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: Axolot1

Dota2Tracker

Description

Dota2Traker is an app to track dota2 matches record. User can follow famous plays or their friends dota2 account, keep up with their match history and view the full detail of a match.

Intended User

An app for dota2 player

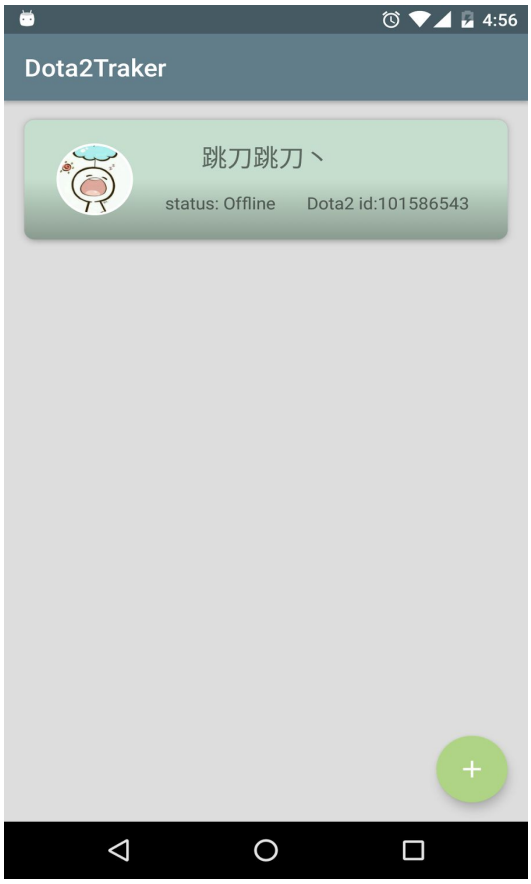
Features

- Track of player's dota2 account infomation
- View match history
- View match details, such as damage, in battle rate

User Interface Mocks

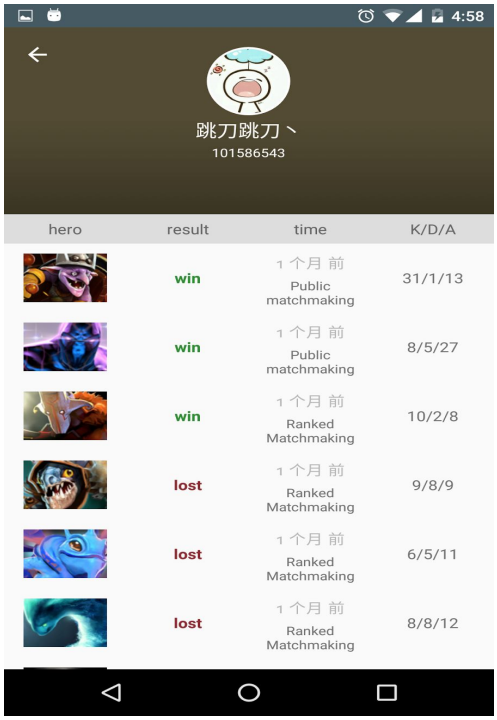
Screen 1

Main screen, for users to add the player they want to follow.



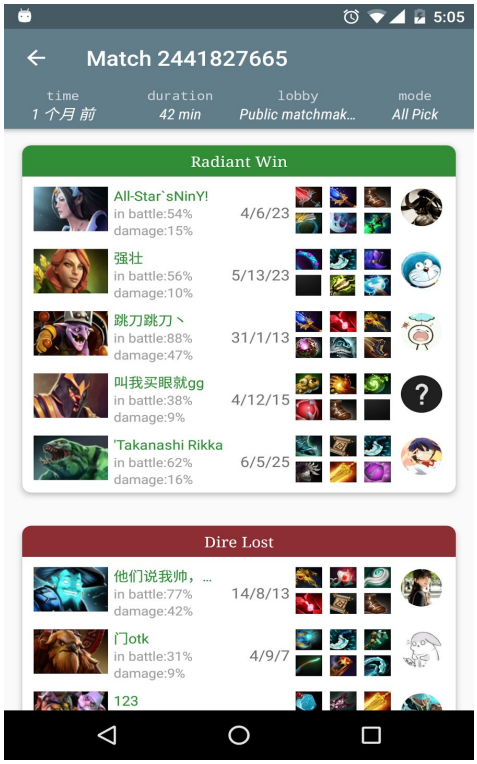
Screen 2

Player match history screen, it's show when user click the player card item in main screen.



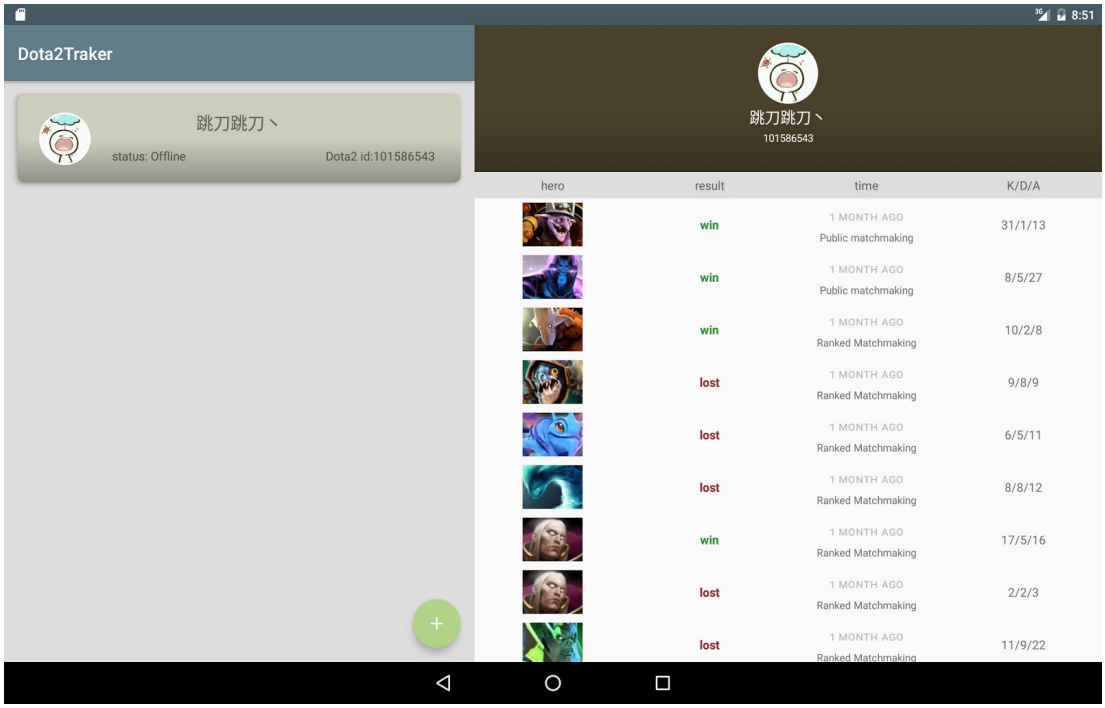
Screen 3

Match detail screen, it's show when user click the match history item in screen2



Screen 4

Tablet main scrren



Key Considerations

How will your app handle data persistence?

Save data into sqlite and build a Content Provider.

Describe any corner cases in the UX.

No corner case, just normal UX flow, when user click back button, they just return to the last screen.

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

Glide to handle the loading and caching of images.
Schematic to build Content Provider.
Dagger2 to do object injection.
Retrofit to perform http request.
ButterKnife to do view injection.
Eventbus simplifies communication between Activities, Fragments, Threads, Services.
Rxjava to add reactive extension for java and chain the http request.
CircleImageview to add circle Imageview.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries
- Test the steam api
- Plan what data should save to sqlite
- Write the dagger module and component.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for PlayerCardFragment
- Build UI for PlayerHistoryFragment
- Build UI for MatchDetailFragmnet
- Build UI for tablet

Task 3: Implement Content Provider

- Create content provider
- Save data through content provider
- Use loader to move data to view

Task 4: Implement periodic task

- Setup gcm service
- Write the periodic task logic
- Hanlde error case

Task 5: Implement other project requirement

- Add app widget
- Add Google Play Services
- Create Build Variant.