

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Main & Detail Activity and Fragment](#)

[Task 3: Implementing Skin Gallery](#)

[Task 4: Skills videos screen](#)

[Task 5: Additional libraries](#)

GitHub Username: cpinan

LoL Guide

Description

- Available in multiple languages.
- Champ information updated.
- Learn about champions.
- Save your favorites champions.
- View all skins of champions.
- Videos for each champ.

Allows you to see champions information of League of Legends without using the desktop client.

Also allow the users to save their favorites champions and check them in offline mode and see all related information from a champion, for example: History, abilities, damage, life, etc.

You can have a track of your favorites champions, see their skins, tips and stats.

Next planned releases:

- Build per champion.
- Runes and masteries simulator
- Damage per level of champion including masteries and runes.
- Champ rotation.

- News on the store.
- Game matches report.
- Order by rp, ip and type.
- Android wear watch faces.
- Notifications for news champions.

And more!!!

League of Legends Champions isn't endorsed by Riot Games and doesn't reflect the views or opinions of Riot Games or anyone officially involved in producing or managing League of Legends. League of Legends and Riot Games are trademarks or registered trademarks of Riot Games, Inc. League of Legends © Riot Games, Inc.

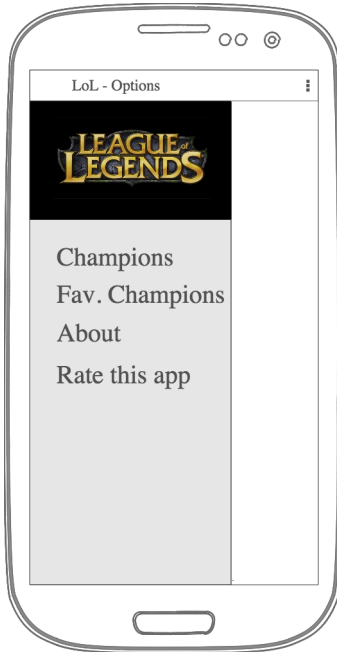
Intended User

League of legend players who want to have a mobile guide.

Features

- Search for champions for name
- Save favorite champions.
- See videos.
- Off-line mode available (Requires first data pulling)
- See price in rp and ip
- Set wallpaper
- View in HD

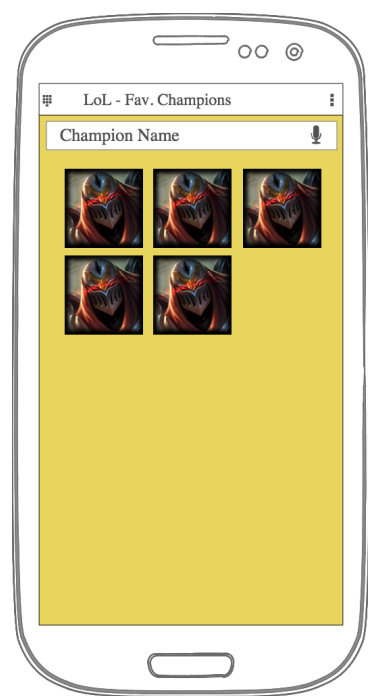
User Interface Mocks



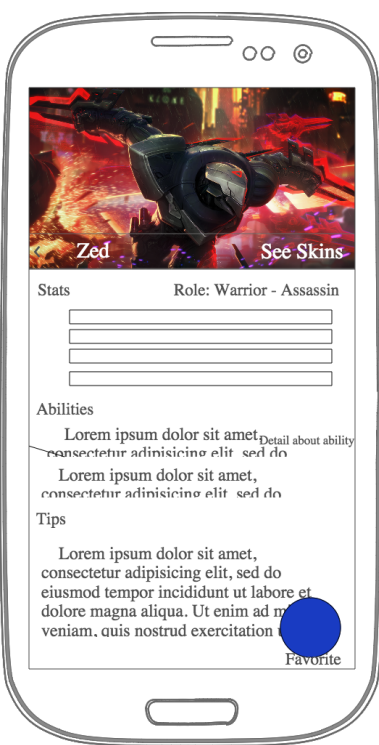
Phone - Drawer.png



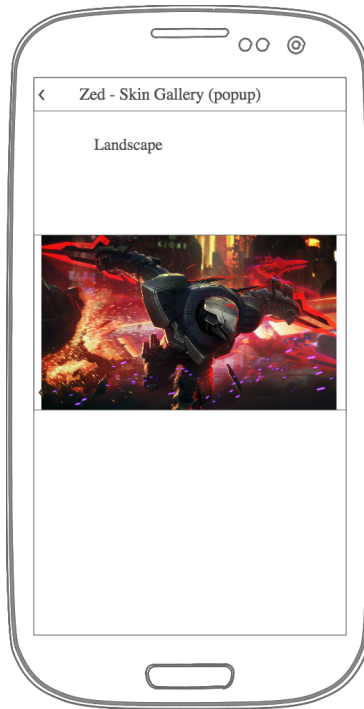
Phone - Champions.png



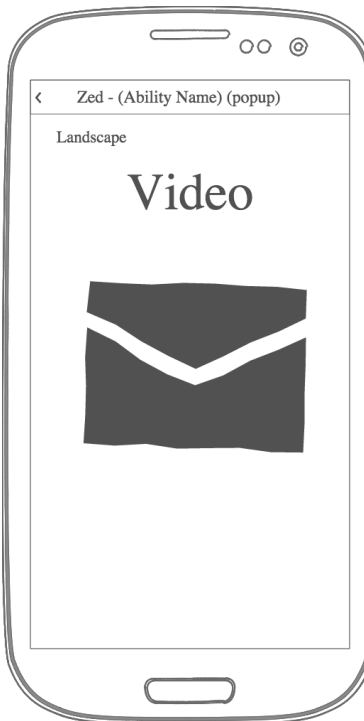
Phone - Fav. Champions.png



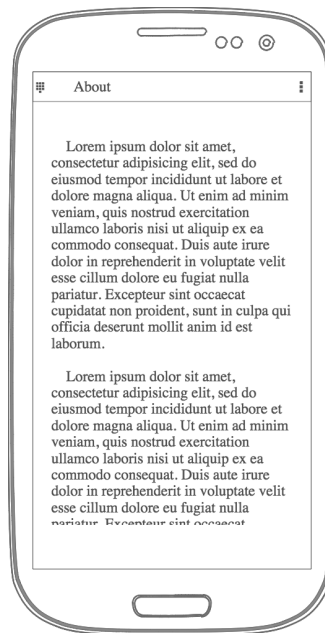
Phone - Detail Champ.png



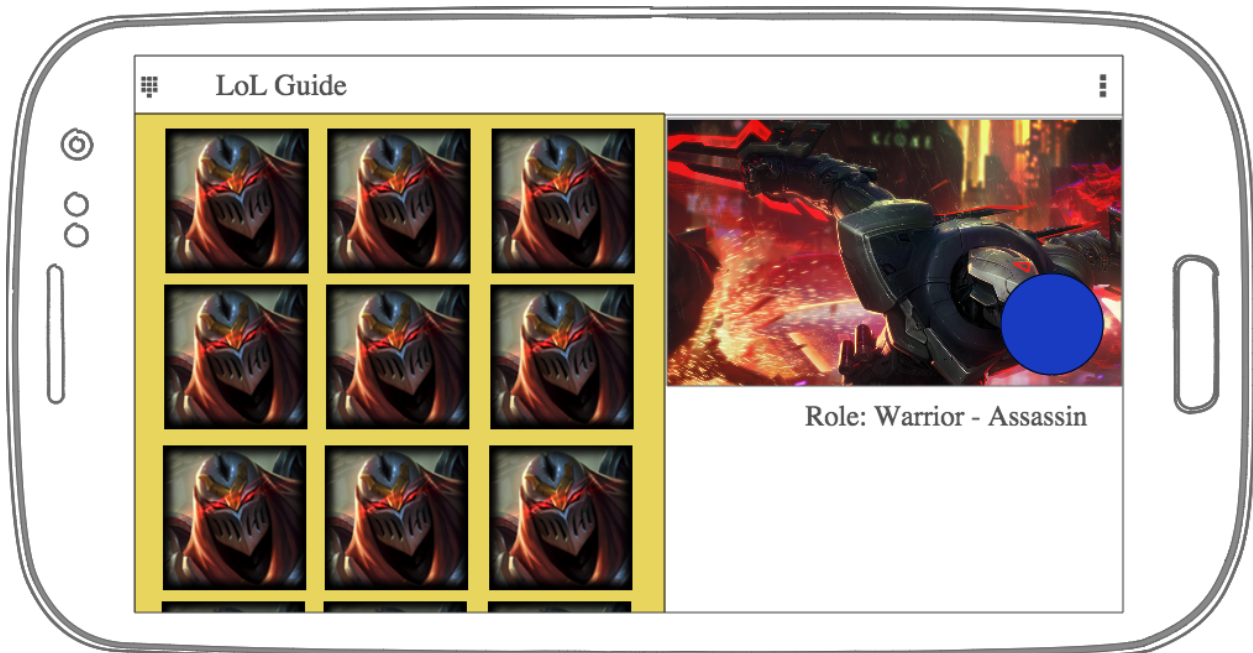
Phone & Tablet - Skin Gallery.png



Phone & Tablet- Videos.png



Phone & Tablet- About.png



Tablet - Dashboard.png

Key Considerations

How will your app handle data persistence?

I'm planning to use an external library "SugarORM" to practice, also I'm planning to use SharedPreferences and Glide. Save the images in the storage just to be available in offline mode.

Describe any corner cases in the UX.

If the user is requesting a service and press the back buttons this request is going to be paused and resumed if the user returns to the app.

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

Glide to handle loading and caching images.
Glide palette to change the colors for background.
ButterKnife to improve the views injection.
Retrofit 2.0 for services connection.
OkHttp and GSON integrated with Retrofit 2.0
Analytics to track navigation inside the app.
AdMob to show advertisement.
Crashlytics to test crash in the app in release mode.
RecyclerView, CardView, Support, Design, etc.
SugarORM for local storage.
Android Espresso for UI Test cases.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries.
- Configure local storage.
- Create test cases for services.
- Add API into gradle
- Test cases for offline mode.
- Configure Crashlytics.

Task 2: Implement UI for Main & Detail Activity and Fragment

- Build Navigation Drawer
- Build MainActivity
- Build MainFragment using MVP
- Build DetailActivity and DetailFragment using MVP
- Support for landscape and tablets.

Task 3: Implementing Skin Gallery

- Implement Activity Skin
- Implement ViewPager with TabLayout.
- Add View full image, save in memory and set as wallpaper.

Task 4: Skills videos screen

- Add VideoView and call from the server.
- Add description for each video.
- Allow gallery of videos on this screen.

Task 5: Additional libraries

- Enable AdMob
- Enable Google Analytics
- Test entire app.
- Add UI test cases with Android Espresso