Term Project for Advanced Software Paradigms (CSCI-210-10)

The Department of Computer Science

The George Washington University

TODAY

System Design

Version: 1.1

Oct. 23, 2010

Prepared for: Prof. Abdelghani Bellaachia

Prepared by: Maoxu Li & Zhong Xie

Description

Developing an Android application named Today to facilitate recording and tracking on moods, sharing moods with friends and making new friends via moods matching.

Team Members

Maoxu Li

lim@gwmail.gwu.edu

Zhong Xie

xz1989@gwmail.gwu.edu

Table of Contents

Today – A mobile tool of moods management

Maoxu Li, Zhong Xie

The department of Computer Science, The George Washington University

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

System Design

I. Overview

This project will design and implement a local application running on Android platform with the graphic user interface and the interactions with the web services running on central server (The development of the server side, including database and web services, will be completed as anther class project).

This document outlines the major design of the system based on previous requirements analysis (Please refer to the document of requirements specification for details). The section II will describe the Client/Server architecture in system-level and the 3-layer structure of the client application. The section III gives the detailed class model design in the 3 layers of client application, in which several design patterns are introduced and Object-Oriented design paradigm is used. Section IV and section V is some issues about implementation and development plan.

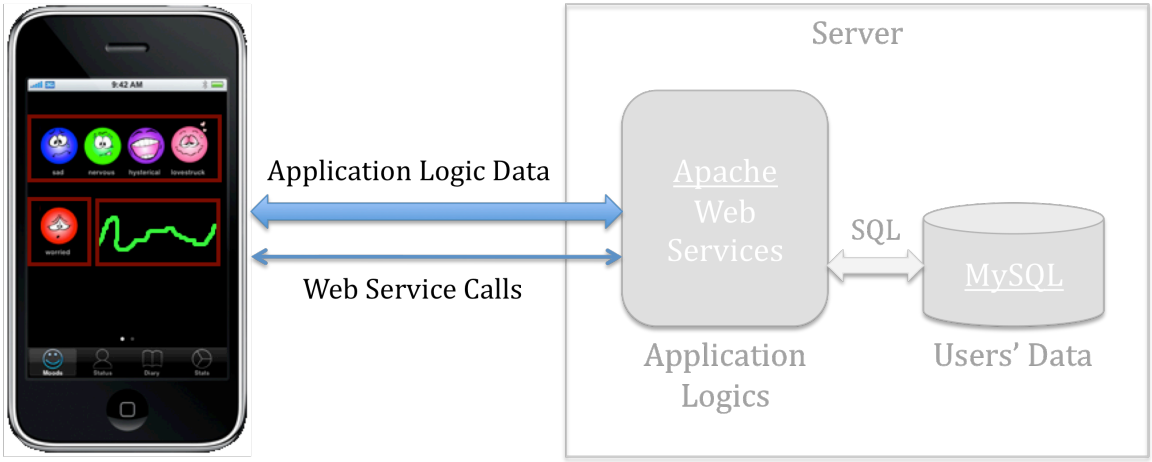
II. System Architecture

1. Client/Server architecture

As a typical networking application, this system is designed with 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 a web service, i.e., a web service

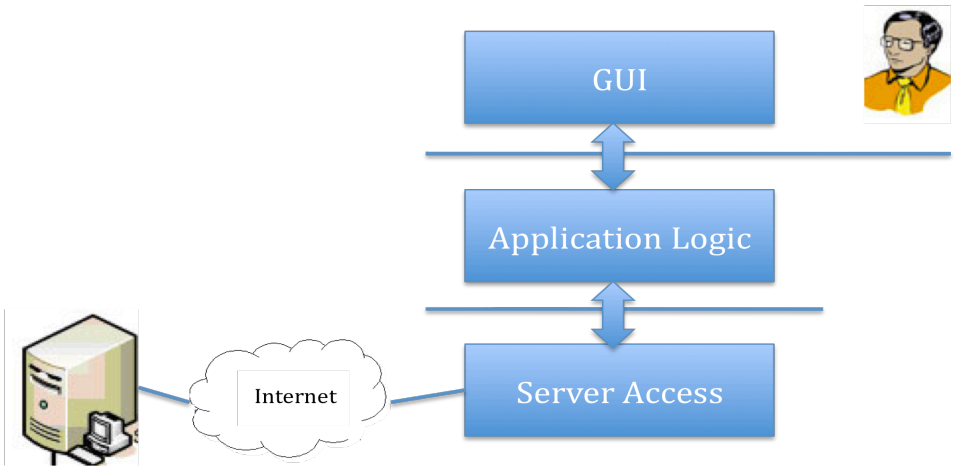
is deployed on server side, and then the client application will access the server via the web service to access the data stored on the server as well as update the data.

This project will complete the client side, i.e., a GUI application running on Android platform and the interactions with the web service. The development of server side, including database and web service, is not part of this project (which will be completed as another class project). The system-lelvel architecture is shown below.



2. Layered structure of client application

The client application running on Android platform is a typical GUI application with the capability of interaction with network server. Three-layer structure is usually used in this type of application, which is shown as below.



a. Graphic User Interface (GUI)

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

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.

c. Server Access

Server access layer is designed to be separate from local functional logic to decouple the implementation of local logic and the interface with server access. With this structure, it will be easy to adapt to different types of server access interface or modifications of the services. Detailed class design of server access layer is in the next section.

III. Class Model and Client Application

1. 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:

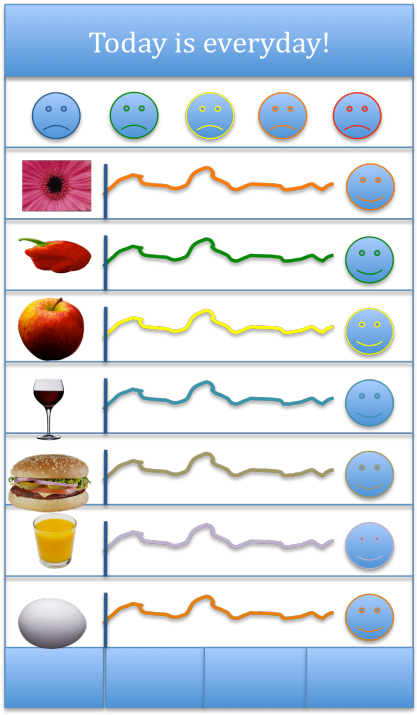
The first row is read-only region that displays user's logo, moods curve during the last days and the latest mood. Double click this region, the application will enter complete mode.

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

A class of SimpleView inherited from View 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 looks like below:



The complete mode screen consists of below regions:

- 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 region displays the list of friends is a variable functional region. It may displays functional screen corresponding to user's action.

- Click on current user's row or one of the friends' rows, variable functional region will display this user's detail information.
- Click on "remove friends" menu, variable functional region will display friends list with check box.
- Click on "add friends" menu, variable functional region will display user search screen and the following search result screen.
- Click on "moods matching" menu, variable functional region will display matching users list.
- Click on "setup" menu, variable functional region will display system setup screen.

A class of CompleteView inherited from view is defined to manage the complete mode screen.

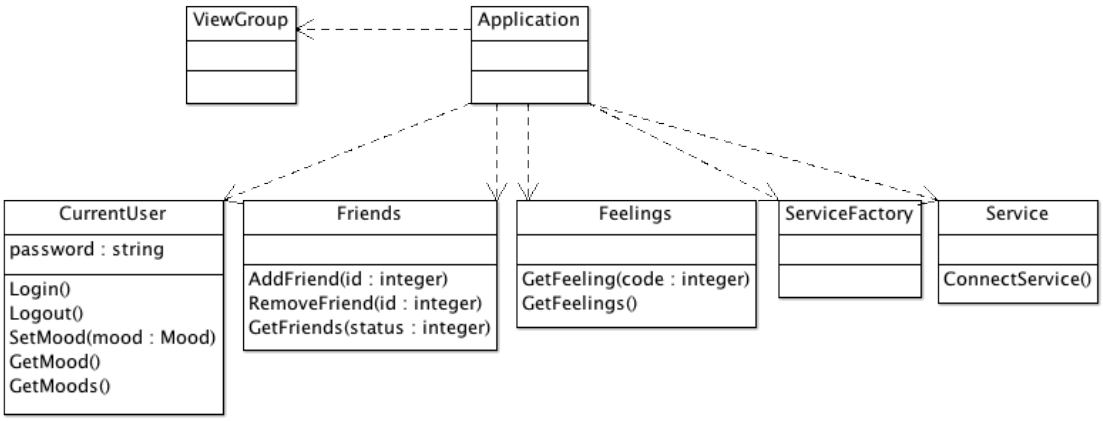
c. Hierarchy of GUI

According to the framework of Android, all of the GUI elements will be assigned within a hierarchy of ViewGroups and Views. The detailed GUI layout and the related GUI elements, as well as the action logics, will not be described entirely here. All of those designs will be refined with the progress of implementation.

2. Class Model of Local Logic

a. System class model

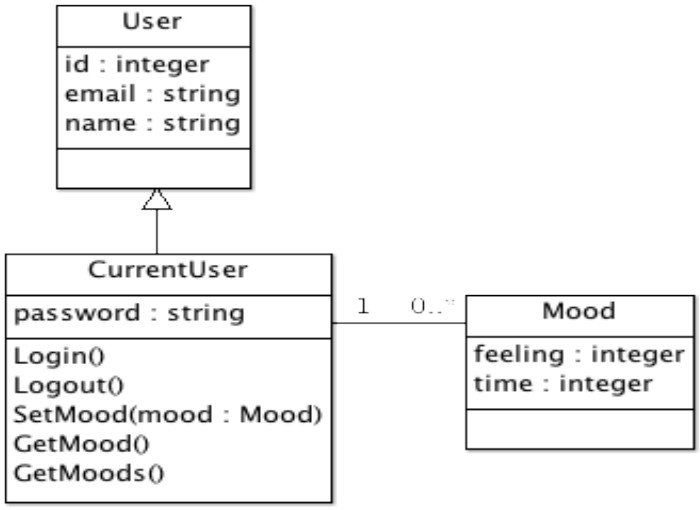
Application uses some classes to manage GUI, current user, user's friends, feelings for choosing, creation service access object via factory, and so on. Below is system-level class model.



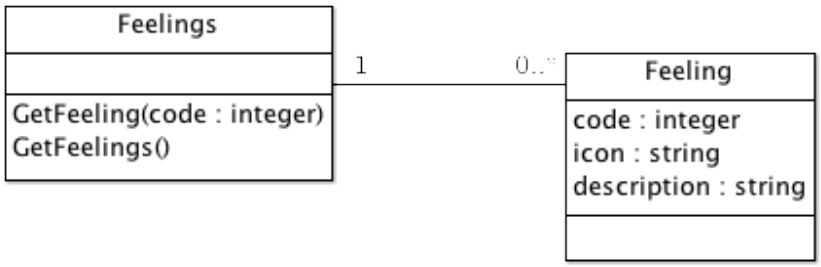
b. Current user and his mood

The major function of the system is mood management. In the context of this application, mood is defined as a temporal feeling of a user at a certain time while moods is defined a mood sequence within a range of time.

CurrentUser is an inherited class of User, it manages user's information, login and logout server, maintains a list of user's moods, and update the latest mood.

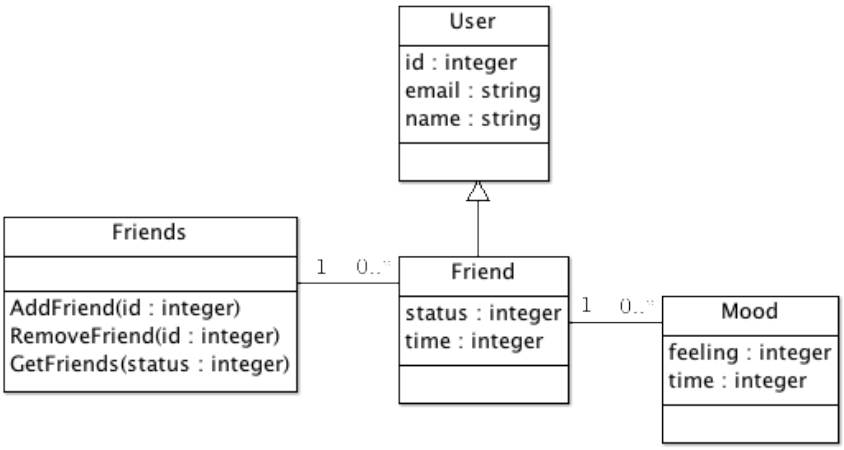


In addition, the user updates his current mood by clicking a feeling icon in a list. The feeling icons list is globally managed.



c. Friends and their moods

Friends manage a list of Friend of current user. As CurrentUser class, Friend is a descendent class of User.

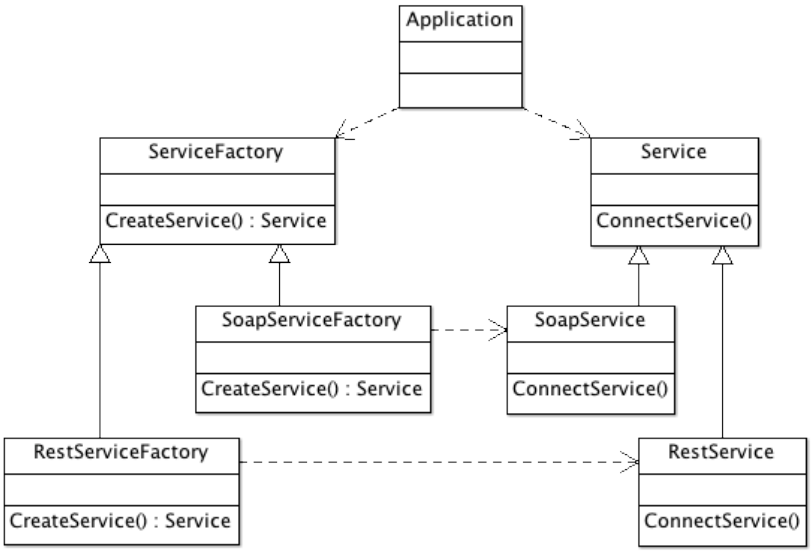


d. Users searching and moods matching

In the actions of add friends or moods matching, the system will return a list of users that meet the conditions. Obviously, these results can be managed with the descendent class of User. These objects will be temporary only exist during the process or the actions. The classes will be designed later with the progress of implementation.

3. Connect to server and access services

A design pattern of factory method is used to manage the creation of objects used to access services. Two standards of web services are supported, which are SOAP and REST. Service is the service access interface defined by local application.



IV. Implementation Issues

1. Platform

The application will be 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

2. Configuration Management

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

V. Development Plan

1. Timeline

Here is not a complete software development plan but some key times that act as checkpoints to verify the development progress and to adjust the development plan.

	Tasks	Output	Due	Memo
1	Problem identification and initial analysis	Abstract	Sep. 27	
2	Requirements Analysis and initial design	Requirements	Oct. 11	
3	Design and framework coding	Designs	Oct. 25	
4	Coding	Prototype	Nov. 8	Demo
5	Coding And integration test	System and documents	Nov. 29	

2. Tasks distribution

	Tasks	Issues	Distribution
1	Abstracts	Document	M. Li Z. Xie
2	Requirements	Document	M. Li Z. Xie
3	System design	Document	M. Li Z. Xie
4	Application Framework	3-layer framework classes and interfaces	M. Li Z. Xie
5	Logic function modules (GUI, logic classes, service access)	Current user functions	M. Li
		Add and remove friends	M. Li
		Friends moods	Z. Xie
		Moods matching	Z. Xie
6	Integrate testing	Plan and results	Z. Xie
7	Final project report	Documents	M. Li Z. Xie