Luca de Alfaro
@ UCSC

Search this site

Welcome!
Archived Projects
Classes
CMPS 121 - Mobile Applications
CMPS 121 Mobile Applications Winter 2016
Final Projects
Homework 1
Homework 2
Homework 3
Homework 4
Instructions for grading homework
Instructions for submitting homework
Lecture 1
Lecture 10
Lecture 11
Lecture 13
Lecture 14
Lecture 15
Lecture 17
Lecture 2
Lecture 3
Lecture 4
Lecture 5
Lecture 6
Lecture 7
Lecture 8
Lecture 9
CMPS 121 Spring 2012 - Mobile Applications
CMPS 121 Spring 2014 -

Classes > CMPS 121 - Mobile Applications > CMPS 121 Mobile Applications Winter 2016 >

Homework 3

Due: Friday February 19, 11pm
Upload Instructions: Upload your solution **here**.

In this assignment, you have to build an app that enables you to group chat anonymously with people nearby.

The API

You are going to use the following server API.
Base URL: `https://luca-teaching.appspot.com/localmessages/default/`

post_message

Parameters:

- lat : latitude (float)
- lng : longitude (float)
- user_id : your user id (string)
- nickname : nickname to talk to other users (string)
- message : the message you are posting. (string)
- message_id : a random string id you associate with each message. (string)

Example:

```
https://luca-teaching.appspot.com/localmessages/default/post_message?lat=9.9993&lng=10.0004&user_id=31&nickname=Hobbes&message=Tuna&me
```

```
{"result": "ok"}
```

get_messages

Parameters:

- lat : latitude (float)
- lng : longitude (float)
- user_id : user id (string)

This returns a certain number of local messages, in arbitrary order.

Example:

```
https://luca-teaching.appspot.com/localmessages/default/get_messages?
```

Mobile Applications

CMPS 121 Spring 2015 Mobile Applications

CMPS 121 Winter 2013 - Mobile Applications

CMPS 183: Web Applications

CMPS 276 - Software Engineering

CMPS 290G

How to host a simple app on Google Appengine

Raspberry PI Boot Camp

Collaborators

Contact Information

Creating and using a git repository

Interested in UCSC?

Is de Alfaro under A or D?

Past Projects

Publications

Publications (by topic)

Resources

How to create a simple app on appengine

SlugIOT

Talks

TestDocs

The Wikipedia Authorship Project

External Links

Resume

Luca's personal home page

CrowdGrader

UCSC School of Engineering

UC Santa Cruz

G+1 0

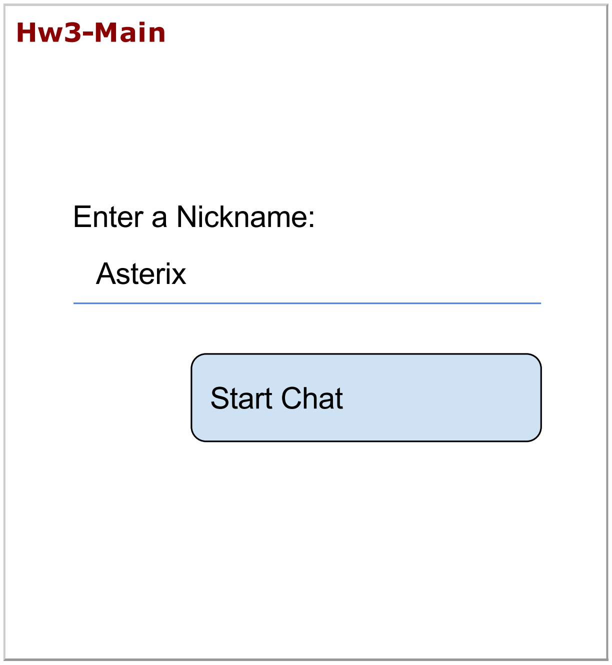
```
lat=9.99999&lng=10.0001&user_id=39
```

```
{"result_list": [{"timestamp": "2016-02-09T00:42:27.823020", "message": "Tuna", "nickname": "Hobbes", "user_id": "31", message_id=8755}, {"timestamp": "2016-02-09T00:41:53.328900", "message": "Holaaaa", "nickname": "PeterPan", "user_id": "38"}, {"timestamp": "2016-02-09T00:34:24.536050", "message": "Holaaaa", "nickname": "PeterPan", "user_id": "38", message_id=8755}], "result": "ok"}
```

Your app

You have to develop an app that behaves as follows.
The app consists of two activities, MainActivity and ChatActivity.

MainActivity



Hw3-Main

Enter a Nickname:

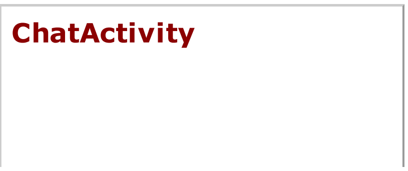
Asterix

In MainActivity, the first time, you create for yourself a random user_id and you store it in the Preferences.
Then, you ask the user to enter a nickname, and you acquire the location.

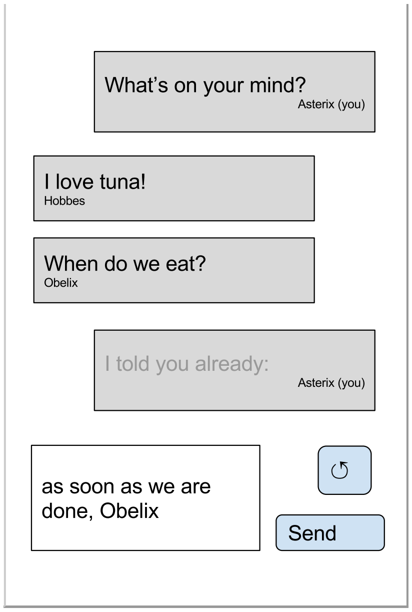
Once *both* the user has entered a nickname, *and* the location is known to 50m or better, you enable the Start Chat button. When pressed, the Start Chat button stores the nickname in the app preferences, and leads to ChatActivity.

ChatActivity

Look



ChatActivity



What's on your mind?
Asterix (you)

I love tuna!
Hobbes

When do we eat?
Obelix

I told you already:
Asterix (you)

as soon as we are done, Obelix

In the chat activity, at the bottom there is a place where you can post a message, and a button that you can use to refresh the list of messages.
Above, there are the various messages. You need to display the messages in chronological order (newest at the bottom), and you need to have some way of distinguishing the messages that are from you, and the messages that are from others. For this purpose, you can compare the user_id of each message with your own user_id (which you can find in the preferences).

Behavior

When the user clicks "refresh", you need to refresh the list of messages (using API method get_messages).

When the user types a new message and clicks *Send*, you have to:

- Add it to the bottom of the list of messages. ADVANCED: Use some way of denoting that it's not confirmed yet that the message has been sent (e.g., add a "sending" icon, or set the text to gray rather than black).
- Send it using the API call post_message.

ADVANCED: When the post_message call returns {"result": "ok"} you can remove the icon or text treatment you used to indicate that the sending is still pending, and you can display the sent message like all other sent messages.

Suggestions

Start early! Besides this, you can find sample code for almost all you need to do.

- Shobhit's findrestaurants app contains an example of how to get a location, how to enable a button once the location is known, and how to pass parameters to an HTTP call.
- Getting the location is also described here.
- Hw2 told you how to decode HTTP call results.
- How to switch activities.
- Listview is discussed here and here.

localchat.zip (1686k)	Luca de Alfaro, Mar 5, 2	v.1	⬇

Comments

You do not have permission to add comments.