David Dylegowski

CS 188/297
Final Project Report – SocialDialer

Table of Contents

I. Introduction....3
II. Development Environment....3
III. Design....3
A. User Interface Design....3
B. Class Design....7
IV. Implementation Details....9
A. Twitter API....9
B. Problems Encountered and Workarounds....10
1. Camera Snapshots....10
2. Making Phone Calls in Emulator....10
3. GIF images....11
4. PIM Contacts in Emulator....11
5. Form Class....11
V. Conclusion....11

I. Introduction

Social dialer is a phone book cell phone midlet application integrated with Twitter. For each contact, the user can associate a Twitter account with the contact. In the contact list, the status of the Twitter account is displayed in a scrolling ticker and the contact's Twitter image is used in the contact list. The user can add, remove, and modify contacts from the list. The user can also send an SMS text message to the contact and start a phone call with the contact.

II. Development Environment

The development environment used to develop this application was Java Platform Micro Edition Software Development Kit 3.0. The emulator used was DefaultFxTouchPhone1 with support for MIDP 2.0, CLDC 1.1, JSR 75, JSR 172, and JSR 135. This emulator is a touchscreen device, but it can run on a keypad device as well, as long as it used MIDP 2.0, CLDC 1.1, and the JSRs listed above.

III. Design

A. User Interface Design

The user interface was designed to be intuitive for users of other contact list applications they may be already be familiar with. The main page of the application is a listing of all existing contacts. For each contact, the name, Twitter screen name, and mobile phone number is displayed. An image for the contact may also be displayed, if one exists from the contact's Twitter account or from the phone's camera. If the selected user has a Twitter status, the date of the status and the status text is displayed in a ticker. The currently selected contact's background is colored orange to easily distinguish it from the unselected contacts.

Illustration 1: Main Screen

The user is provided menu options on the main screen of the application. The user can add a new contact, edit an existing contact, call the contact, send the contact an SMS text message, or exit the application.

Illustration 2: Main Screen Menu

The edit contact form allows the user to edit the contact's name, twitter screen name, and mobile phone number. The user may also try to use the camera on the device to take a snapshot of the contact.

Illustration 3: Edit Contact Form

When a user calls a contact, feedback in the form of an alert is given when the call is unable to be made. The alert is displayed for 3 seconds before dismissing. The user can dismiss the alert by executing the "Done" action.

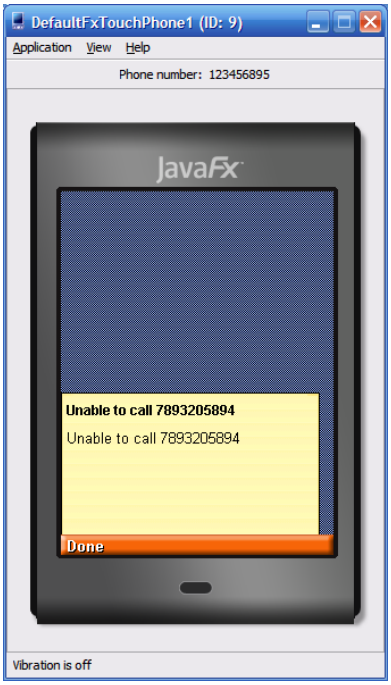

Illustration 4: Call Alert

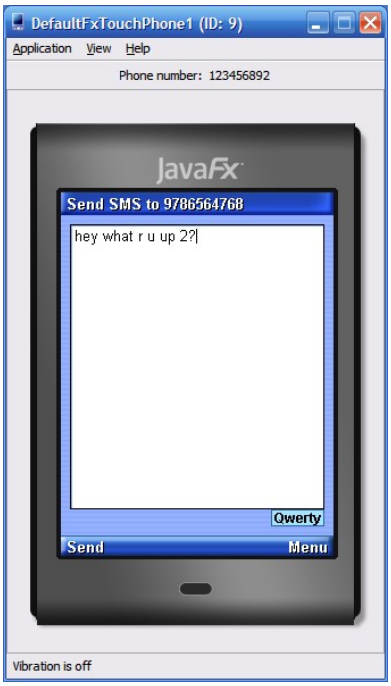

Illustration 5: Send SMS Form

When a user sends an SMS text message to a contact, feedback in the form of an alert is given when the message is sent successfully or if the message fails. The alerts are displayed for 3 seconds before dismissing. The user can dismiss the alert by executing the "Done" action.

Illustration 6: SMS Alert

B. Class Design

The class design for this application extends classes in the MIDP 2.0 API, most notably classes for the user interface such as Form and CustomItem. The SocialDialerContact class is a class that represents each contact. The contact list is implemented using the ContactListForm class, which is a subclass of Form. Each contact is displayed in the list using a subclass of CustomItem, ContactListFormItem. The class diagram for the application are displayed below.

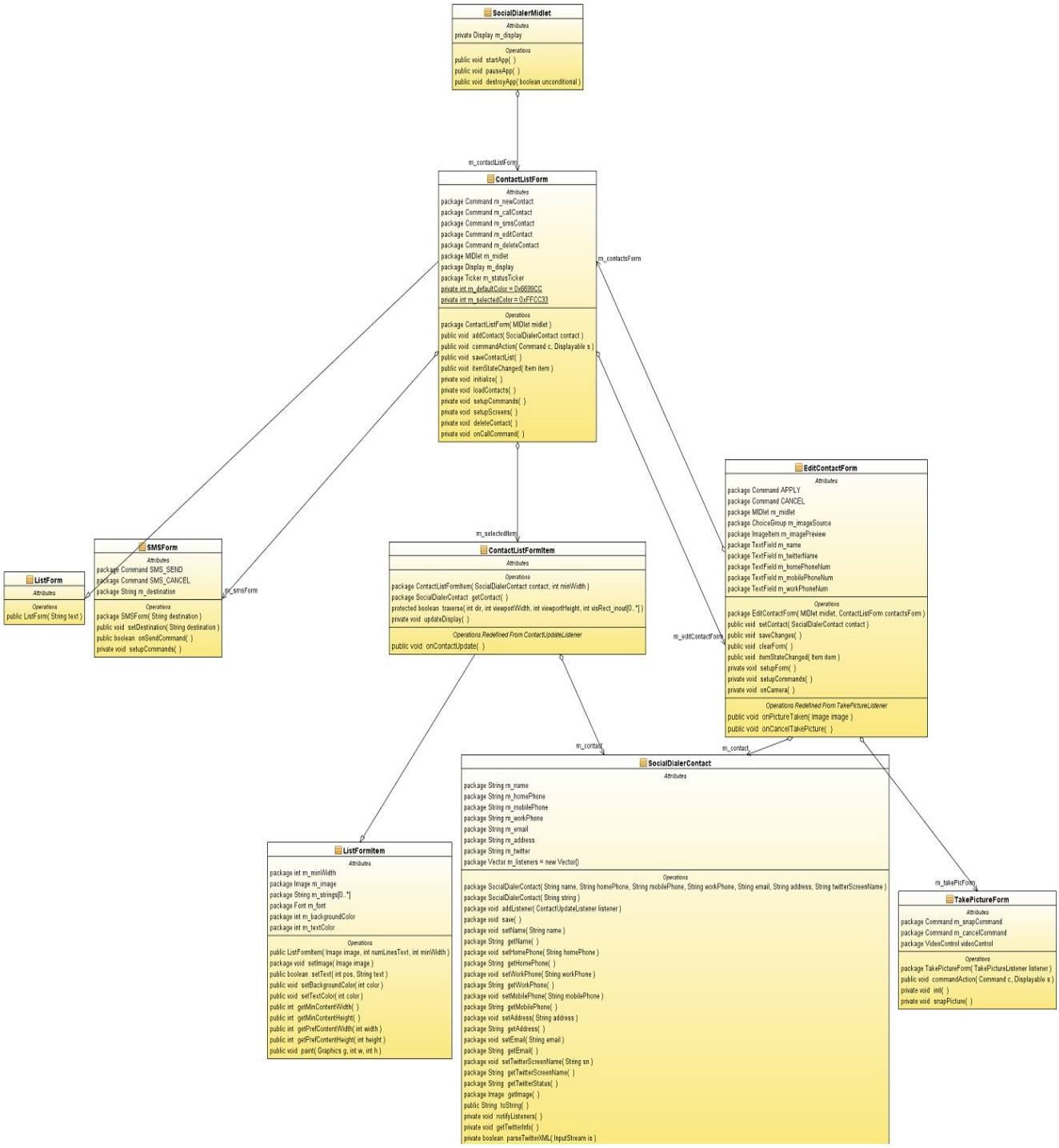

Illustration 7: Class Diagram

IV. Implementation Details

This application was written using J2ME in the Java Platform Micro Edition Software Development Kit 3.0 and utilizes MIDP 2.0 and CLDC 1.1. All contact information is stored in a text file, so JSR 75 was used for FileConnection. All Twitter information is provided via XML, which required JSR 172 for XML Parsing. In addition, MMAPI in JSR 135 was used to get access to the device's camera in order to take a snapshot to add to a contact.

A. Twitter API

Twitter provides a simple HTTP API that returns XML for a user's profile. The URL is in the form of http://twitter.com/users/show/USERNAME.xml. An example Twitter XML file is as follows:

<?xml version="1.0" encoding="UTF-8"?>

<user>

<id>783214</id>

<name>Twitter</name>

<screen_name>twitter</screen_name>

<location>San Francisco, CA</location>

<description>Always wondering what's happening. </description>

<profile_image_url>http://a1.twimg.com/profile_images/75075164/twitter_bird_profile_normal.png</profile_image_url>

<url>http://twitter.com</url>

<protected>false</protected>

<followers_count>2750786</followers_count>

<profile_background_color>ACDED6</profile_background_color>

<profile_text_color>333333</profile_text_color>

<profile_link_color>038543</profile_link_color>

<profile_sidebar_fill_color>F6F6F6</profile_sidebar_fill_color>

<profile_sidebar_border_color>EEEEEE</profile_sidebar_border_color>

<friends_count>134</friends_count>

<created_at>Tue Feb 20 14:35:54 +0000 2007</created_at>

<favourites_count>1</favourites_count>

<utc_offset>-28800</utc_offset>

<time_zone>Pacific Time (US & Canada)</time_zone>

<profile_background_image_url>http://s.twimg.com/a/1260393960/images/themes/theme18/bg.gif</profile_background_image_url>

<profile_background_tile>false</profile_background_tile>

<notifications></notifications>

```
  <geo_enabled>false</geo_enabled>
  <verified>true</verified>
  <following></following>
  <statuses_count>605</statuses_count>
  <status>
    <created_at>Thu Dec 10 22:19:53 +0000 2009</created_at>
    <id>6545585723</id>
    <text>Twitter in Italiano! http://blog.twitter.com/2009/12/ed-ecco-litaliano.html</text>
    <source>web</source>
    <truncated>false</truncated>
    <in_reply_to_status_id></in_reply_to_status_id>
    <in_reply_to_user_id></in_reply_to_user_id>
    <favorited>false</favorited>
    <in_reply_to_screen_name></in_reply_to_screen_name>
    <geo/>
  </status>
</user>
```

The tags used for this application are <profile_image_url> and <status>. These tags were parsed using the SAXParser in JSR 172. A custom DefaultHandler class, TwitterUserHandler, was implemented to capture values from the <profile_image_url> and <status> tags in the XML.

B. Problems Encountered and Workarounds

The application was not entirely developed as intended due to some bugs and limitations in Java Platform Micro Edition Software Development Kit 3.0. There are workarounds for some of the issues, but others went without a solution.

1. Camera Snapshots

One of the features of this application that was unable to be completed was allowing the user to take a picture from the camera and setting the picture in the contact list. Unfortunately, in the emulator, there is a timeout when taking a snapshot. The following is the exception thrown:

javax.microedition.media.MediaException: Timed out while making a Camera Snapshot

2. Making Phone Calls in Emulator

The midlet class provides the platformrequest method which allows you to make phone calls using a url of format “tel:XXXXXXXXXX”. When running in the emulator, however, it is not possible to make a phone call since there is no voice service connected to the emulator. This call will work on a device with voice service.

3. GIF images

In the emulator, GIF images are not implemented for the Image class. Some images from Twitter are given in GIF format and therefore are not displayed in the contact list items. No problems were experienced using images in JPG and PNG formats.

4. PIM Contacts in Emulator

The emulator is unable to read in PIM contacts. As a workaround, this application utilizes a file that contacts are read from and written to upon an add or edit. Unfortunately, if this application were to be deployed on a device, it would not integrate with the existing contacts on the device.

5. Form Class

One of the shortcomings of the Form class is that it does not provide an easy way to determine the currently selected item in the Form. In this application, this needed to be implemented to determine which ContactListFormItem was selected in the form. The ContactListFormItem is a subclass of CustomItem, which provides the traverse method, which is called by the system when traversal has entered the item. In ContactListFormItem the traverse method is overridden to call notifyStateChanged().

In order to handle the state changed events from ContactListFormItem, the ContactListForm, which displays all the ContactListFormItems, implements the ItemStateListener interface. Whenever a ContactListFormItem calls notifyStateChanged(), the ContactListForm's itemStateChanged(Item item) gets called with the item. Here is where the item is checked to see if it is a ContactListFormItem, and if it is, the ContactListForm updates its member variable for the currently selected item. Now the background of that item can be changed and menu actions for the selected item can be executed. This solution seemed too complicated for a fairly simple operation that is usually included in most user interface toolkits.

V. Conclusion

The SocialDialer application is a fully functional contact list integrated with Twitter. Some of the problems encountered unfortunately do not make the application quite as useful as initially intended. The biggest of the drawbacks is that this application does not integrate with PIM, forcing a user to manually add each of their contacts to the application and have to manage those contacts separately from the contacts available from the PIM. Another one of the drawbacks was that the emulator could not take snapshots from the camera, so a user cannot use the camera to set the image for a contact. These drawbacks are enough to prevent distribution of this application, but nonetheless, developing this application was a good J2ME learning experience.