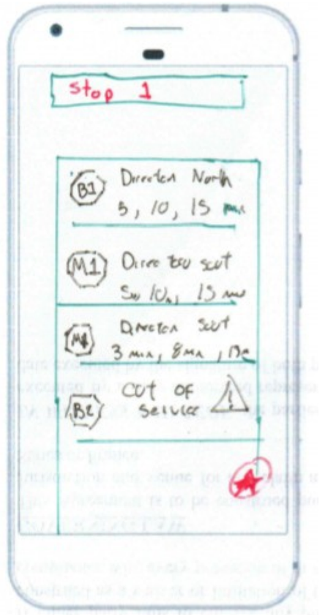
Two FAB buttons, one to add the trip to favorites and the other to start the trip.

Screen 4: Trip details



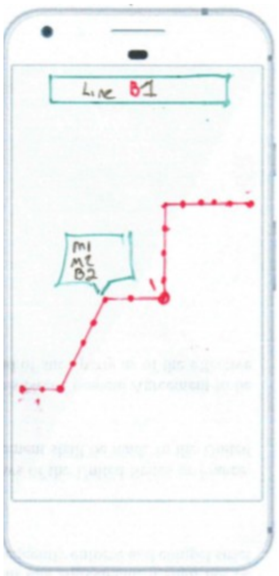
If the user taps in the option tab at the bottom of the previous screen or swipes up, the app will show the details of the trip.
Also the user can swipe left or right to see the other options.
The FAB buttons allow to save in favorites, start the trip and to set an alarm to get notified when to go.

Screen 5: Stop details



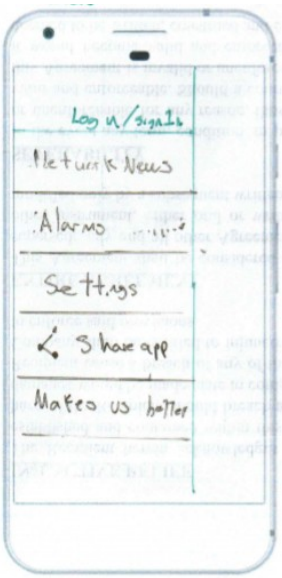
This screen will be shown when the user taps in a station in the map. Here the user can see all the information regarding the stop.
There is a FAB that allows the user to add to favorites the stop.

Screen 7: Line planning



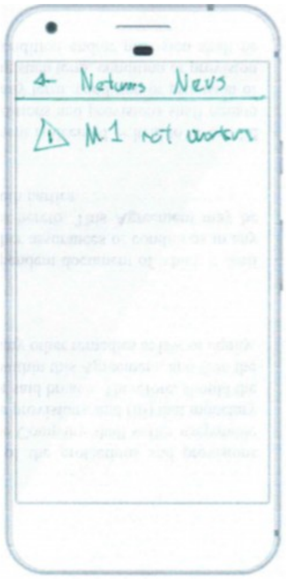
This screen will show in the map all the line including all the stops. Also if the user taps in one a pop up will show all the information regarding the stop.

Screen 8: Menu



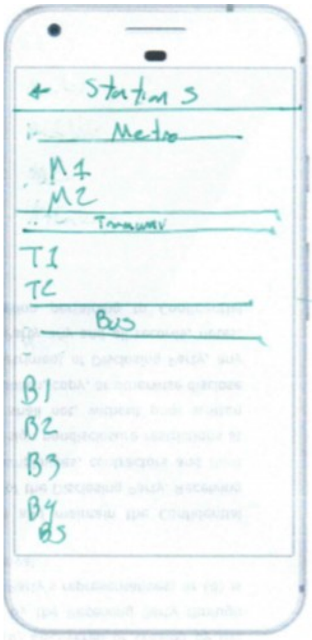
This screen will show the buttons for accessing the networks newsfeed, alarms, settings, share the app and a comments section.

Screen 9: Newsfeed screen



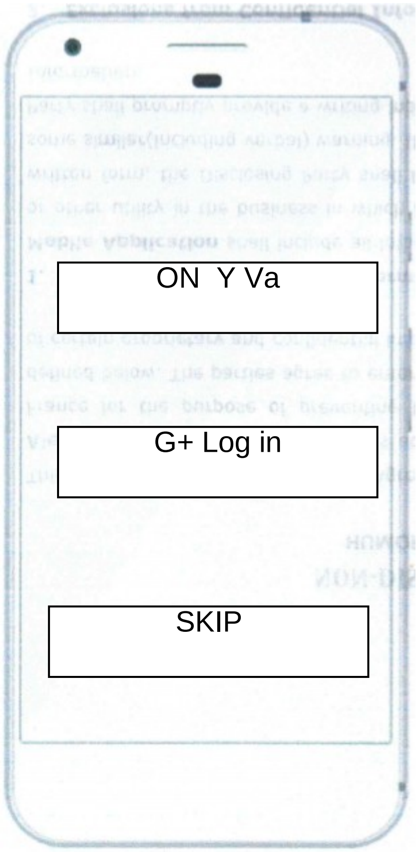
This screen will show all the news and info from the public transport network.

Screen 10: List Screen



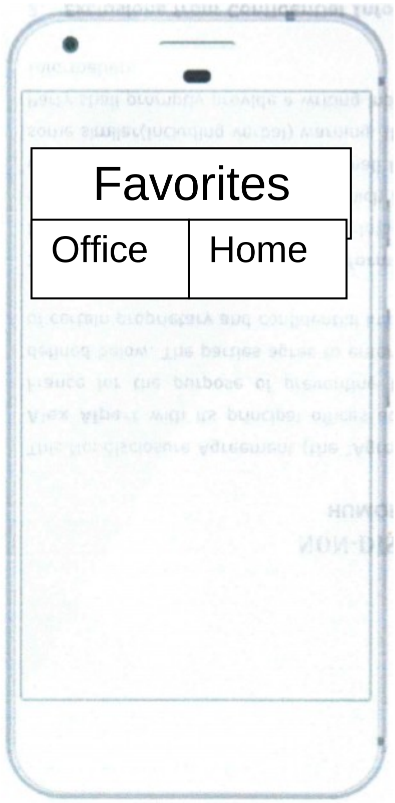
This screen will show a list of all lines. This will be reutilized by favorites and the list of all the stations.

Screen 11: Log in Screen



In this screen the app will give the option to log in or to skip the log on. This screen will be only shown the first time the app is launched

Screen 12: Widget Screen



In this widget the user can quick access his favorites and with one tap the app will launch and calculate the trip

Key Considerations

Programming Language

The app will be solely written in JAVA

Android Studio and Gradle

The version used for Android Studio will be 3.1.0+
The Gradle version will be 4.4

How will your app handle data persistence?

We will use a two different ways to store data. One will be used via a local database and a content provider. The access to the Database will be via AsyncTask
If the users has log in with a google account, we will store the information in a Firebase Real-time database.

API Temps Reel Tisseo – Real Time Tisseo API

We will the API to feed the app. The API is part of the OpenData of the region.

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

Retrofit version 2.4.0 will be use to manage the access to the REST API.
Google Maps version 15.0.1 platform will be used for all treatment related to the map.
ButterKnife version 8.1.1 will be used to inject the views.
GSON version 2.8.5 to transform the JSON into objects
Google Account Log In version 15.0.1 will be used to implement the log in

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

As expressed before we will use the firebase Real time data base to make a backup the information of the user.

The log in function will be managed via Google Sing in.

The most important part of this app is the maps capabilities. For this we will use google maps.

Resources

All the colors, strings, themes, will be stored in the resources files. Also when possible all the static constants will be stored in the resources files.

Accessibility

The app will include content description and non-audio version of notification and alerts.

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

Starting point, it will lay out all the foundations for the app.

- Create a new project in Android Studio
- Create a new repository in GitHub
- Push the initial commit
- Set up all the libraries
- Add the permissions to the manifest
- Sing up for all the APIs and get the keys.
- Add all the image resources(icons, logos)
- Commit and Push all the changes.

Task 2: Create the Java Model for the API

Create the Java model of the API

- Get the JSON model for API Temps Reel Tisseo
- Use JSONSCHEMA2POJO to generate the Java model. It's important to make the classes Serializable or Parceable.
- Commit and push

Task 3: Design the Database to save the information.

Design and implement the database for the app. This will be a single table containing all the specific information (favorites, alarms, and preferences)

- Create the package contentProviderUtils
- Here we will implement the Model class for the database
- Create the content provider
- Add all the necessary information to the manifest.
- Create the utility class or interface to access the database.
- Commit and Push

Task 4: Networking utilities

In this task we will set up and add all the utilities classes to access the REST services

- Create the networkUtils package
- Create the API declaration interface
- Create the utility class for the network package
- Commit and Push

Task 5: Create the Main Activity

In this activity we will verify if the user is already logged in or if it is the first time the app has been launched

- If it is the first time, the app will call the log-in activity
- If not, the app will call the map activity.
- Commit and push

Task 6: Create the Login Activity

Implementation of the login screen. The user will have the option to login with his google account or continue without login.
If he decides not to log, a popup will be shown that his information will not be backed up. If the user chooses to log, the app will launch automatic the Google Sign-in API. This screen will only be shown the first time the user launches the APP. Otherwise we will arrive to the map screen.

- Create LogInActivity.
- Implement google Sign-in
- Create the layout
- Commit and push

Task 7: Create the Map Activity

In this activity we will get the current location, and then load all nearby stations.
Then we will add the markers to the map. Basic navigation functionality will be implemented.
If the user zooms out enough the stations markers will be gone leaving only the current location marker.

- Create MapActivity
- Implement google maps.
- Code all listeners for the buttons.
- Implement all the transitions so the app feels natural when launching a new activity
- Create the map_activity_layout
- Commit and Push

Task 8: Create the Search Activity

This activity will use google maps and the tisseo API to locate the point of intererst searched by the user. It will show real time search and will propose points matching the query.

- Create the SearchActivity
- Implement the Search Overview
- Create the search_activity_layout
- Implement the transition to make it look that the user has returned to the map screen.
- Commit and push.

Task 9: Create the Directions Activity

This activity will show the trajectory for the proposed trip. At the top we will see the starting and ending point. At the bottom of the screen the user will have a list of possible routes. Each time a new route is chosen, the app automatically updates the trajectory. If the user swipes up the app will transition to the trip details giving the impression that the screen is just on top of the map.

- Create DirectionsMapActivity.
- Implement google maps
- Query the base for the trajectory
- Map it
- Implement the trajectory list to look like a pager
- Add actions for the FABs
- Create the directions_map_activity_layout

Task 10: Create the Directions Details Activity

This activity will be a View with a list of all instructions to complete the trip

- Create DirectionsDetailsActivity.
- Create the ViewHolders for the RecyclerView
- Create the adapter
- Create the directions_details_activity_layout
- Add actions for the FAB

Task 11: Create the Stop Details Activity

This activity will be a view with a list of all the line that stop by. Here each ViewHolder corresponds to a line with the name and the next passages. Also the user can add it to his favorites

- Create StopDetailsActivity.
- Create the ViewHolders for the RecyclerView
- Create the adapter
- Create the stop_details_activity_layout
- Add actions for the FAB

Task 12: Create the Network News Activity

This activity will be a view with a list of all relevant news to the Toulouse public transportation

- Create NetworkNewsActivity.
- Create the ViewHolders for the RecyclerView
- Create the adapter
- Create the network_news_activity_layout

Task 13: Create the Favorites List Activity

This activity will be a view with a list of all the saved points (bus stops, adresses, etc). On click the app will calculate the trip and will show the Directions Activity. The user can delete a favorite with a swipe. Is he deletes something a snack bar will show with the option to undo.

- Create FavoritesListActivity.
- Create the ViewHolders for the RecyclerView
- Create the adapter
- Create the favorites_list_activity_layout
- Code the access to the database

Task 14: Create the Favorites Widget Activity

This widget will show a list of favorites and will launch the app on click and arrive to the DirectionActivity.

- Create the widget with the wizzard.
- Create the widget_layout

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**”