GitHub Username: yoonhok524

Time is Gold

Description

We have 24 hours in a day.
How do you spend the time?
You may spend some time in your desk to work or study.
But did you really do your best?
Maybe you can say “yes”, but my case it’s not easy.
I spend some time to check the sns, web surfing, or playing game.
And before sleep, I regret.
What did I do today!
So I want to check how do I spend the time.
And this app will help to analyze my time spending pattern.

Intended User

Everyone, who want to know how do I spend my time.

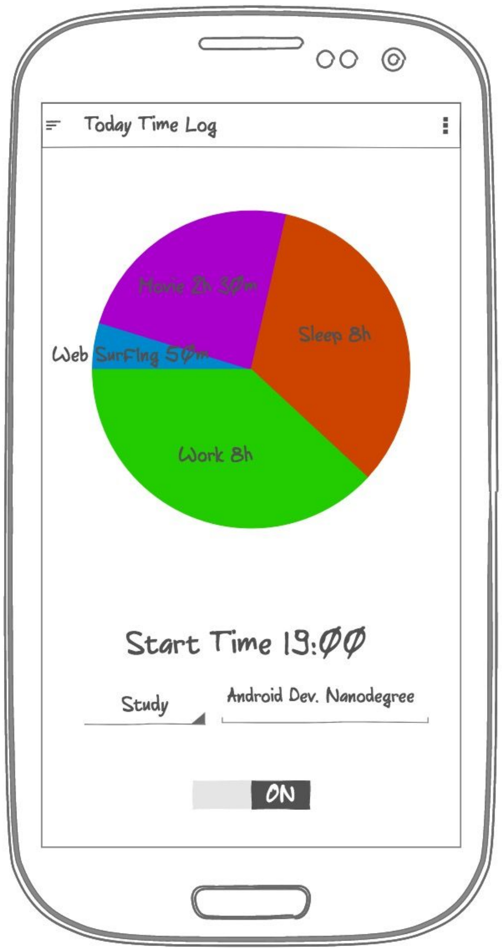
Features

- Write activities with spend time
- Analyze spending time pattern to provide statistics information to only authenticated users with google account.
- Home Widget to start/stop time log

User Interface Mocks

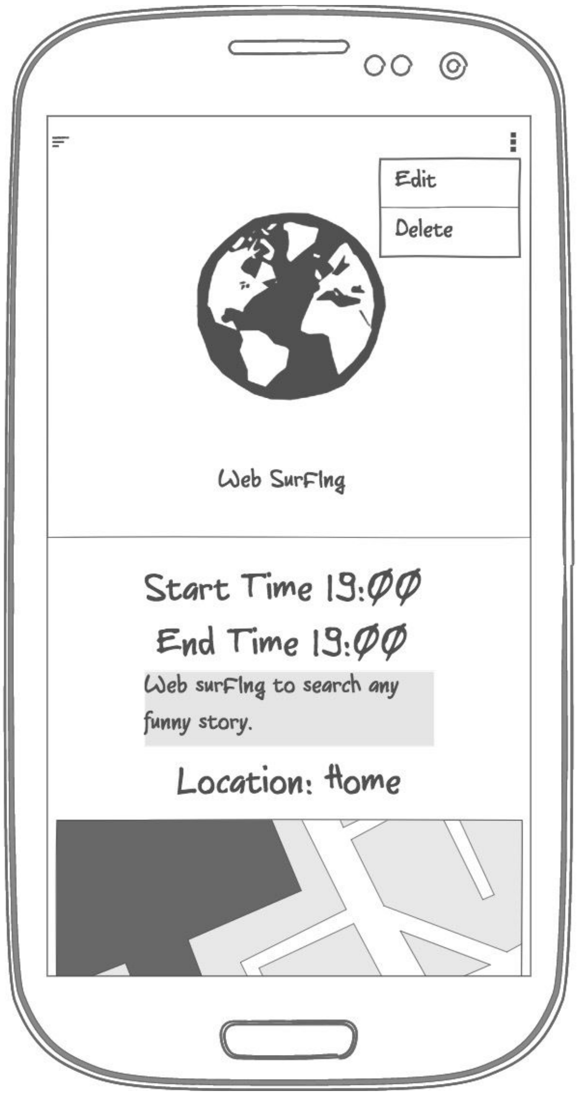
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



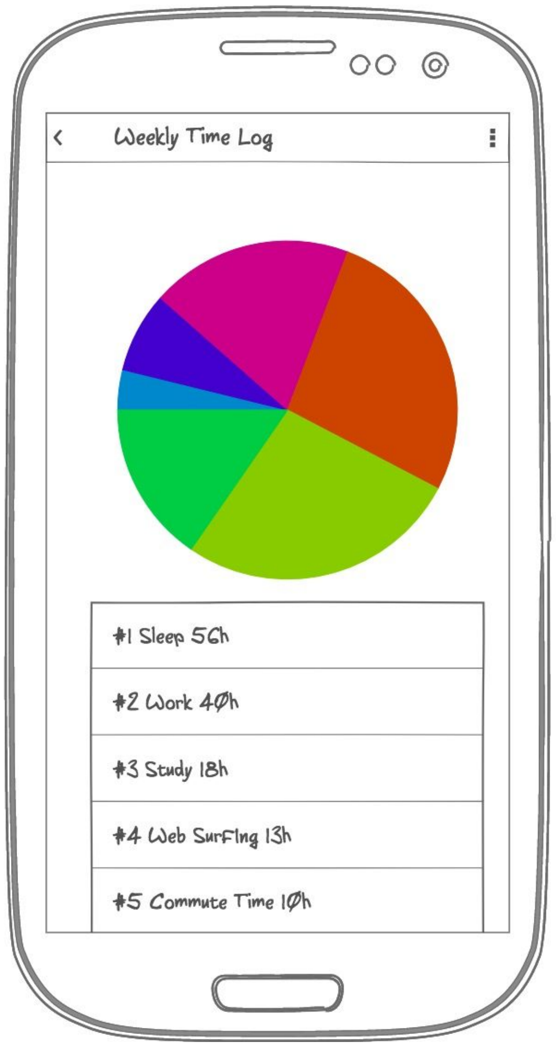
Main View to show today's time spending states.
And user can start/stop current activity using the on/off switch.
When user start a new activity, user can select a category and input the title.
(Activity doesn't mean the android component.)

Screen 2



When user click one of the activity, user can see this screen.
It displays Category, title, start/end time and location.
Through the action button, user can edit/delete this activity contents.

Screenshot 3



#1 Sleep 56h

#2 Work 40h

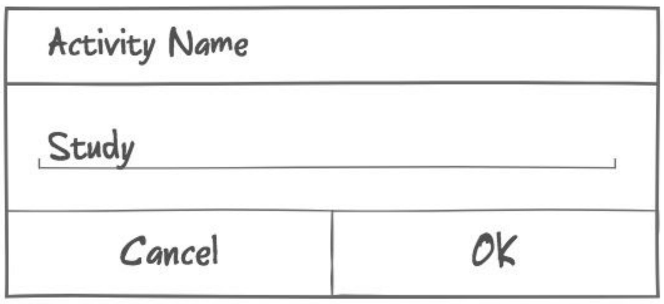
#3 Study 18h

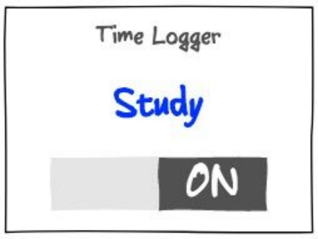
#4 Web SurFIng 13h

#5 Commute Time 10h

User can see the statistic about user's time spending pattern (weekly, monthly)
There are circle chart and list view.
Chart shows all of time spending ratio.
And list view shows the ranking of the time spending activity.
User can notice what activity is needed time.

Screen 4 - App Widget





User can start/stop the activity using the home app widget.
When user start a new activity, user can select a category.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Firebase Realtime Database, SQLite, Content Provider and CursorLoader are used to manage the data.

- Summary Statistic data will be stored in Firebase Realtime database.
  Because this kind of data is not very big, it is reasonable to store in remote db.
- Actual time logging data will be stored in SQLite. I can choose Firebase Realtime Database, but this kind of data can become very huge. And I cannot cover the price. So this raw data is stored in local, and summary data will be stored in remote.
- Statistic information will be provided to only authenticated users with google account.
- Authentication will be implemented by Firebase Auth.
- Using Content Provider and CursorLoader, data will display in the view.

Describe any edge or corner cases in the UX.

- This app will provide home widget, to start/stop the time logging.
- This app has different layout for screen size and orientation.
- If network is not available, statistic information should display with internal cache information. (But Firebase Realtime Database already support this case, so I don’t need to consider this case.)

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

- ButterKnife will be used to reduce findById code. So I can make my code clean and increase performance.
- MPAndroidChart will be used to show the various chart. It already support various chart api, so I don’t need to invent same thing.

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

- Geo api will be used to include geo information into the event.
- Firebase Realtime Database will be used to store statistic data.
- Firebase Crash Report will be used to monitor any exceptions.
- Firebase Auth, if the user want to see statistic information, they should sign in with google account.

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

- Create a github repository
- Create simple android project
- Configure libraries (include firebase)
- Define data model classes

Task 2: Define UI for Each Activity and Fragment

- Define activity/fragment classes.
- Create/Make xml layout.
- Do same thing for large screen.

Task 3: Define MVP Architecture

- Create source code files for MVP skeleton code.
  - Contract interface
  - Presenter class

Task 4: Create Data Source classes

- Create simple data source classes
  - This class have a responsibility about CRUD operate of event objects stored in Local (SQLite) and Remote(Firebase Realtime Database).

Task 5: Implements Content Provider and Loader

- Create Content Provider class
- Create CursorLoader
- AsyncTask will be implemented to access the local database.

Task 6: Implements Firebase Authentication

- Basically user can use this app without authentication. But if user want to see statistics information, they should signin with google account.
- implement signin, signout menu
- implement the conditional code for Statistics Menu. (Statistics menu should be shown only for authenticated user.)

Task 6: Write Test Code

- Write junit test code
- Write espresso code

Task 7: Implement View/Presenter classes

- Implement View and Presenter classes.
- Implement same thing for large screen.

Task 8: Implement home widget

- Implement home widget.