Term Project for Advanced Software Paradigms (CSCI-210-10)
Department of Computer Science
The George Washington University

TODAY

A mobile tool of moods management

Version: 1.0

Nov. 25, 2010

Prepared for: Prof. Abdelghani Bellaachia

Prepared by: Maoxu Li & Zhong Xie

Description

Developing an Android application named Today to capture everyday feelings and share with friends.

Team Members

Maoxu Li	**Zhong Xie**
lim@gwmail.gwu.edu	xz1989@gwmail.gwu.edu

Table of Contents

Today – A mobile tool of moods management

Maoxu Li, Zhong Xie

Department of Computer Science, The George Washington University

lim@gwmail.gwu.edu, xz1989@gwmail.gwu.edu

I. Introduction and Motivation

Social networking applications like Facebook and Twitter let us connecting with our friends, and we now know what they are saying and what they are doing. Mobile networking extends social applications into our everyday life. With applications running on smart phones, we may do everything, everywhere, at anytime, which used to be done only front of desktop computer. Two of the most-attended mobile platforms are iPhone and Android.

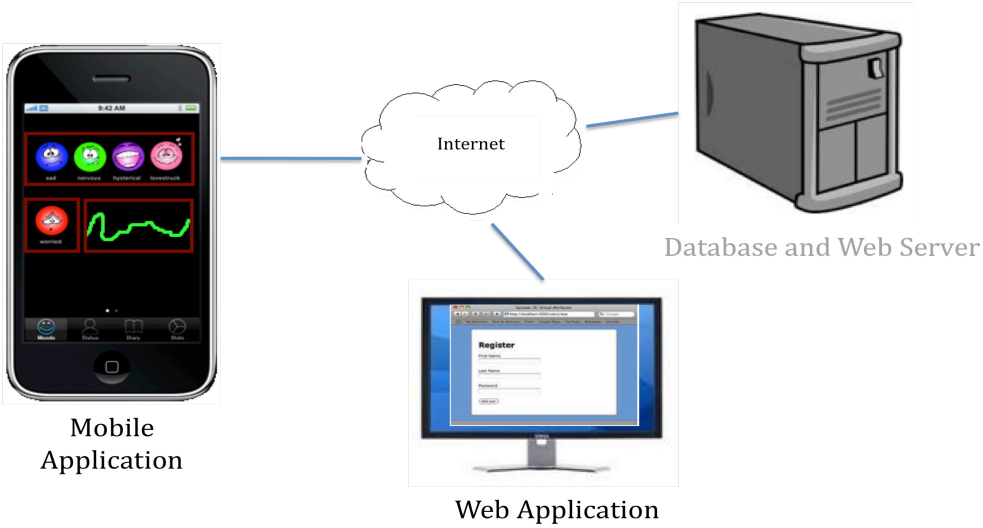
How are you today? How are you feeling every day? Do you frequently ask yourself and your friends?

This project will develop a mobile application on Android platform, named Today, to let you capture and manage your everyday feelings everywhere and anytime. The system will also let you share moods with friends, analyze moods changing during past days and find new friends with similar moods.

II. Problem and solution

The system will be designed as a daily used user-friendly tool. To let life easy, the user interface of the application must be very simple and easy to use. It will display a list of icons corresponding to various feelings. You just click one of the icons to capture your current feelings everywhere and anytime. The feelings data are permanently stored. Then the application may provide some visual analysis on the feelings data. For example, using a curve to denote how your moods change during the past days so that you can track on your moods and remind yourself to adjust moods time by time. Another example is matching moods to let you find who has similar moods like yourself.

The most convenient platform is obviously mobile phone. In general, the system can be realized as a Client/Server or Browser/Server system. On server side, central database is deployed to store the users' data. On client side, a mobile application with Graphic User Interface (GUI) lets users to capture and share their feelings. Obviously, some management functions, like user registration and users management, may be realized as a web application to facilitate deploy and usage. But local application will provide better user experience to realize the daily used functions like capture feelings and display moods data. The entire system architecture may be designed as below:



The mobile platform will be Android. The server platform will be Apache and mySQL. The mobile application running on Android platform has a graphic user interface. It interacts with the server via web service.

III. Requirements Analysis

1. Overview

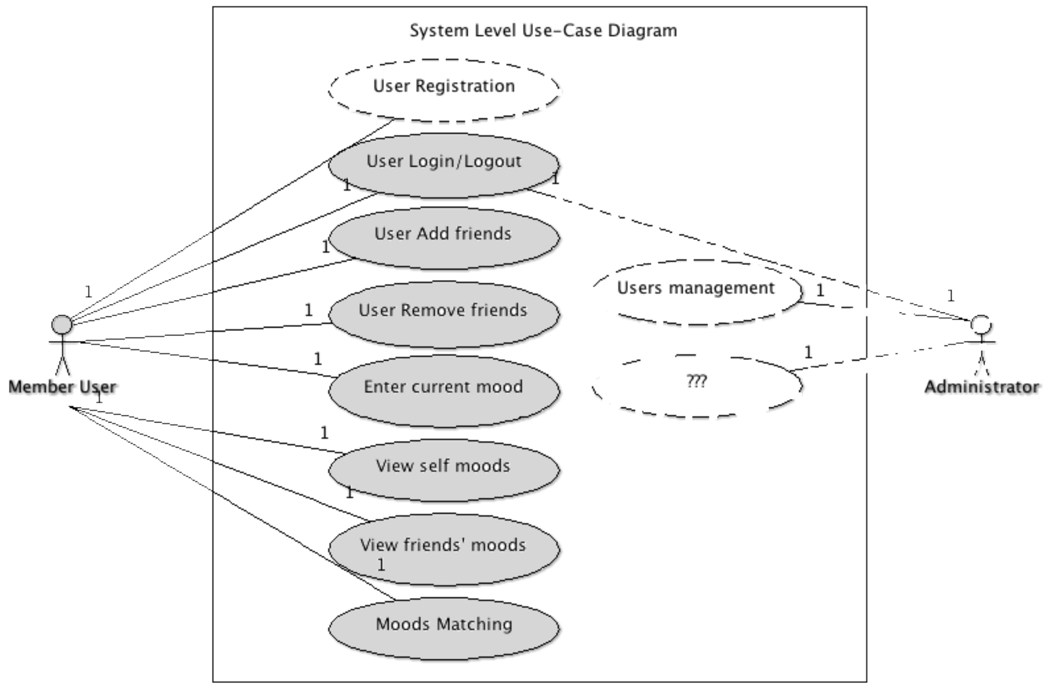
The core function of the system is moods management. The system uses concept of "friendship" to share feelings among users. A user can view his friends' feelings data. So friendship management is also an aspect of functions. The functions of the system include:

- Registration, log in and log out the system to control access.
- Management of friendship.
- Capture user's feelings data and stored in database.
- Analysis of users' feeling data: long-term moods changing and moods matching.

2. Use-cases

This document use use-cases to describe the requirements of functions and related user interface.

As a multiple users system, a predefined user role of Administrator is introduced to conduct some management functions. User registration function, as well as users management functions, is only available on web application. The overall use-cases of the system are shown below. The rest part of this section will describe all of the use-cases respectively, including the details of function logic and related user interface.



a. User Login/Logout

The users must have registered as member users of the system. Users need supply their login credentials in order to gain access to the system.

Precondition: User is not logged in, but is a member of the system.

1. Application displays “login screen”.
2. User enters username/email and password to login in the system.
3. Application connects to server to authorize the user.
4. If successful, the user is authenticated with proper rights and the application displays corresponding “main screen”.
5. If failed, application displays an error message and returns to login screen.

Post Condition: User is logged in and application displays main screen.

User may logout after login to return to Login screen.

User Interface:

1. Login screen:
    a. Input: username/email, password
    b. Button: login
    c. Links: Sign up, forgot password

b. User adds/removes friends

The system's notation of "friends" is people that are able to share moods each other. In order to become a friend, a user must confirm you.

Precondition: User is logged in.

1. Application displays "friends screen" with the list of current friends.
2. User selects one of more friends from the list, then click "remove" to remove friends.
3. User click "add friend" to enter a screen, then input username or email address to display the user's information, click "request" to request add this user as friend.
4. System sends an email to the selected user.
5. When the user confirms the "friendship", system sets these users to be friends.

Post Condition: User has new friends associated with his account.

User Interface:

1. Friends screen:
    a. Friends list with multiple choices box.
    b. Button: Add friend, Remove friend

c. Capture user's feeling

Application allows user to choose current feeling in "Moods screen".

Precondition: User is logged in

1. Application always displays feelings icon list in "moods screen".
2. User clicks the proper icon to enter current feeling.
3. System saves the entered feeling into the database.

Post Condition: A new record of user's mood is stored in the database.

User Interface:

1. Moods screen
    a. Feelings icon list: a list of icons corresponding to some feelings, like excited, very happy, good, just so so, sad, desperate, and so on.

d. Displays user's moods

The application always displays current user and his friends' moods in "Moods screen".

Precondition: User is logged in

1. Application always displays users' moods curves in "Moods screen".
2. User may click the "mood-curve" to view details.
3. Application displays the details of user's mood in a new screen.

Post Condition: Users' "mood-curve" is displayed and details can be viewed.

User Interface:

1. Main Screen
    a. A list of moods curves of current user and friends. A "mood-curve" denotes the mood changing during the past days.

e. Moods Matching

The system can analyze users' moods and find users with similar moods. The user may make friends with these "matched" users.

Preconditioned: User is logged in.

1. Application displays "moods matching screen".
2. Application displays a list of matched users with check boxes.
3. User selects the interesting user to views details of matching.
4. User selects interesting users and adds as friends.
5. The process to add as friends is same as former.

Post Condition: User finds some friends with certain moods.

User Interface:

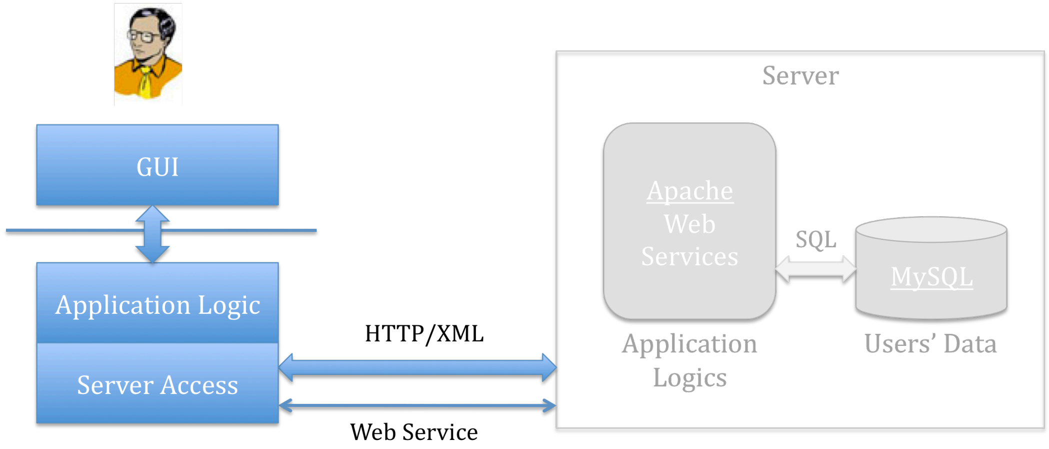
1. Moods matching screen
    a. List: matched users with check boxes
    b. Button: refresh, add as friends

IV. System Design

1. Layered Architecture

Firstly, the system is Client/Server architecture. On server side, central database is deployed to store users data, including users list and user's information, user's friends list, user's moods records, and some other application related data. On client side, a GUI application running on Android platform is developed to interact with the user and interact with the server to capture, store, analyze, and display user' and his friends' moods. The interface between server and client application is designed as web services, i.e., some web services is deployed on server side, and then the client application will access the server via the web services to access the data stored on the server as well as update the data.

The client application running on Android platform is also layered structure as below:



a. Graphic User Interface (GUI) layer

GUI is the presentation layer of the application. Typical applications with GUI are developed based on certain Graphic User Interface framework, such as MFC on Windows and Cocoa on Mac OS X, as well as Android SDK on Android platform.

This application will design the GUI based on Android SDK, which define a hierarchy of GUI objects. In the next section, the class hierarchy in GUI layer will be described in details.

b. Local Application Logic layer

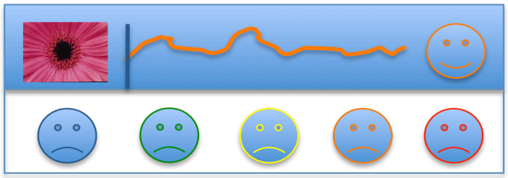
Local application logic layer manages a group of objects, which maintain the data needed by functions. Functional logic will be realized through the interaction among the objects. In next section, the logic classes related with all functions will be defined as well as the interactions among classes. Service access is also designed in logic layer.

2. Graphic User Interface (GUI)

GUI design is based on the previous functional analysis, i.e., the functional use-cases. In the context of GUI application development with Android SDK, the GUI consists of a hierarchy of GUI elements, including Views or Widgets. Here we describe the major GUI objects/classes design corresponding to all functions.

a. Simple mode

In simple mode, the application will display only a widget on the main screen of the device. The GUI elements in the widget is layout as shown:



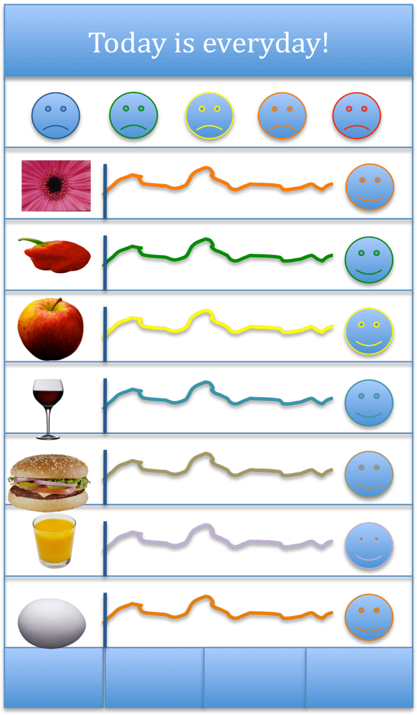
The first row is read-only region that displays user's logo, moods curve during the past days and the latest mood. Double click this region, the application will enter complete mode.

The second row is a list of icons that denote several of feelings. User may click one of the icons as his current feeling.

A class of Simple inherited from Activity is defined to manage the simple mode screen.

b. Complete mode

In complete mode, the application displays a view that is full of the screen. By default it displays moods view like below:



Details of Moods view:

- Applications title, double click to enter simple mode
- A list of feelings from that user may click one as current mood
- User's current mood and moods curve
- A list of friends and their icon, moods curve, and current mood
- Functions menu (buttons)

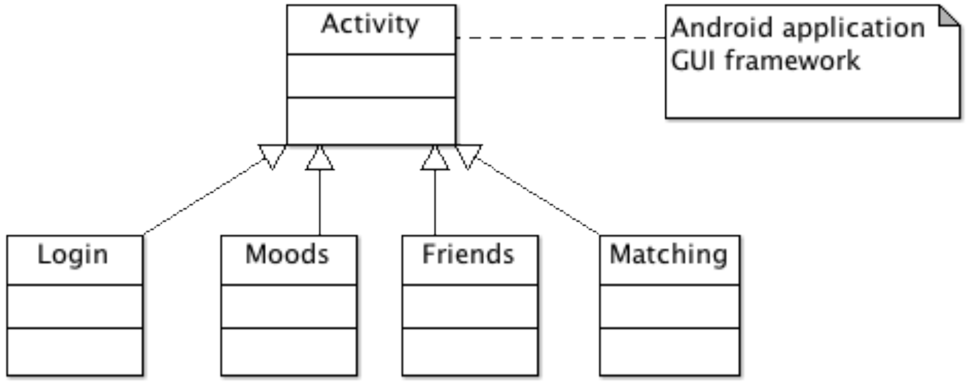
The menu buttons link to other main views. The 4 major view of the system are:

- Moods view
- Matching view
- Friends view
- Login view

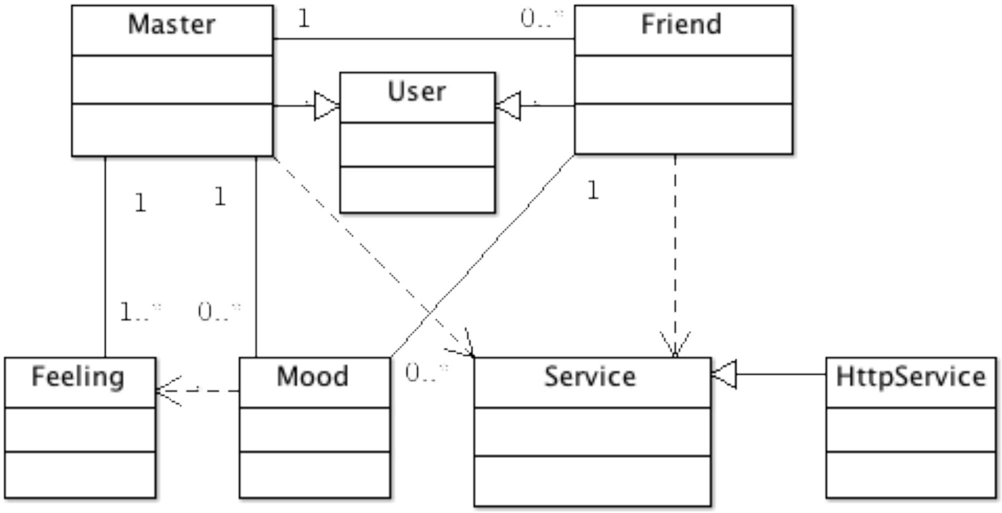
According to the framework of Android, a window and its actions is managed by an activity object. Four classes inherited from activity are used to manage Moods, Matching, Friends, and Login view. Class Model of Local Logic

3. Class model

GUI layer class model is shown below:



Logic layer class-model is shown below:



The Master class manage everything corresponding to current user. It is a singleton class. Service is a interface used to access web service. Strategy pattern is used here to adapt different service access protocols.

V. Implementation Issues

1. Database

MySQL system is deployed on server side to store data. Below is data table used:

- Users and Acl table

- Friends table
- Moods table

2. Web server

Apache web server is deployed to run web applications. Related techniques:

- SpeedPHP, a PHP MVC framework
- Management web application
- HTTP/XML based web service

3. Client platform

The client application is developed with Android SDK. An Android smart device emulator is used as debug and testing platform. Running on a real device is not considered in this project.

- Running platform: Android platform and Android Dalvik VM
- Development platform: Eclips + ADT
- Application framework: Android SDK
- Programming language: Java

4. Version control system

Git is used as version control system. A repository is created on Github to manage the documents and codes of this project.

https://github.com/limlabs/tie

VI. Testing

Unit testing is done while the functional modules are developed.

System testing is done based on the functional described above. All functions are tested according to the functional logic described in the use-cases. Please refer the details in system design section.

VII. Timeline and task distribution

Here is some key times that act as checkpoints to verify the development progress, as well as the task distribution in each stage.

	Tasks	Output	Date	Major	Minor
1	Problem identification and initial analysis	Abstract	Sep. 27	M. Li	Z. Xie
2	Requirements Analysis and initial design	Requirements	Oct. 11	M. Li	Z. Xie
3	Design and framework	Design	Oct. 25	M. Li	Z. Xie
	Logic layer implementation	Logic classes and package	Nov. 29	M. Li	
4	GUI Layer implementation	GUI classes and package	Nov. 29	Z. Xie	
5	Testing and Debug	Demo system	Nov. 29	Z. Xie	M. Li

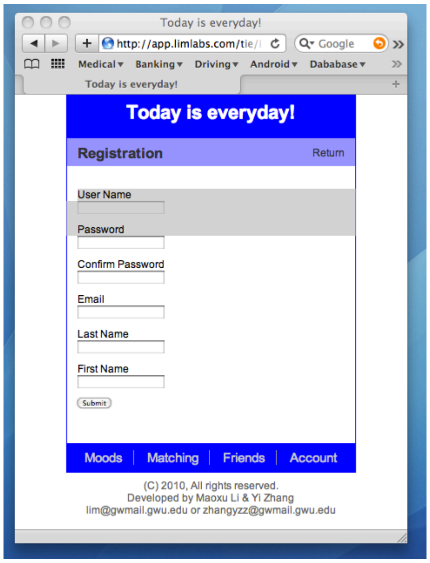
VIII. References

- http://www.android.com
- http://www.speedphp.com

IX. Appendix: User Manual

1. User registration

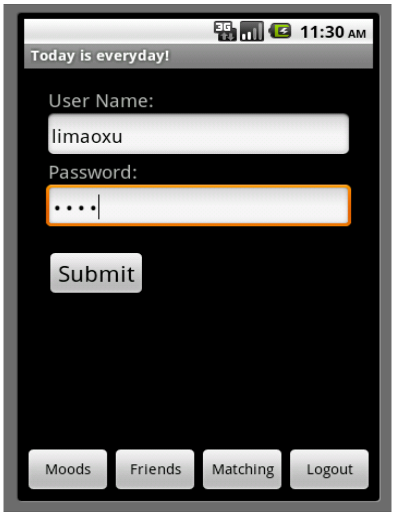
User must register first via web page to get a user name and password. Registration url is: http://app.limlabs.com/tie



(C) 2010, All rights reserved.
Developed by Maoxu Li & Yi Zhang
lim@gwmail.gwu.edu or zhangyzz@gwmail.gwu.edu

2. User Login

The login screen is as below. Input username and password then click submit. If log in successfully, the application will switch to moods screen.

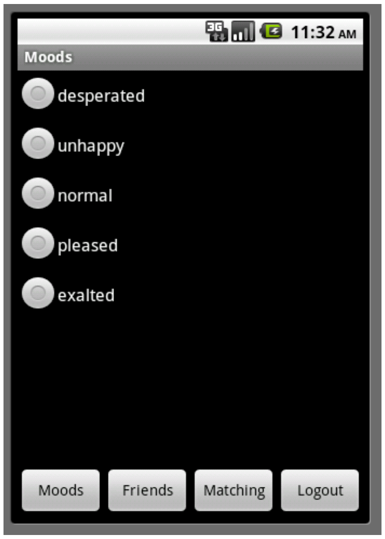


User Name: limaoxu

Password: ••••

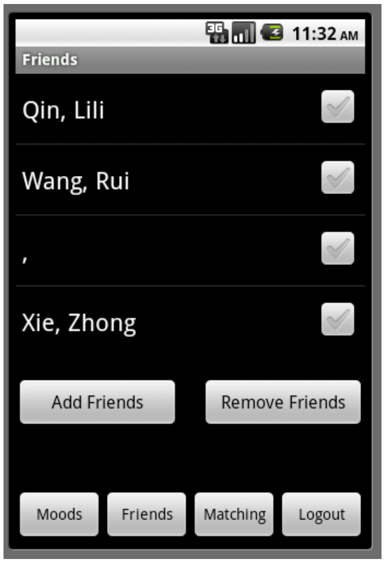
3. Moods capture

User only need to click one of the feelings icons corresponding to current feeling. then current mood will be transfer to server and stored in database.



4. Friends management

In friends management screen, user may choose some friends to remove or add a new friend with username or email address.


11:32 AM
Friends
Qin, Lili ☐
Wang, Rui ☐
, ☐
Xie, Zhong ☐
Add Friends
Remove Friends
Moods
Friends
Matching
Logout