ITIS/ITCS 4180/5180 Mobile Application Development

Midterm Bonus Exam

In-Class Programming Assignment

Basic Instructions:

---

1. This is the Midterm Bonus, which will count for 7% of the total course grade.
2. In every file submitted you MUST place the following comments:
    a. Midterm Bonus.
    b. File Name.
    c. Your Full Name.
3. This Midterm Bonus exam is an individual effort. Each student is responsible for her/his own Midterm Bonus and its submission.
4. Once you have picked up the exam, you may not discuss it in any way with anyone until the exam period is over.
5. During the exam, you are allowed to use the course videos, slides, and your code from previous homeworks and in class assignments. You are NOT allowed to use code provided by other students or from other sources.
6. Answer all of the parts of the exam.
7. Please download the support files provided with the Midterm Bonus and use them when implementing your project.
8. **Export your Android project as follows:**
    a) From eclipse, choose “Export…” from the File menu.
    b) From the Export window, choose General then File System. Click Next.
    c) Make sure that your Android project for this Midterm Bonus is selected. Make sure that all of its subfolders are also selected.
    d) Choose the location you want to save the exported project directory to. For example, your Desktop or Documents folder.
    e) The first time you export, make sure you select Create directory structure for files.
    f) Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
9. Submission details:
    a) When you submit the Midterm Bonus, compress your exported Android project into a single zip file. The format of compressed file name is:
        i. MidtermBonus.zip
    b) You should submit the Midterm Bonus through Moodle:
        i) Submit the zip file.
10. **Failure to do the above instructions will result in loss of points.**
11. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

Midterm Bonus (100 Points)

In this assignment you will develop "MessageMe." The App allows the user to send and receive messages from other users. The app uses parse.com to managed the user authentication and mailbox management.

Important App Requirements:

Points will be deducted if your application does not follow the below requirements:

1. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 17**. The app should display correctly on 3.2" QVGA (ADP2) (320x480: mdpi). Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.
2. For this assignment you will be provided with a project folder which includes the activity initial implementations and UI. Import the provided project and add your implementation.
3. **All parse.com communication should be performed using the background mechanisms provided by parse.com and should not block the main thread.**

Part A: User Signup and Login (20 Points)

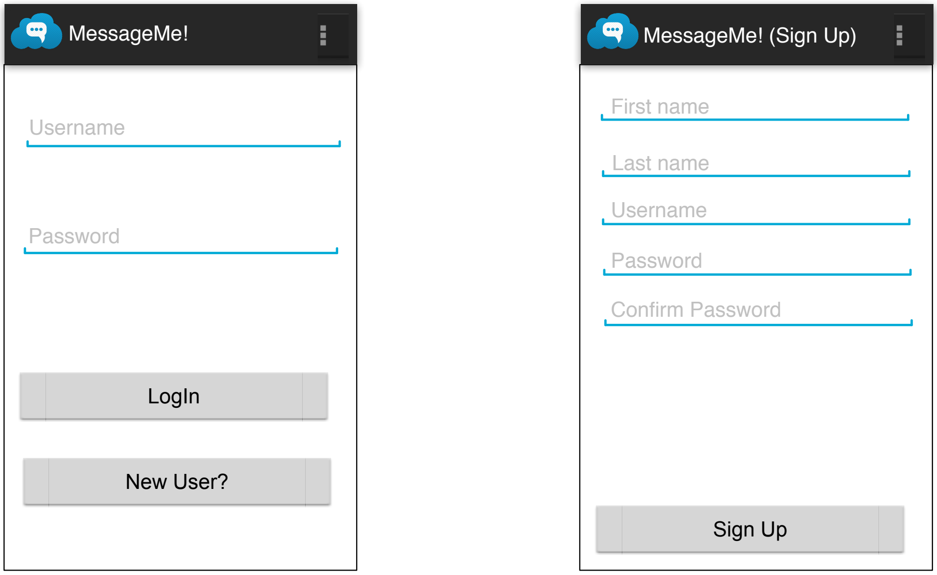
The app should implement both login and signup functionalities. You should use parse.com to store the user's full name, username and password in the User class. The requirements are as follows:

1. The launcher activity should be set to the Login activity. When the app first starts, the Login activity should check if there is a current user session, by using the parse provided methods to check if there is a valid current user:
   a) If there is a current valid user, then start the Inbox activity, and finish the Login activity.
   b) If there is no current valid user, then the Login activity should be used to provide user login.
2. Create a Login activity (Figure 1(a)):
   a) The user should provide their username and password. The provided credentials should be used to authenticate the user using parse.com. Clicking the "Login" button should submit the login information to parse.com to verify the user's credentials.
      - If the user is successfully logged in then start the Inbox activity, and finish the Login activity.
      - If the user is not successfully logged in, then show a toast message indicating that the login was not successful.

   b) Clicking the "New User ?" button should start the Signup activity and finish the login activity.
3. Create a Signup activity (Figure 1(b)):
   a) Clicking the back key should finish the Signup activity and start the Login activity.
   b) The user should provide their username, password, first and last name. The provided credentials should be stored in the User class in parse.com.

c) Clicking the “Sign Up” button should submit the user’s information to parse.com to verify the user’s credentials.

- The app should verify that there is data entered in all the fields, and that the password and confirm password are the same. Display a toast message if there is any error in data entry.
- If an account with the same username already exists, display an error message indicating that the account account was not created and the user should select a different username.
- If an account with the provided credentials does not already exist, then store the new account information and display a Toast indicating that the user has successfully login. Then start the Inbox activity and finish the Signup activity.
- To enable the TA to test your app, you should create all the accounts listed in Table 1.



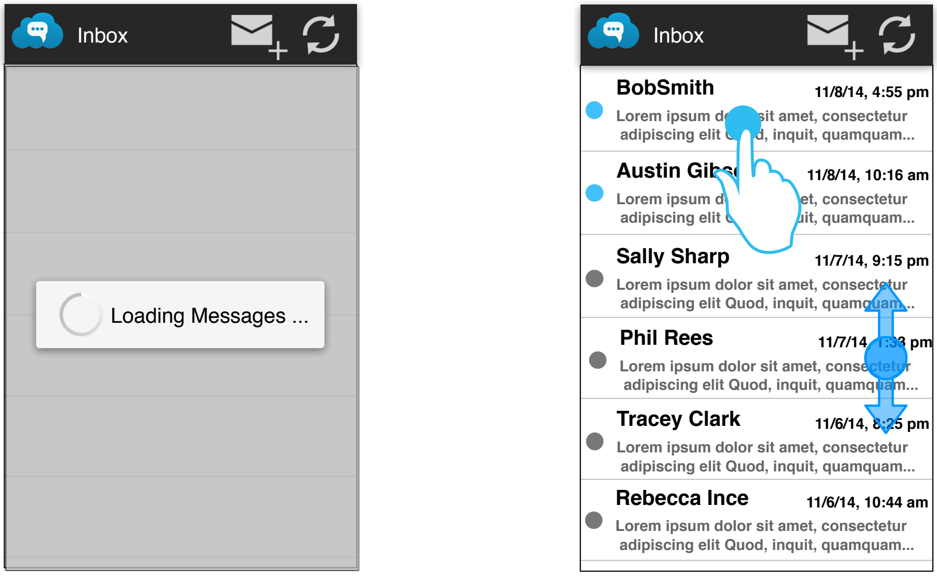
(a) Login Activity

(b) SignUp Activity

Figure 1, Wireframe for Login and SignUp Activities

UserName	First Name	Last Name	Password
agibson	Austin	Gibson	test1
ssharp	Sally	Sharp	test2
preese	Phill	Reese	test3
tclark	Tracy	Clark	test4
rinc	Rebbeca	Ince	test5

Table 1. Users accounts to be created.



Inbox

Loading Messages ...

Inbox

BobSmith 11/8/14, 4:55 pm
Lorem ipsum dolor sit amet, consectetur
adipiscing elit Quod, inquit, quamquam...

Austin Gibson 11/8/14, 10:16 am
Lorem ipsum dolor sit amet, consectetur
adipiscing elit Quod, inquit, quamquam...

Sally Sharp 11/7/14, 9:15 pm
Lorem ipsum dolor sit amet, consectetur
adipiscing elit Quod, inquit, quamquam...

Phil Rees 11/7/14, 1:38 pm
Lorem ipsum dolor sit amet, consectetur
adipiscing elit Quod, inquit, quamquam...

Tracey Clark 11/6/14, 8:25 pm
Lorem ipsum dolor sit amet, consectetur
adipiscing elit Quod, inquit, quamquam...

Rebecca Ince 11/6/14, 10:44 am
Lorem ipsum dolor sit amet, consectetur
adipiscing elit Quod, inquit, quamquam...

(a) Loading Messages

(b) Inbox after loading messages

Figure 2, Wireframe for Inbox Activity

Part B: Loading and Parsing Messages (Inbox Activity) (40 Points)

The Inbox activity should display the messages sent to the currently logged in user. The requirements are as follows:

1. In parse.com create a new Message parse class to store the message data, which includes the sender, receiver, message, and isRead (Boolean). Note that, parse.com automatically keeps track when a row is created (createdAt), which can be used as the sent date/time.
2. Retrieve all the messages sent to the currently logged in user. The Messages should be retrieved in the background using the Parse provided background query mechanism. While the messages are being retrieved the activity should display an alert progress dialog, which should be dismissed when the results are retrieved, See Figure 2(a). Check the documentation provided at https://parse.com/docs/android_guide#queries
   a) You can use the ParseQueryAdapter class. Check the documentation at https://parse.com/docs/android_guide#ui-queryadapter
3. The message row items should be displayed as shown in Figure 2(b). The items should be displayed in descending order by the message sent date (most recent at the top and older messages at the bottom).
4. The message row items should display the sender's name, time/date sent, and a short preview of the message, which is two lines with maximum 45 characters in length, so format the message to fit these constraints. A blue circle beside the message row item indicates the message has not been read, and a grey circle indicates the message has been read. Use the provided `circle_blue.png` and `circle_grey.png` files provided.
5. Clicking a message row item should start the ReadMsgActivity, which provides a more detailed view of the selected message item.
6. The ActionBar should contain two icons: a compose icon and a refresh icon.

6.1. Clicking the compose icon should start the ComposeMsgActivity, which allows the current user to compose a new message.

6.2. Clicking the refresh icon should refresh the message list. It should query parse.com and update the list based on the results retrieved from parse.com results. While the messages are being retrieved the activity should display an alert progress dialog, which should be dismissed when the results are retrieved, see Figure 2(a).

7. The Inbox Activity should be refreshed when ever the activity is resumed.

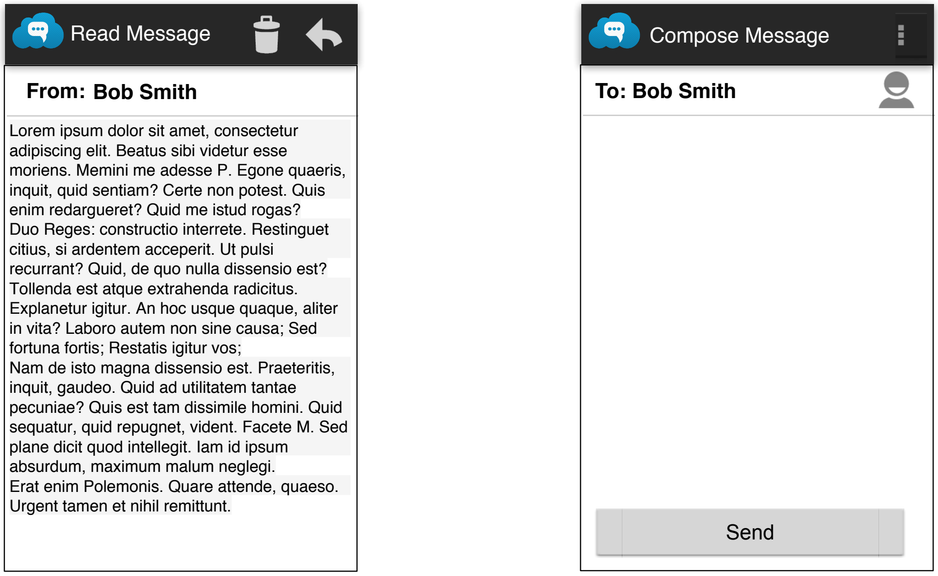
Part C: Read Message Activity (20 Points)

The ReadMsgActivity shows details of the selected message item. The user can view the message, reply to sender, or delete the message. The requirements are as follows:

1. The activity's wireframe is shown in Figure 3(a). The UI should display the message sender's name, and the complete message text.
2. The message status should be updated to read status and should the updated status should be stored on parse.com using the background update features provided by parse.com.
3. Pressing the back key should return to the Inbox Activity.
4. The ActionBar should contain two icons: a trash, and a reply arrow.

4.1. Clicking the trash icon should delete the message from parse.com and return to the Inbox Activity after the message is successfully deleted. This operation should be performed in the background. The Inbox Activity should be refreshed to reflect the up to date messages stored on parse.com.

4.2. Clicking the reply arrow should start the **ComposeMsgActivity.**



From: Bob Smith

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Beatus sibi videtur esse moriens. Memini me adesse P. Egone quaeris, inquit, quid sentiam? Certe non potest. Quis enim redargueret? Quid me istud rogas?
Duo Reges: constructio interrete. Restinguet citius, si ardentem acceperit. Ut pulsi recurrant? Quid, de quo nulla dissensio est?
Tollenda est atque extrahenda radicitus. Explanetur igitur. An hoc usque quaque, aliter in vita? Laboro autem non sine causa; Sed fortuna fortis; Restatis igitur vos;
Nam de isto magna dissensio est. Praeteritis, inquit, gaudeo. Quid ad utilitatem tantae pecuniae? Quis est tam dissimile homini. Quid sequatur, quid repugnet, vident. Facete M. Sed plane dicit quod intellegit. Iam id ipsum absurdum, maximum malum neglegi.
Erat enim Polemonis. Quare attende, quaeso. Urgent tamen et nihil remittunt.

To: Bob Smith

(a) Read Msg Activity

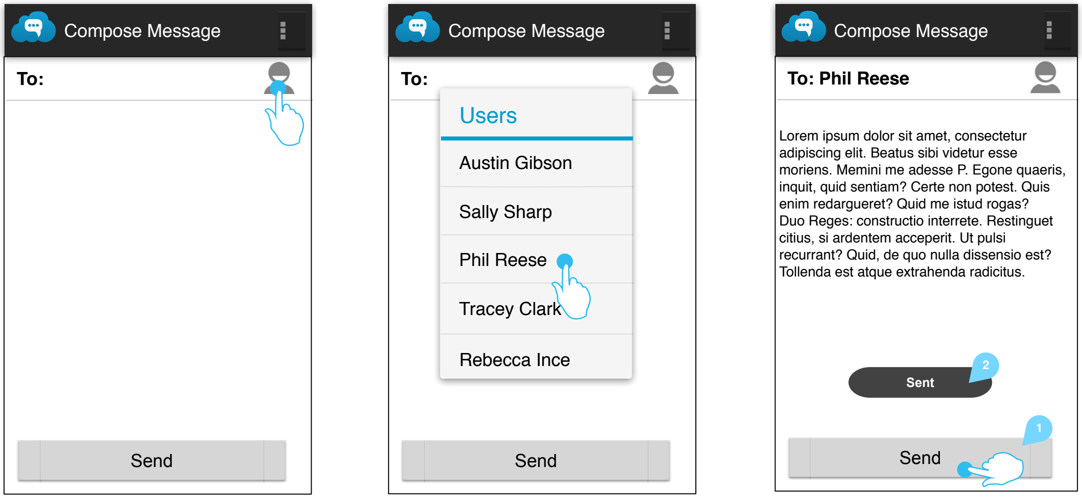
(b) Compose Msg Activity entered via clicking the reply button in the Read Msg Activity

Figure 3, Wireframe for Inbox Activity

Part D: Compose Message Activity (20 Points)

The ComposeMsgActivity allows the user to either compose a new message or reply to a message item selected from the Inbox. The requirements are as follows:

1. If the Compose Message Activity is started by Read Message Activity, then the “To:” TextView should be pre-initialized with the name of original message sender, to which the user would like to reply. The user should not be allowed to change the pre-initialized name. See Figure 3(b).
2. If the Compose Message Activity is started by the Inbox activity by selecting the composed icon, then the “To: ” row should be empty, see Figure 4(a).
   a) Clicking the “Contact” icon should display an alert dialog containing a list of all the full names of the users stored in the User class on parse.com. The list of users should be retrieved in the background by submitting a parse.com query. Selecting a name should initialize the “To: ” TextView field. See Figures 4(b)&(c).
3. After allowing the user to type in the desired message, pressing the “Send” button should upload the message to parse.com. After the message is successfully sent, a Toast message should be displayed indicating the message has been successfully sent, finish the ComposeMsgActivity and return to the Inbox Activity.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Beatus sibi videtur esse moriens. Memini me adesse P. Egone quaeris, inquit, quid sentiam? Certe non potest. Quis enim redargueret? Quid me istud rogas? Duo Reges: constructio interrete. Restinguet citius, si ardentem acceperit. Ut pulsi recurrant? Quid, de quo nulla dissensio est? Tollenda est atque extrahenda radicitus.

(a) Selecting the users icon (b) Users loaded from parse.com (c) Sending the message

Figure 4, Wireframe for Compose Msg Activity