TEAM PHOENIX

UBC CPSC 410 Project – Ever Friend

Guides to CPSC 410 Project Mandatory Sections (User's Manual)

UBC CPSC 410 PROJECT – EVER FRIEND

Guides to CPSC 410 Project Mandatory Sections (User's Manual)

Team Phoenix, November, 2012
UBCTeamPhoenix@groups.live.com
https://github.com/UBC-CS410/Project-Phoenix

Table of Contents

Introduction

From adding your friend from Twitter to make comment on their geo-located status, making social connections in Ever Friend are just a few clicks away.

Thank you for choosing Ever Friend.

With Ever Friend you can keep up with friends, wherever you are.

Keeping up with friends is faster than ever.

- See what friends are up to and show them on the map
- Share updates
- Leave comments on others status

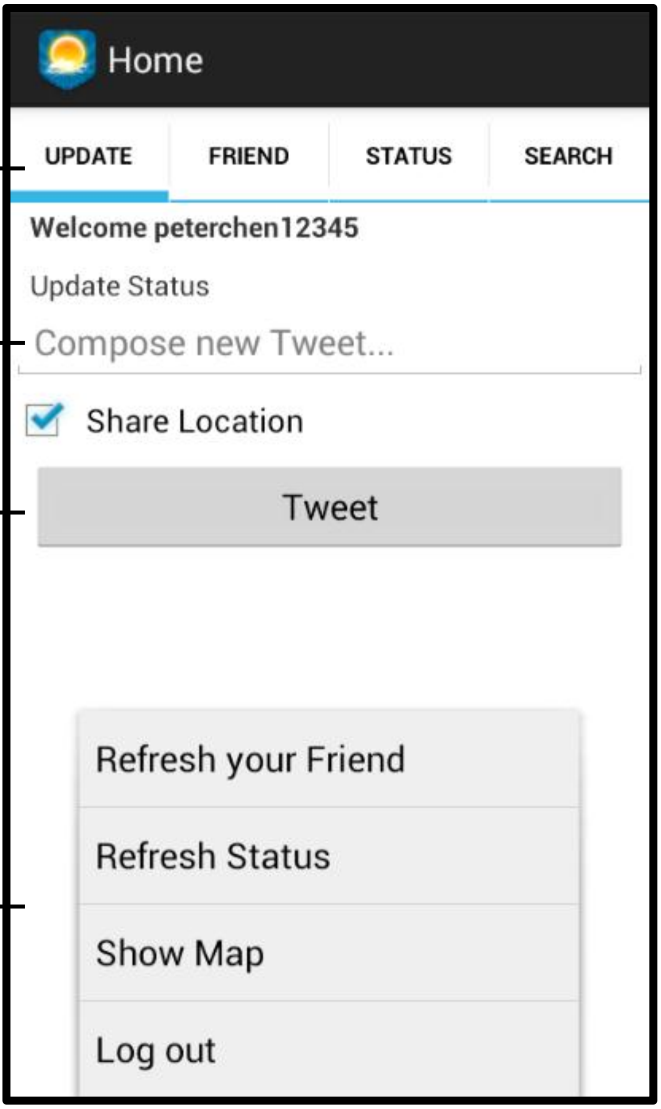
Ever Friend is a free lightweight user-friendly mobile social network. User can update status, make comment and see other people's status and their location on Google map.

Chapter
1

Getting to Know Your App

Thank you choosing Ever Friend. Please be sure to read all instructions thoroughly to get the most from the app, and have your phone connected to the Internet.

Main Screen

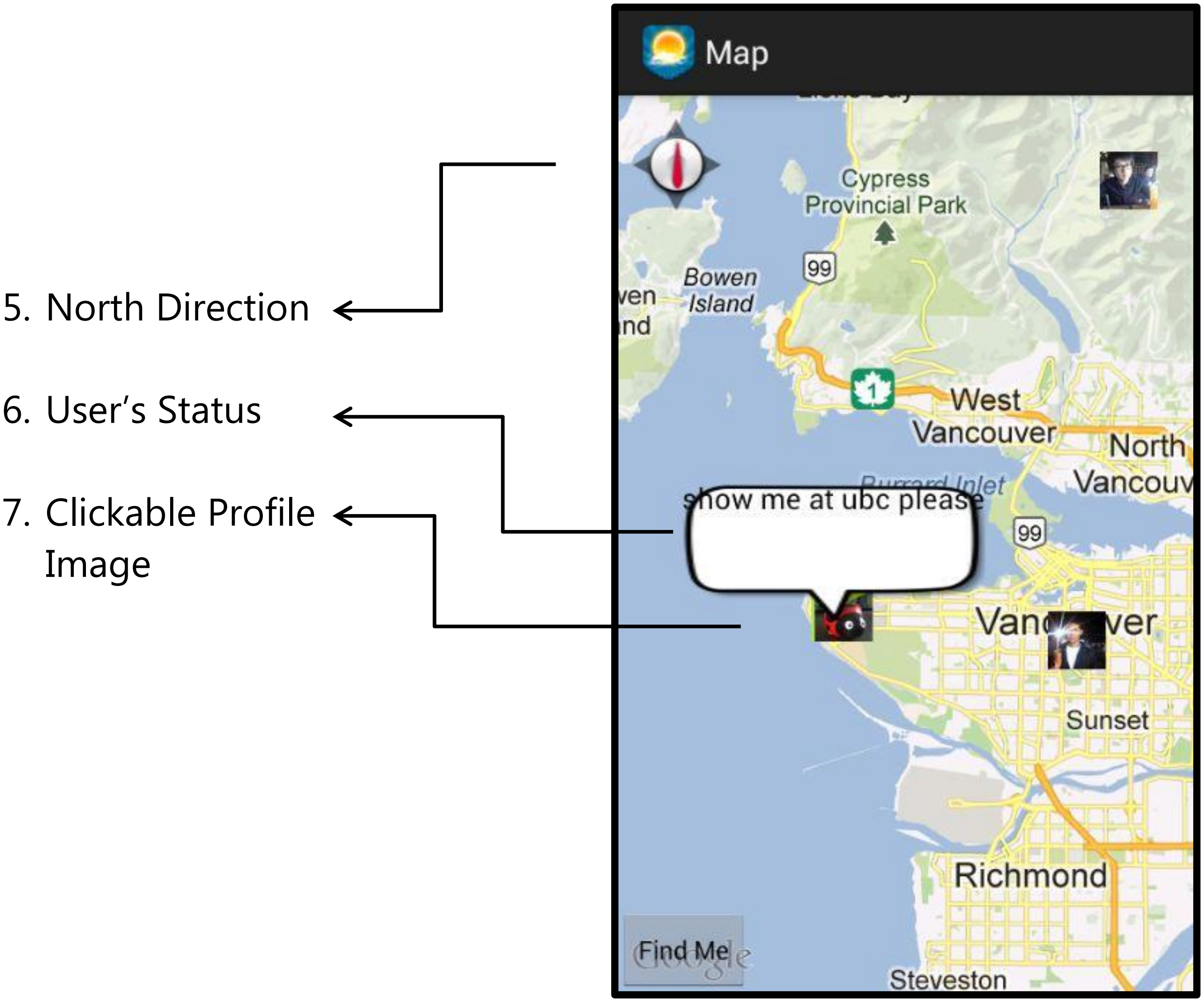


Welcome peterchen12345

1. Tab Switcher
2. Text Input Field
3. Action Button
4. Popup Menu

☑ **Note:**
To show the popup menu, press hardware or Android system "Menu" button.

Map View



show me at ubc please

5. North Direction
6. User’s Status
7. Clickable Profile Image

Chapter

2

Tutorial

This section describes how to use Ever Friend Android app, how to ready the app for use and how to add your first friend and update your status.

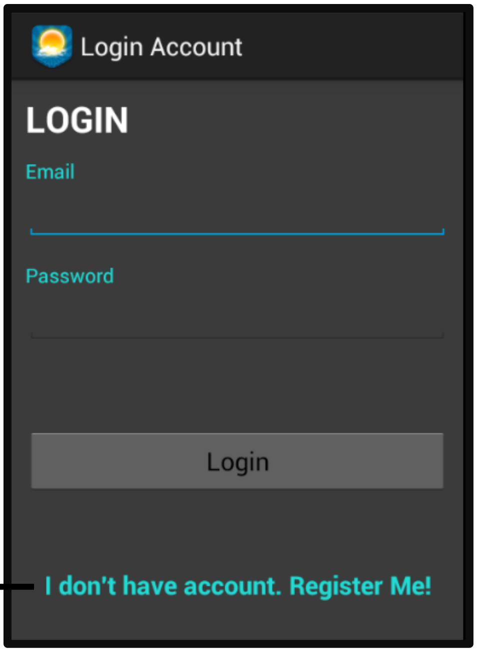
2.1 Register Your First Account

To use Ever Friend, you first need to register an account.

Follow the steps below to navigate the registration process.

1 Go to Register Page

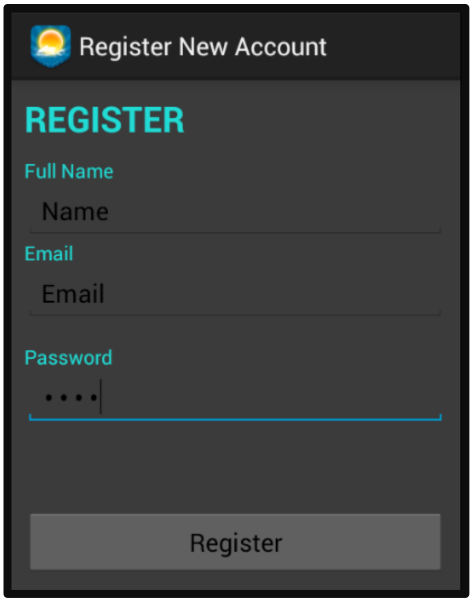
Press the "**Register Me!**" button on the button of the Login screen



Register Me button

2 Enter Your Information

Enter your name, email and password. Press "**Register**" button to finish.



Register New Account

REGISTER

Full Name

Name

Email

Email

Password

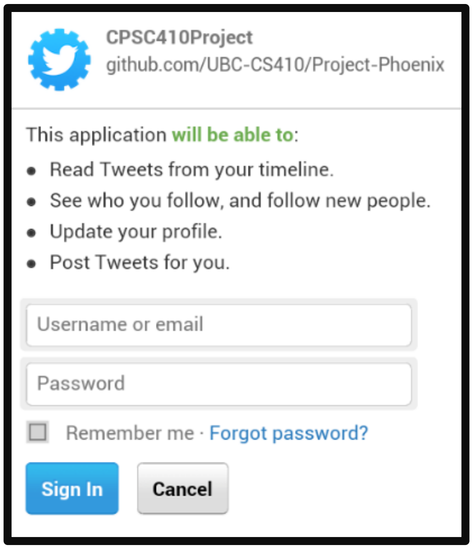
••••

Register

---

3 Sign in Your Twitter

Your will be directed to browser to login your Twitter. Once you have finish, you are all set and ready to use Ever Friend.



CPSC410Project
github.com/UBC-CS410/Project-Phoenix

This application will be able to:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

Username or email

Password

□ Remember me · Forgot password?

Sign In | Cancel

Note:

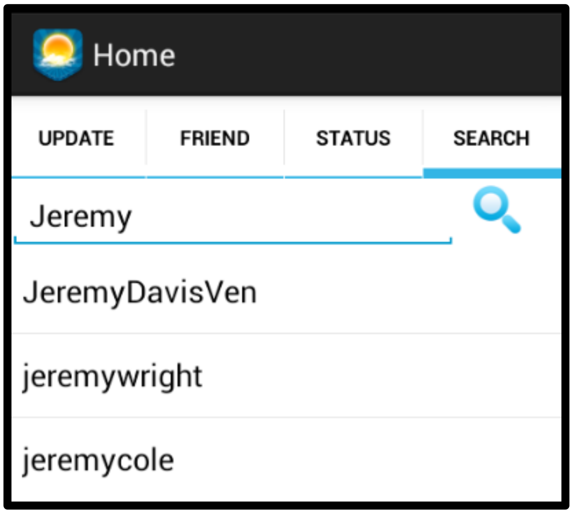
1. Currently, Ever Friend 1.0 only support Twitter login, user is required to have a Twitter account to use Ever Friend.
2. To switch Twitter account after you login, please clear your browser's data first, or choose not to let the browser to remember your account.

2.2 Add Your First Friend

Add your friend into Ever Friend. You will be able to see what they are up to. Follow the steps below to find and add your first friend.

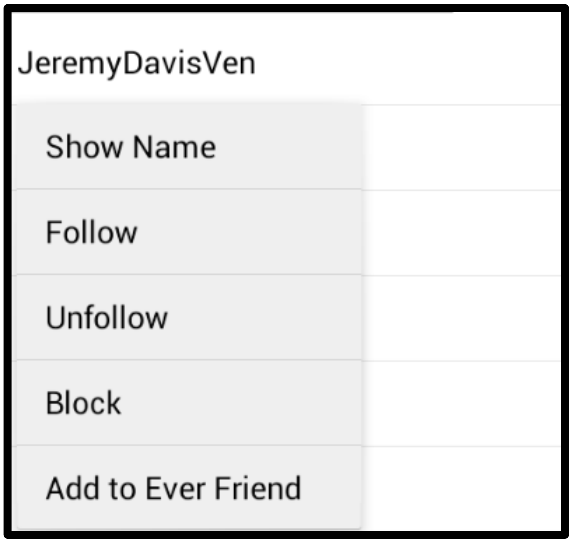
1 Search Your Friend

Go to "**Search**" tab, enter the name of your friend and click the "**Search**" button on the right.



2 Add Your Friend

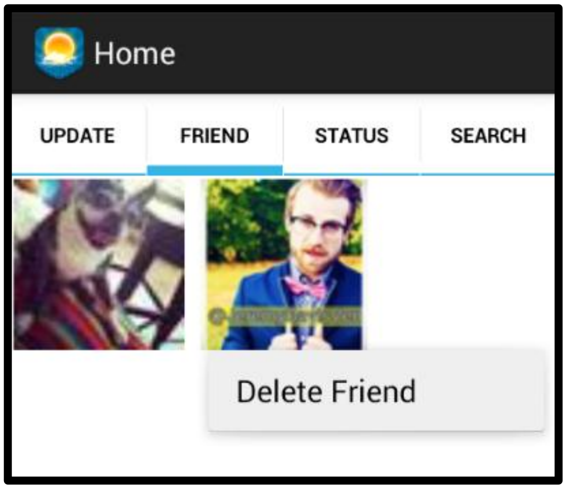
Click the person you want to add, a popup menu will show up. Click "**Add to Ever Friend**"



3 See Your Friend

Go to “**Friend**” tab to see your friends you just added. Now they will be able to see your status updates on their Ever Friend app.

To delete your friend, click on their profile image, and click “**Delete Friend**”

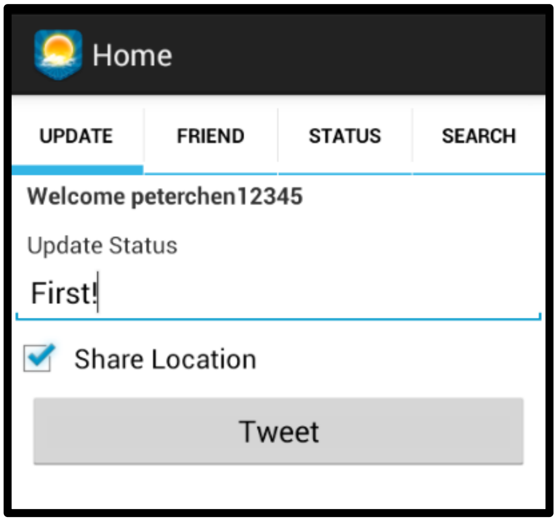


2.3 Update Your Status

Update your status so that your friends can see them too.

Follow the steps below to post your first status.

1 Write and Post



Welcome peterchen12345

Update Status

First!

☑ Share Location

-Go to "**Update**" tab and compose your first status.

-By default, your status update will be shown on the map, and your friend can see them. If you do not want that, uncheck the "**Share Location**" box in the middle.

Click "**Tweet**" button to send.

Note:

Do not be confused by the word "Tweet". It will post your status to both Twitter and Ever Friend.

2.4 Write Comment

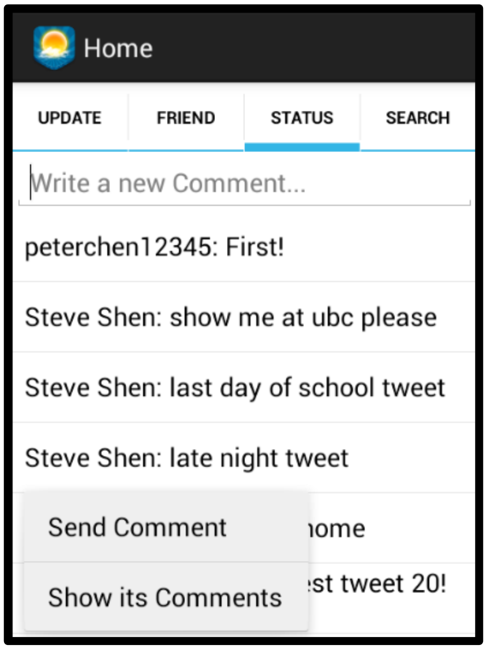
You can comment on other's status (also yours); if you keep going, you can even have a chat.

Follow the steps below to post your first comment.

1 Choose the Status You want to comment on

Go to "**Status**" tab and write comment before you click any status.

After you have written your comment, click the status you want to comment on and click "**Send Comment**" button from the popup menu.



peterchen12345: First!

Steve Shen: show me at ubc please

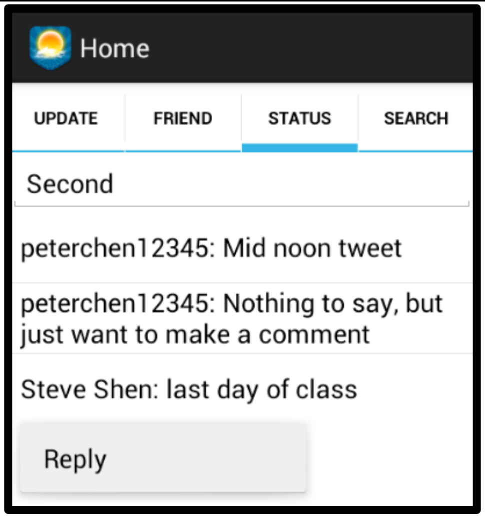
Steve Shen: last day of school tweet

Steve Shen: late night tweet

2 See your comment and keep chatting.

Click “**Show its Comments**” button from the same menu you just click.

You can keep sending comment. Just write a new comment and click any comment on the screen and click “**Reply**”.



Second

peterchen12345: Mid noon tweet

peterchen12345: Nothing to say, but just want to make a comment

Steve Shen: last day of class

2.5 Map View

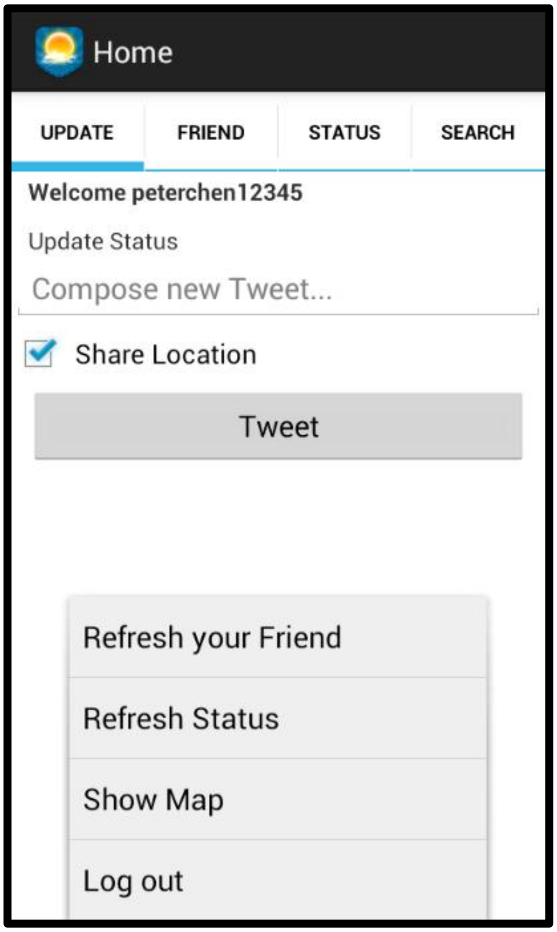
You can also see your friend's latest status on the map.

Follow the steps below to show the Map view.

1 Under Any Tab, Press Hardware or Android System "Menu" Button.

A popup menu will show up.

Click "**Show Map**" from the menu.

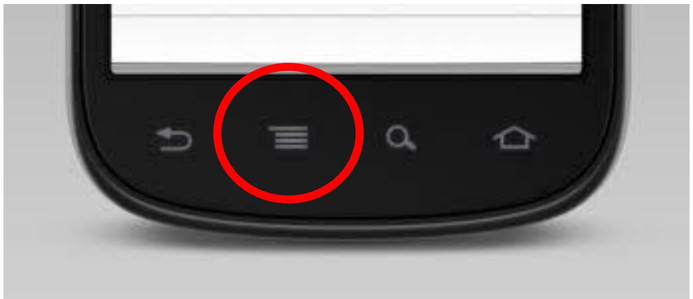


Welcome peterchen12345

Update Status

Compose new Tweet...

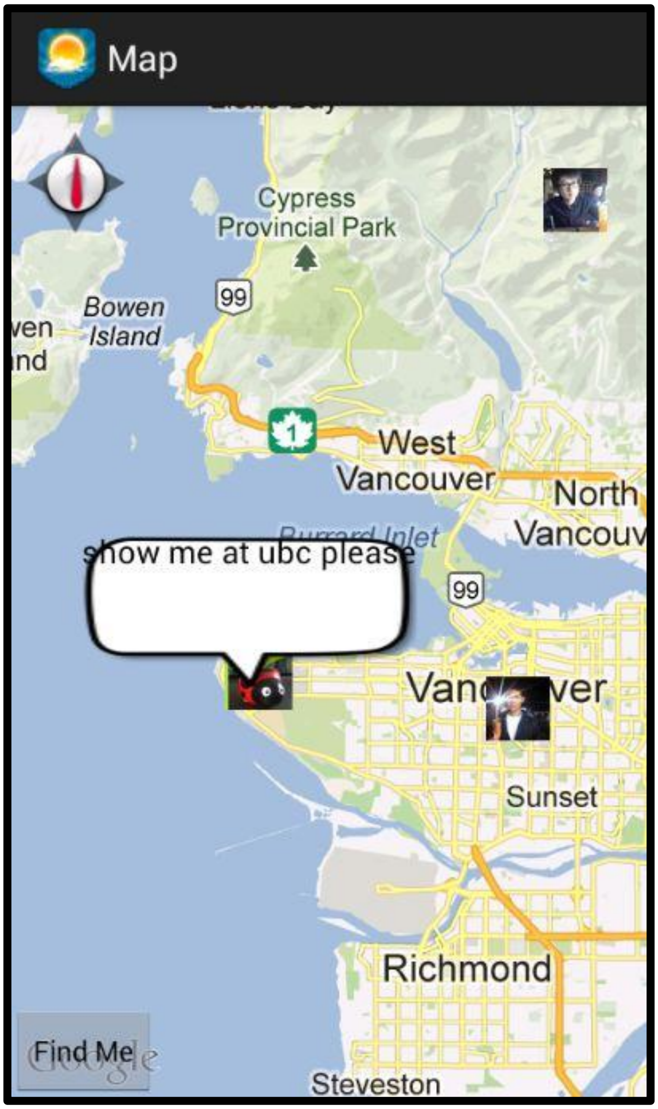
☑ Share Location



2 See Other's Status on the Map

Simply click on their icon, their status will pop up right away.

You can switch to Satellite view. Just select "**Satellite View**" from the popup menu.



show me at ubc please

Chapter

3

Mandatory Features

This section describes how to me meet mandatory features that are required in CPSC 410 project page.

3.1 Updates

1. Each Ever Friend user is able to search and add friend into our server system, and also delete them form our server system if they want to. Therefor Ever Friend meets the requirement of "Users must be able to subscribe/unsubscribe to a subset of content on the site e.g. Watch, Like, Friend, Follow, etc…"

2. Each Ever Friend is also able to post status in the "Update" tab, the status will not only send to Twitter, but also be stored into our sever database. Their friend will able to see the status from other Ever Friend client. Therefor Ever Friend meets the requirement of "Users must be able to contribute and update content"

3. Ever Friend will automatically refresh status and friend page after user clicks on tabs and menu items. Also, we have a dedicated database table to store users' latest status. Therefor Ever Friend meets the requirement of "Users should never see stale data"

4. Ever Friend stores users' relationships using two unique identifiers: user's Twitter ID and their Ever Friend's friends Twitter IDs. For example user 1 adds user 2, 3 as friend. User 2 adds user 1, 3 as friend. The table look like this:

User ID	Friend ID
1	2
1	3
2	1
2	3

When both users delete friendship with 3, Ever Friend simply delete the second and fourth rows, the data is not disturbed. Therefor Ever Friend meets the requirement of "Implementation must be free of concurrency bugs"

3.2 Integration

Ever Friend uses two of popular web API: Twittter4j and Google Map API. Also we run our own server using Apache, MySQL and PHP.

TWITTER4J Google maps

Twitter4J API let us able to search people, retrieve Twitter ID and profile images send Tweets. Google Map API let us able to pin user's locations and their status on the map. Those APIs are imported into our code so they are never just a page we link to or embed.

Therefor Ever Friend meets the requirement of

“-Your application will use at least 2 Web services

-Web services must provide API or REST-based service

-Must not just be a page that you link to or embed

-Web services must be relevant to your application”

3.3 Location-Awareness

1. One of the best feature of Ever Friend is that user can post status on the map. However users can deselect the “Share Location” option, then their status’ location will not be stored into our database. Therefore Ever Friend meets the requirement of “Your application must leverage the user’s location preferences, subject to privacy choices”.

2. Ever friend will alter user when opening the map view if they disable the GPS. Our map view will constantly follow user’s current location. If user gets lost in the map view, there is always a “Find Me” button in the menu. Therefore Ever Friend meets the requirement of “Your application’s interface should reflect where the user currently is in (near) real-time”.

Chapter

4

Security

Ever Friend use two layers login system. The first layer is our own login system running on own server. Our server and MySQL database is protected with completed password. Each user's password is stored in encrypted code which is very hard to decode for unauthorized access.

The second layer of login in is Twitter via OAuth 2.0. OAuth is a widely used secure authorization. When user login into Twitter, Ever Friend requires OAuth to send back a confirmation URL in order to continue.

Chapter

5

Separation of Concerns

Our SoC strategy is to separate client and server apart, so that server acts as a middleware and provides services to clients. So that client will not have to worry about data handling. Our server is a WAMP server with MySQL-PHP as our data storage mechanism.

In client side, we separate activity classes, and helper class into different packages. So that our page creation code is separate from other code which enhances the modularity of our application.

Chapter

6

Evolution

Our original design was to create a social network photo manager. Our current design is to create a social network. So there are many changes to our original design. Please see the evolution document (2.2_New_Design Documents (Evolution).docx) for further details.

Chapter

7

Testing

We use Robotium as our code test framework. We have wrote test cases based on different scenarios. Which cover all the functionalities that users might perform. However we could not get the automatic test coverage tool working for our project. But we did a manual calculation, and our test cover every functionalities except Map which should be well over the required 60%.

We also have a paper based acceptance test suit. Please see “410 Paper based acceptance test suit.docx” for further details.

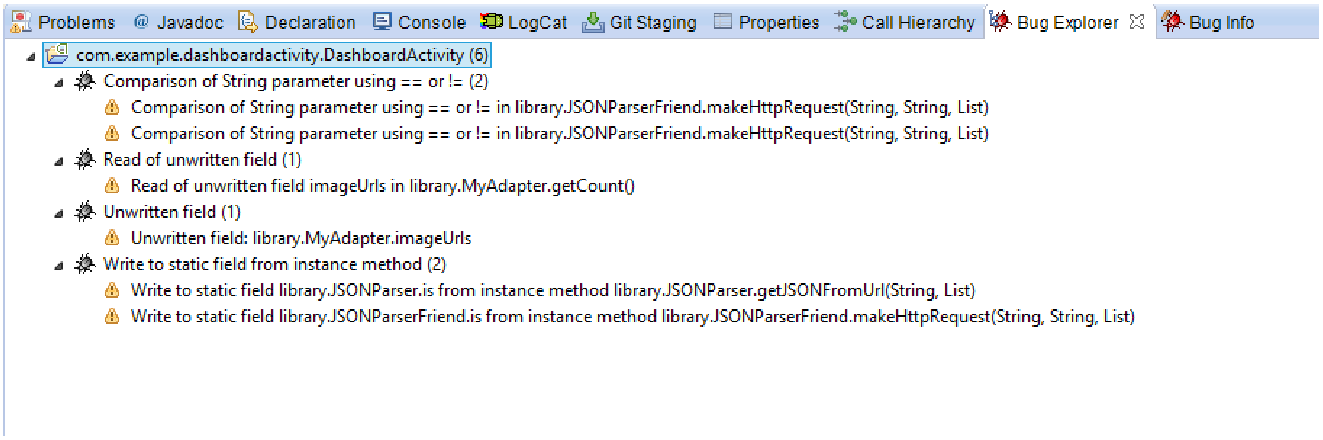
Chapter

8

Verification

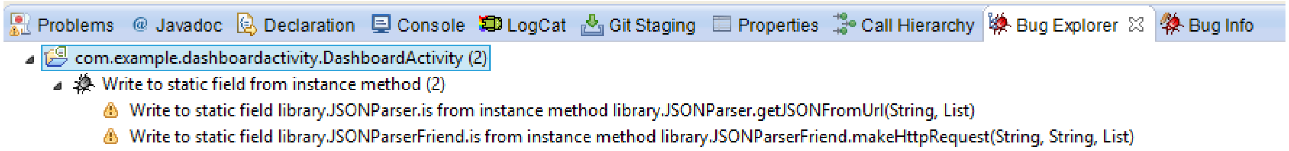
We use “Find Bug” as our static analysis tool, and we fixed many meaningful bugs. Here is the screen shots of the result:

First Run:



```
com.example.dashboardactivity.DashboardActivity (6)
    Comparison of String parameter using == or != (2)
        Comparison of String parameter using == or != in library.JSONParserFriend.makeHttpRequest(String, String, List)
        Comparison of String parameter using == or != in library.JSONParserFriend.makeHttpRequest(String, String, List)
    Read of unwritten field (1)
        Read of unwritten field imageUrls in library.MyAdapter.getCount()
    Unwritten field (1)
        Unwritten field: library.MyAdapter.imageUrls
    Write to static field from instance method (2)
        Write to static field library.JSONParser.is from instance method library.JSONParser.getJSONFromUrl(String, List)
        Write to static field library.JSONParserFriend.is from instance method library.JSONParserFriend.makeHttpRequest(String, String, List)
```

Second Run:



```
com.example.dashboardactivity.DashboardActivity (2)
    Write to static field from instance method (2)
        Write to static field library.JSONParser.is from instance method library.JSONParser.getJSONFromUrl(String, List)
        Write to static field library.JSONParserFriend.is from instance method library.JSONParserFriend.makeHttpRequest(String, String, List)
```

Chapter

9

Bonus components

In all, we keep all of our code very clean, easy to read and well commented. Also we have a clean, fast and slick user interface from ground up. We think those are the two major points we are arguing for the bonus point.