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: gecsko

Mini Diary

Description

Mini Diary app is a mini logging for our daily life. User can use the app to take a mini note at any meaningful moment. In every logging, user can take note to describe the moment and their feeling. Also, the app will auto log the user current location and weather for every log. User can review their mini diary at anytime in everywhere.

Intended User

Intended user is who want to log their life moment, such as : teenager to log their school life and young lady to log their favorite restaurant or shopping mall.

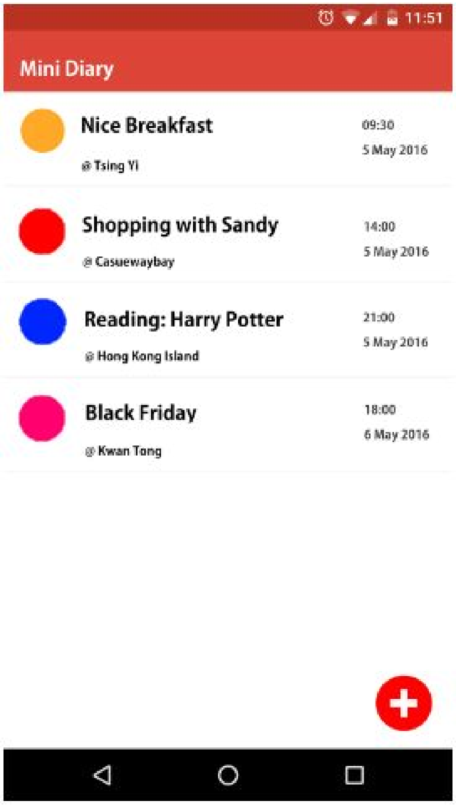
Features

- Save user moment description
- Auto fill in user current location
- Auto fill in weather condition
- User pick a color to represent their mood at the moment

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 Photoshop or Balsamiq.

Screen 1 : Main View



11:51

Mini Diary

Nice Breakfast
@ Tsing Yi
09:30
5 May 2016

Shopping with Sandy
@ Casuewaybay
14:00
5 May 2016

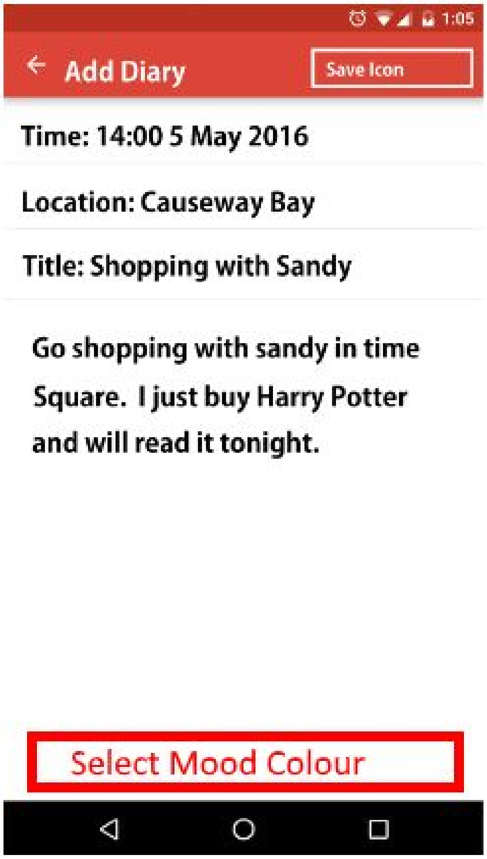
Reading: Harry Potter
@ Hong Kong Island
21:00
5 May 2016

Black Friday
@ Kwan Tong
18:00
6 May 2016

This is the main view of the app. This view use ListView to display added mini diary Info and user can press FAB to intend to “Add Diary Activity” and add new mini diary.

Mini Diary Info shows Diary title, location, time, date and mood colour in circle.

Screen 2 : Add Diary Screen



Time: 14:00 5 May 2016

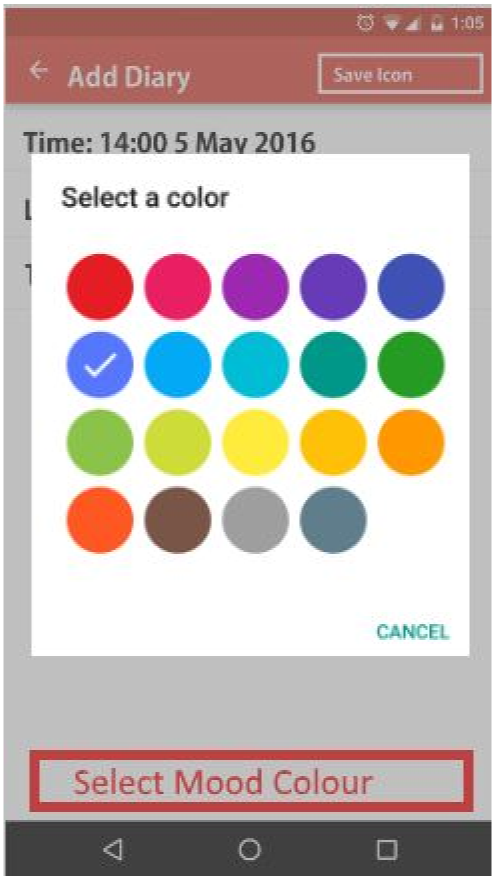
Location: Causeway Bay

Title: Shopping with Sandy

Go shopping with sandy in time Square. I just buy Harry Potter and will read it tonight.

This is a add new mini diary screen. App can auto load date and time from system and load location from web weather API. User can note down the diary title, content and select mood colour for this diary by colour dialog.

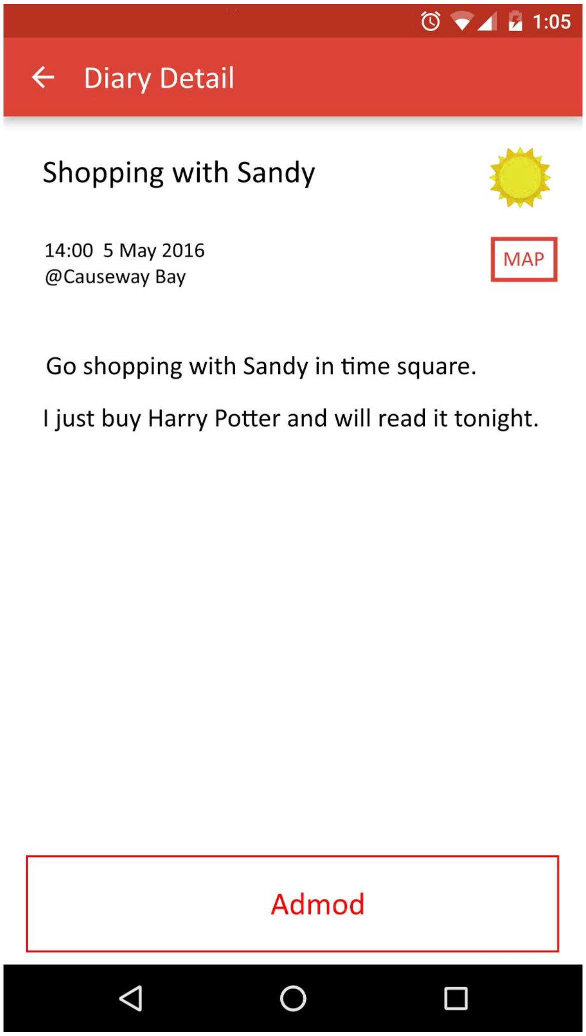
Screen 3 : Select Mood Colour Dialog



Time: 14:00 5 May 2016

This is a mood colour selection dialog. After user clicked select mood colour button in add diary screen, colour selection dialog will pop up. User can select a colour to represent their mood in writing diary moment.

Screen 4 : Mini Diary Detail Screen



Shopping with Sandy

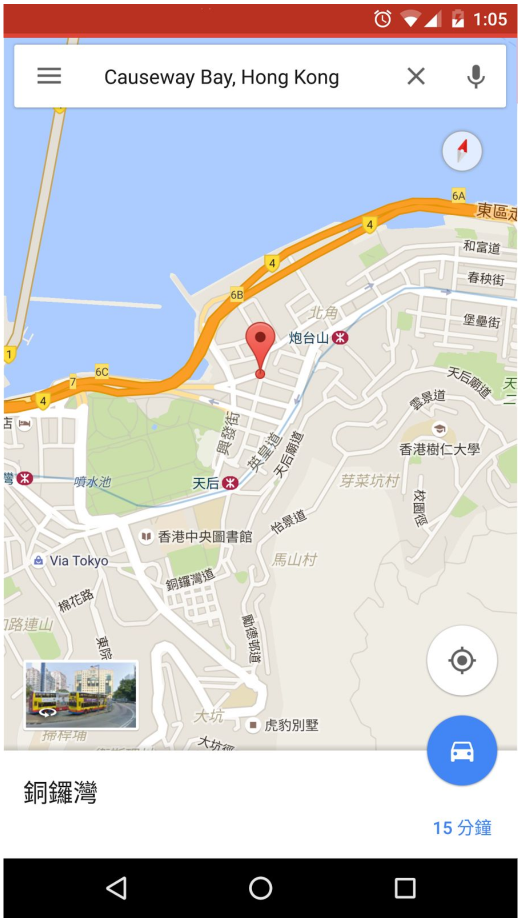
14:00 5 May 2016
@Causeway Bay

Go shopping with Sandy in time square.

I just buy Harry Potter and will read it tonight.

When the user select an item on main screen list views, the detail mini diary screen will display the corresponding diary log. The content include title, time, date, location, weather and description. The text colour will follow the mood colour of that diary. All data get from content provider. And user can press the map button direct to display the location on google map.

Screen 5 : Display Map with logged location



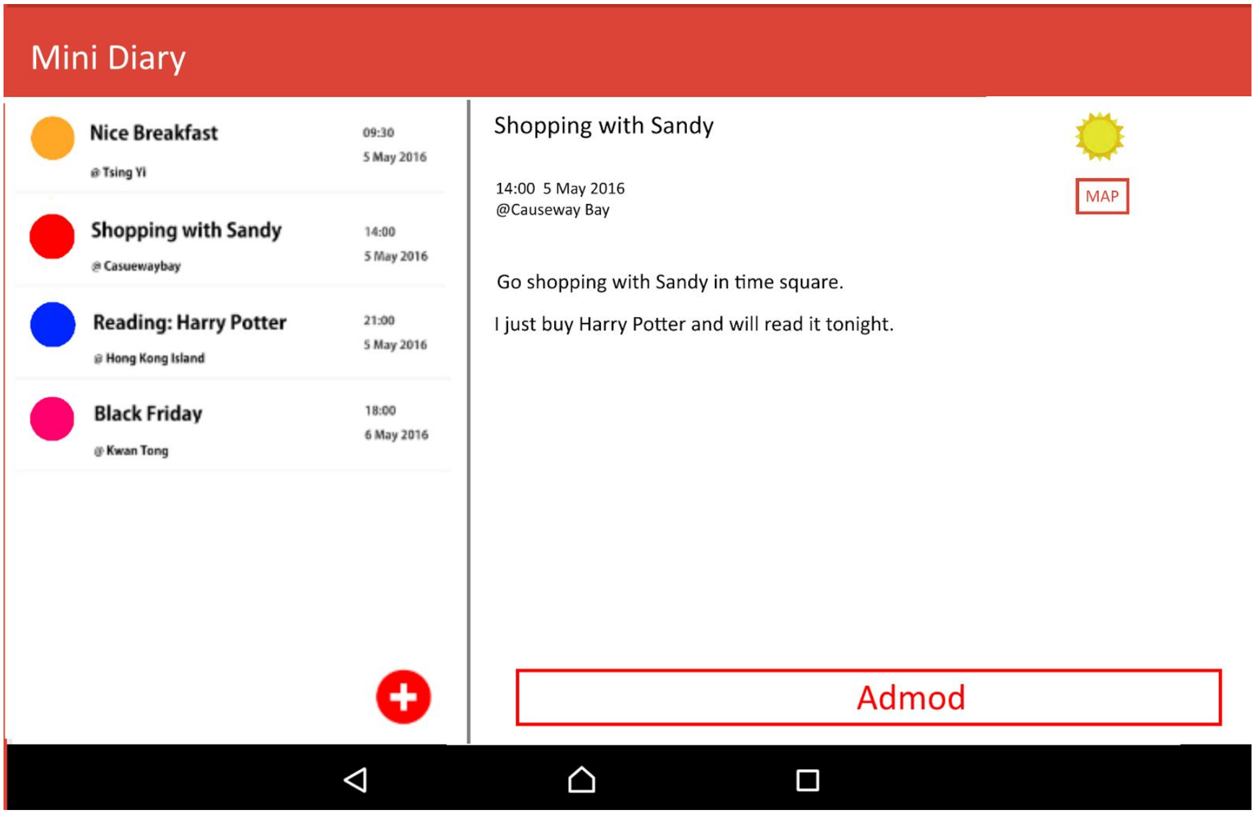
Causeway Bay, Hong Kong

銅鑼灣

15 分鐘

When Map button clicked on detail mini diary screen. Google map will be intended and show the logged location.

Screen 6 : Alternative Screen for tablet\large Screen



Nice Breakfast	@ Tsing Yi	09:30 5 May 2016
Shopping with Sandy	@ Casuewaybay	14:00 5 May 2016
Reading: Harry Potter	@ Hong Kong Island	21:00 5 May 2016
Black Friday	@ Kwan Tong	18:00 6 May 2016

Shopping with Sandy

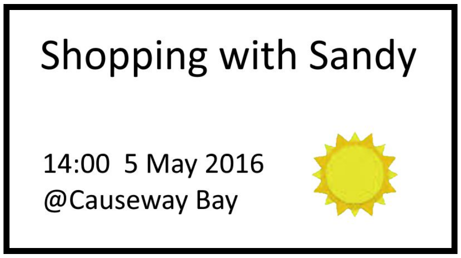
14:00 5 May 2016
@Causeway Bay

Go shopping with Sandy in time square.

I just buy Harry Potter and will read it tonight.

This is a view for larger screen display. The main screen shows on left and the detail screen will show on right. If MAP button is clicked, the detail screen will be replaced by map screen. When FAB “add button “, the add mini diary will display on right screen.

Screen 7 : Widget



Shopping with Sandy

14:00 5 May 2016
@Causeway Bay

This is a view for widget display. Widget will show the newest mini dialog which contain title, date, time, location and weather.

Key Considerations

How will your app handle data persistence?

Content Provider is used with sqlite as backend. All data is stored in SQLite. A table column include : Data, Time, location, Weather, Mood Colour, title and Description. Each row for one mini diary.

Describe any corner cases in the UX.

The app is able to work in offline condition. Even though the app is not able to capture weather information from web API or user location, user can still be able to add their diary.

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

A colour selection library, called spectrum, is used in the app for providing colour dialog for user to select their mood colour. This library provide a simple and clear UI for user to select colour.
Url: https://github.com/the-blue-alliance/spectrum

Volley may be used to handle network traffic.
Url: https://android.googlesource.com/platform/frameworks/volley

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

Create New Android Project with below setting:

API Level : API23
Min SDK : API15

Include Library :

- Spectrum Library
- Volley

- Design support library
- Google service API (Admob)
- Google service API (Google Map)

Task 2: Implement UI for Each Activity and Fragment

Build Each UI for Different Activity or Fragment

- Build UI for MainActivity
    - Named activity_main.xml
    - Use One Fragment Layout Design for sw < 600dp
    - Use Two Fragment Layout Design for sw >= 600dp

- Build UI for MainFragment
    - Inflated by Main Activity to Display added Mini Diary Item
    - Named fragment_main.xml
    - Construct Fragment Layout using Coordinate Layout Design include List View and FAB
    - List View use list_view_item.xml to construct each item structure

- Build UI for Main List View
    - Inflated by Main Activity to Display each listview item
    - Named list_view_item.xml
    - Include 3 Text Views
        - Title
        - Time + Date
        - Location
    - Include 2 image Views
        - Weather icon
        - Mood Colour icon

- Build UI for Detail Activity
    - Named activity_detail.xml
    - Include fragment_detail.xml

- Build UI for Detail Fragment
    - Inflated by Detail Activity for display detail mini dialog
    - Named fragment_detail.xml
    - Include 4 Text Views
        - Title
        - Time + Date
        - Location

        - Content
    - Include 1 Image View
        - Weather icon
    - Include 1 Button View
        - Intent to Google Map
    - Include 1 Admod display

- Build UI for Add Diary Activity
    - Named add_diary_activity.xml
    - Include add_diary_fragment.xml

- Build UI for Add Diary Fragment
    - Inflated by Add Diary Activity for user edit diary content
    - Named add_diary_fragment.xml
    - Include 2 EditText
        - Title
        - Content
    - Include 2 TextView
        - Location
        - Date + Time
    - Include 1 ButtonView
        - Intent to Mood Colour Selection Dialog

- Build UI for Widget
    - Named widget.xml
    - Include 3 Text Views
        - Title
        - Time + Date
        - Location
    - Include 1 Image View
        - Weather icon

Task 3: Data Handling

Design and Implement Data flow.

- Content Provider
    - Setup SQLite Database to handle mini dialog info detail
    - Content Provider is created to handle Database access based on data request from APP

- Weather Info

    - Get a current weather via intend service from web Weather API
    - Store Weather info to Database via Content Provider

- Date + Time Info
    - Get from system time and date
    - Store Date + Time info to Database via Content Provider

- Location Info
    - Get from system info
    - Store location info to Database via Content Provider

Task 4: Implement Logic

Implement Logic in the APP

- MainActivity
    - Use Fragment transit to
        - MainFragment for sw < 600dp
        - MainFragment + other related fragment for sw >= 600dp

- MainFragment
    - Show added diary item by list view
    - Use Adaptor to match related item info from content provider to list view item
    - Create FAB listener
        - Intend to AddDiaryActivity when user click FAB
    - Create listview listener
        - Intend to DeatilActivity when user click the corresponding item

- DetailActivity
    - Use Fragment transit to DetailFragment

- DetailFragment
    - Show the selected diary info
    - Intend to Google Map when Map Button is clicked

- AddDiaryActivity
    - Use Fragment transit to AddDiaryFragment

- AddDiaryFragment
    - Get current Data + Time from system and display on TextView
    - Get current Weather from Weather API and display on ImageView

  - Intend to Spectrum Library to show colour selection dialog for user select mood colour and feedback the colour info
  - Check the inputted info is filled or not when user click save button
  - Store all info via Content Provider to database when checking pass

- Widget
  - Get the newest info via Content Provider from Database
  - Intent to Main View when Widget is clicked

Task 5: Testing

Testing APP each functionality

- Main Activity
  - Test list view can show each item info
  - Press list view item to intend detail activity
  - Press FAB to intend add diary activity

- Detail Activity
  - Test all info can display
  - Press Map Button to intend Google Map

- Add Diary Activity
  - Test Date + Time can display
  - Test location can display
  - Test Alert when no internet connection
  - Test Empty Edit text Alert when save diary
  - Test Colour Dialog function

- Widget
  - Test Widget can display the newest diary info
  - Press Widget to intend Main Activity

- General
  - Handle Orientation Change
  - Handle No Network connection
  - Handel No Google Map installed in Device

Task 6: Final Modification

Final Modification for optimization the APP outlook design

- Colour
- Text Font
- Text Size
- View structure, such as: padding, margin
- Button position
- and etc.

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**"