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: usagiChan

Lol Tell Me

Description

Are you a League of Legends (http://leagueoflegends.com) player and don’t know who is the new champion of this week? Are you playing and need some tips about your champion?
Get all League of Legends’ champions at your fingertips. All the information you need when you want to. Up to date and easy to use. Don’t be outdated.

Information provided by League of Legends © Riot Games, Inc.

Intended User

League of Legends’ players. Between 15 and 30 years old.

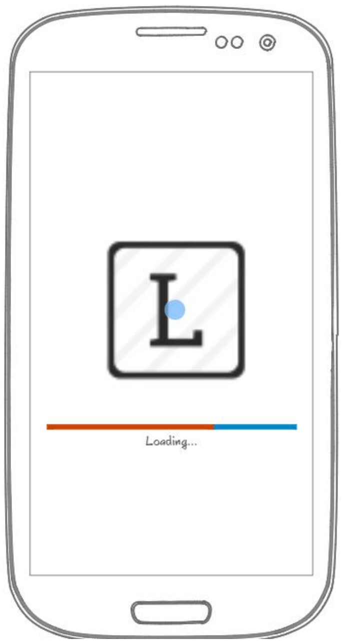
Features

List the main features of your app. For example:

- Displays information
- Saves information
- Changes information lenguage
- Refresh data periodically and manually

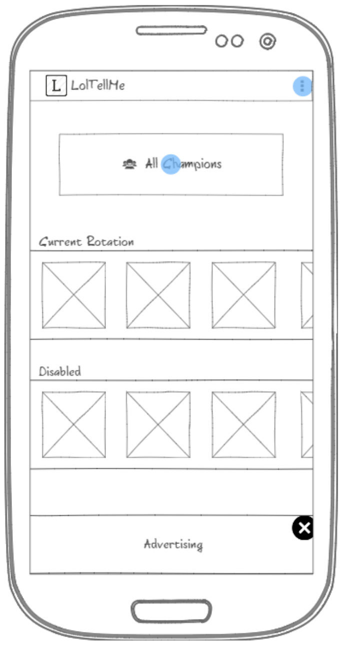
User Interface Mocks

Splash



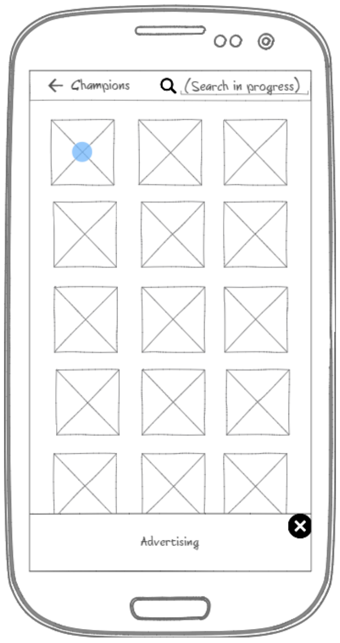
First Screen which shows a progress bar while the app's content is downloaded from the web.

Main Screen



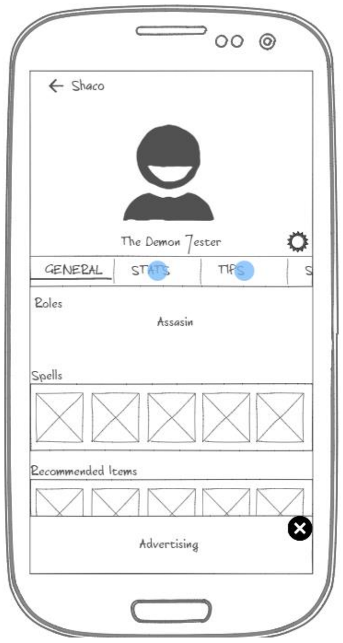
Displays easy to access data and an option to see the rest of the content.

Champions



Shows all champions indicating if they are currently free or disabled. Also has an option to easily search between all this items by its name or tag.

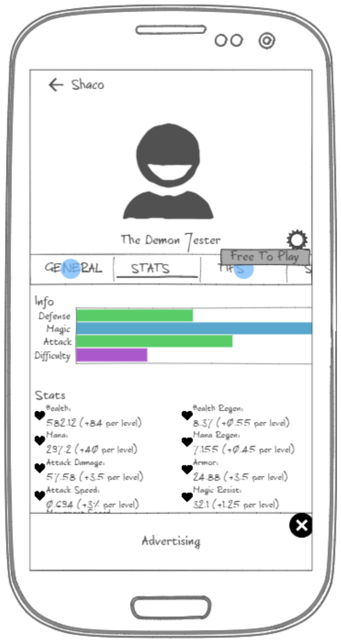
Champion Detail - General



Displays all relevant information about the current champion selected. Shows with an icon if the champion is currently free or disabled.

The general tab shows the main roles of the champion, its spells, and the recommended items for it.

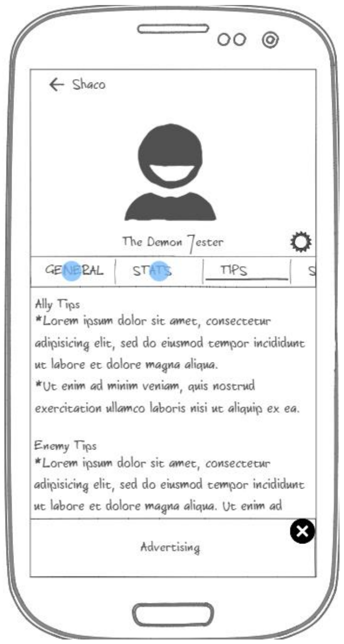
Champion Detail - Stats



Health:
582.12 (+84 per level)
Mana:
297.2 (+40 per level)
Attack Damage:
57.58 (+3.5 per level)
Attack Speed:
0.694 (+3% per level)
Health Regen:
8.37 (+0.55 per level)
Mana Regen:
7.155 (+0.45 per level)
Armor:
24.88 (+3.5 per level)
Magic Resist:
32.1 (+1.25 per level)

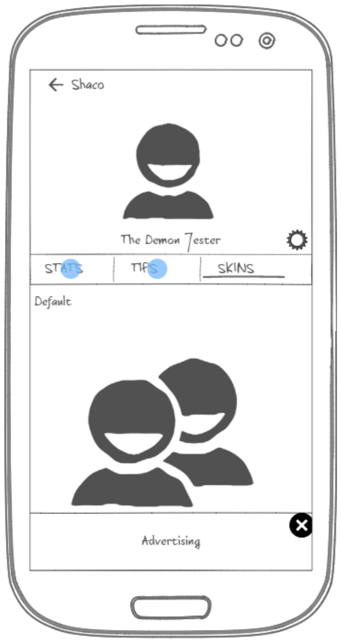
The stats tab shows the statistical information related to the user’s specialty and difficulty in comparison with the rest of champions. It also shows the base stat numbers of the champion and its increments per each level.

Champion Detail - Tips


Shaco
The Demon Jester
GENERAL
STATS
TIPS
Ally Tips
*Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
*Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea.
Enemy Tips
*Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
Advertising

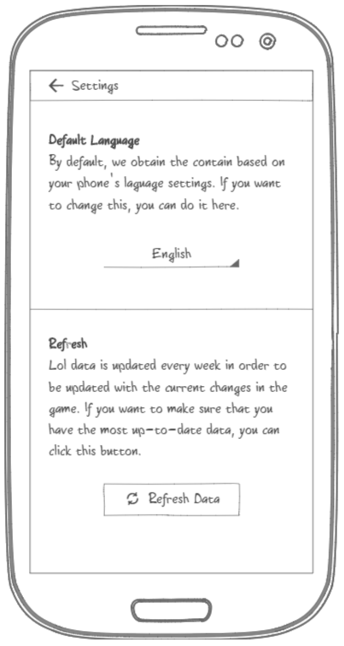
The tips tab shows tips to allies who are playing with this champion or enemies who are against it.

Champion Detail - Skins



Shows all the champion skins via a view pager to easily navigate between them.

Settings



Default Language
By default, we obtain the contain based on your phone's laguage settings. If you want to change this, you can do it here.

English

Refresh
Lol data is updated every week in order to be updated with the current changes in the game. If you want to make sure that you have the most up-to-date data, you can click this button.

In the settings screen, the user has the possibility to change the language of the data displayed and to download an up-to-date version of the data. This last feature can only be accomplished every certain periods of time because of the limited number of requests offered by the API used.

Key Considerations

How will your app handle data persistence?

The language preferences as well as the last time an update has been requested will be saved using SharedPreferences.
The rest of the data (Champions and items registers) will be saved in a database using Sugar ORM

Describe any corner cases in the UX.

In the Splash/Loader screen, if there is no internet when data is going to be downloaded. A message will be displayed telling the user to turn on his/her internet connection.
If the download is stopped, a variable will save which parts were completed so the app will resume the download when opened again.

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

- **Picasso**. Easy to use library used to handle the loading and caching of images.
- **Sugar ORM**. Database persistence library. Provides a simple and concise way to integrate your application models into SQLite. It is known to be: not much verbose, quick to set up, and hands-free.
- **Retrofit**. Elegant solution for organizing API calls in a project. The request method and relative URL are added with an annotation, which makes code clean and simple. With annotations, you can easily add a request body, manipulate the URL or headers and add query parameters.
- **GSON**. Java library used for serializing and deserializing Java objects from and into JSON. JSON is mostly used because it's lightweight and much simpler than XML.
- **MPAndroidChart**. Powerful & easy to use chart library for Android.
- **Crashlytics**. Library which provides deep and actionable insights, even the exact line of code your app crashed on.

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

- Sugar
    1. Gradle:compile 'com.github.satyan:sugar:1.4'
    2. Configuration
        - <application android:label="@string/app_name" android:icon="@drawable/icon" android:name="com.orm.SugarApp">
            - <meta-data android:name="DATABASE" android:value="sugar_example.db" />
            - <meta-data android:name="VERSION" android:value="2" />
            - <meta-data android:name="QUERY_LOG" android:value="true" />
            - <meta-data android:name="DOMAIN_PACKAGE_NAME" android:value="com.example" />
        - </application>
    3. Create Entities.

- Retrofit
    1. Create APIService interface class
    2. Create Url interface class
    3. Create RequestManager class
    4. Create Response class
    5. Create Error class

- Crashlytics
    1. Download Fabric sdk for android studio.
    2. Open Fabric plugin.
    3. Integrate Crashlytics so click on install button of Crashlytics.

Configure gradle flavors

- Configure debug flavor
- Configure release flavor

Add advertisement

- Configure Construct an InterstitialAd object and set its ad unit ID.

- Request an ad.
- Check that an ad is loaded and then display it.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for Splash / Loader
- Build UI for MainActivity
- Build UI for Champions
- Build UI for Champion Detail – General
- Build UI for Champion Detail – Stats
- Build UI for Champion Detail – Tips
- Build UI for Champion Detail – Skins
- Build UI for Settings

Task 3: Implement Splash / Loader Functionality

- Download content and show progress in progress bar.
- Check internet connection and display a message telling the user to turn on his/her internet connection.
- Save in a variable which parts were completed so the app will resume the download when opened again, in case the download stops.
- Save downloaded data in the database.

Task 4: Implement MainActivity Functionality

- Display current rotation in horizontal views.
- Display disabled champions in horizontal views.

Task 5: Implement Champions Functionality

- Display all champions
- Implement search by champion’s name or tag associated.
- Render to Detail Screen

Task 6: Implement Champion Detail Functionality

- Display champion information: Image, title. Also, show an icon if it is free to play or disabled
- General
    - Show the main roles of the champion.
    - Show its spells.
    - Show the recommended items for it.
- Stats
    - Show the statistical information related to the user's specialty and difficulty in comparison with the rest of champions.
    - Show the base stat numbers of the champion and its increments per each level.
- Tips
    - Show tips to allies who are playing with this champion or enemies who are against it.
- Skins
    - Show all the champion skins via a view pager to easily navigate between them.

Task 7: Implement Settings Functionality

- Option to change the language of the data displayed
- Option to download an up-to-date version of the data.
- Add timer to make sure the download feature can only be accomplished every certain periods of time because of the limited number of requests offered by the API used.

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**"