

ITIS/ITCS 4180/5180 Mobile Application Development
Midterm Makeup

Basic Instructions:

1. This is the Midterm Makeup.
2. In every file submitted you **MUST** place the following comments:
 - a. Your Full Name.
3. This Midterm Makeup is an individual effort. Each student is responsible for her/his own Midterm Makeup 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 the exam parts, all the parts are required.
7. Please download the support files provided and use them when implementing your project.
8. **Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.**
9. **This is a makeup assignment and the tasks will be graded based on correctness and no partial credit will be granted for partial implementations.**
10. Export your Android project and create a zip file which includes all the project folder and any required libraries. The file name is very important and should follow the following format: **MidtermMakeup.zip**
11. **Failure to do the above instructions will result in loss of points.**
12. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**
13. **To enable the TA to test your app, you should create all the accounts listed in Table 1**

| username | first name | last name | password |
|----------|------------|-----------|----------|
| agib | Austin | Gibson | test1 |
| ssharp | Sally | Sharp | test2 |
| preese | Phill | Reese | test3 |
| rinc | Rebbeca | Ince | test4 |

Table 1, Users accounts to be created.

Midterm Makeup (100 Points)

In this assignment you will develop a Photo Gallery App. The App enables multi users to search, view, share photos with friends and mark favorites photos. This app will focus on adding parse.com integration with the midterm makeup preparation app to enable the required functions.

Figure 1 consists of two screenshots of a mobile application interface. Screenshot (a) is titled '500px Gallery (Login)' and shows a login form with two text input fields labeled 'Email' and 'Password'. Below these fields are two buttons: 'Login' and 'Create New Account'. Screenshot (b) is titled '500px Gallery (SignUp)' and shows a sign-up form with four text input fields: 'User First Last Name', 'Email', 'Password', and 'Confirm Password'. Below these fields are two buttons: 'Sign Up' and 'Cancel'.

(a) Login

(b) Sign Up

Figure 1, User SignUp and Login Activities

Part A: User Signup and Login

This Activity is from the Midterm Makeup Preparation. Your app should implement both login and signup functions. You should use parse.com to store the user first name, last name, email address and password in the Parse 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 Main 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 email and password.
 - Check the user input, if the email or password field is left empty, print a toast message indicating that these fields are required and don't submit the login information to parse.com.
 - b) 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 Main 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.
 - c) Clicking the “Create New Account” button should start the Signup activity and finish the login activity.
3. Create a Signup activity (Figure 1(b)):
- a) Clicking the “Cancel” button should finish the Signup activity and start the Login activity.
 - b) The user should provide their first name, last name, email and password. The provided credentials should be stored in the User class in parse.com. Clicking the “Sign Up” button should submit the user’s information to parse.com to verify the user’s credentials.
 - Check the user input, if any of the fields is left empty, or if the password and confirm password do not match, print a toast message indicating the corresponding error message and don't submit the provided information to parse.com.
 - If an account with the same email already exists, display an error message indicating that the account account was not created and the user should select a different email.
 - 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 Main activity and finish the Signup activity.
 - Note that, the username and email should set to the same value in the user’s table.

Part B: Main Activity

This Activity is from the Midterm Makeup Preparation. The Main activity displays a EditText that enables the user to enter a keyword search term. Figure 2 shows the Main activity. The requirements include:

14. The search EditText should be used to enter the search keyword, which should be used by the search api, See Figure 2.
15. Upon clicking the submit button start the Gallery Activity. The intent should include the search term using extras. You will not be given any points if you use a static variable to share data between activities.
16. If the submit button is clicked and the EditText is empty display a Toast Message indicating that the field was empty and do not start the next activity.



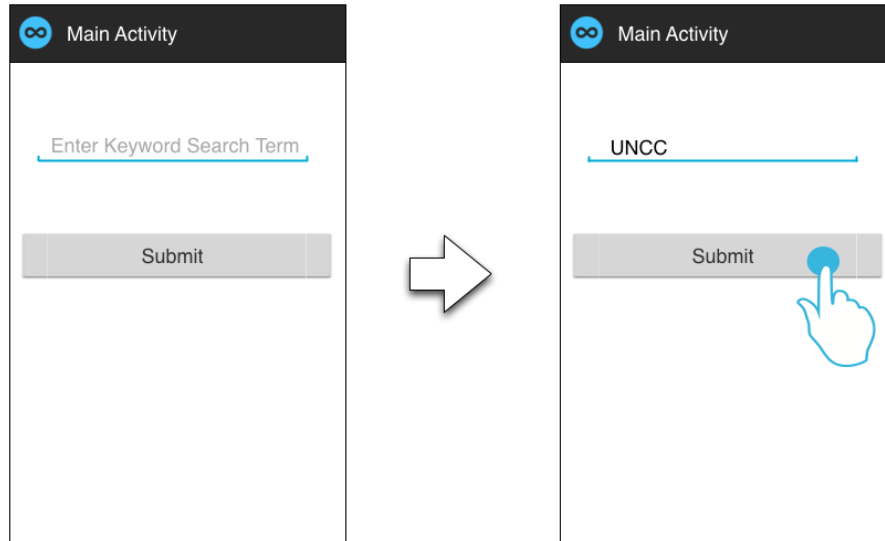


Figure 2. The Main Activity

Part C: Gallery Activity (80 pts)

Parts of this Activity are from the Midterm Makeup Preparation The Gallery activity is responsible for the retrieving the photo search results based on the search keyword term sent from the Main Activity. The Gallery Activity wireframe is shown in Figure 3. The requirements include:

1. The 500px photos/search api should be used to retrieve the photos matching the requested keyword term:
 - Use the api documentation provided in https://github.com/500px/api-documentation/blob/master/endpoints/photo/GET_photos_search.md
 - Set the “**consumer_key**” parameter based on your 500px consumer key.
 - Set the “**term**” parameter to the search term keyword.
 - Set the “**image_size**” parameter to 4.
 - Set the “**rpp**” parameter to 50.
 - Use the api endpoint <https://api.500px.com/v1/photos/search?>
2. Send a GET request and retrieve the photo search results for the selected keyword term. All the parsing and HTTP connections should be performed by a worker thread or an AsyncTask and should not be performed by the main thread. While, the JSON is being downloaded and parsed you should display a progress dialog, see Figure 4.
3. Create a photo class that stores the photo title, photo url, photo view count, owner’s name and owners photo. The JSON parser should return a list of photo objects.
4. **(NEW)** You should use parse.com to store (manage) the users’ favorite photos information. The current logged in user’s favorite photos information should be retrieved from parse.com.
5. **(NEW)** The progress dialog should be dismissed after the parsing is completed, the favorites retrieved from parse.com and the ListView or RecyclerView should be setup and displayed. We recommend you use a RecyclerView. Each list item should display the photo title, photo thumbnail and favorite filled or unfilled star, as shown in Figure 3(b). The photos should be sorted by photo title.

6. Clicking an item should start the Details Activity. The intent should include the required photo information to be sent to the Details activity using extras. You should not use a static variable to share data between activities.
7. **(NEW)** The favorites icon beside each list item should be displayed such that:
 - The filled star should indicate that the current photo is in the user's favorites.
 - The unfilled star should indicate that the current photo is not in the user's favorites.
 - This should be maintained for all the displayed lists.
8. **(NEW)** Clicking the favorites star should toggle the photo's favorites status, such as:
 - Clicking an unfilled star should store the photo item to the user's current favorite photos on parse.com, and the star should be changed to filled.
 - Clicking a filled star should delete the photo item from the user's current photos on parse.com, and the star should be changed to unfilled.

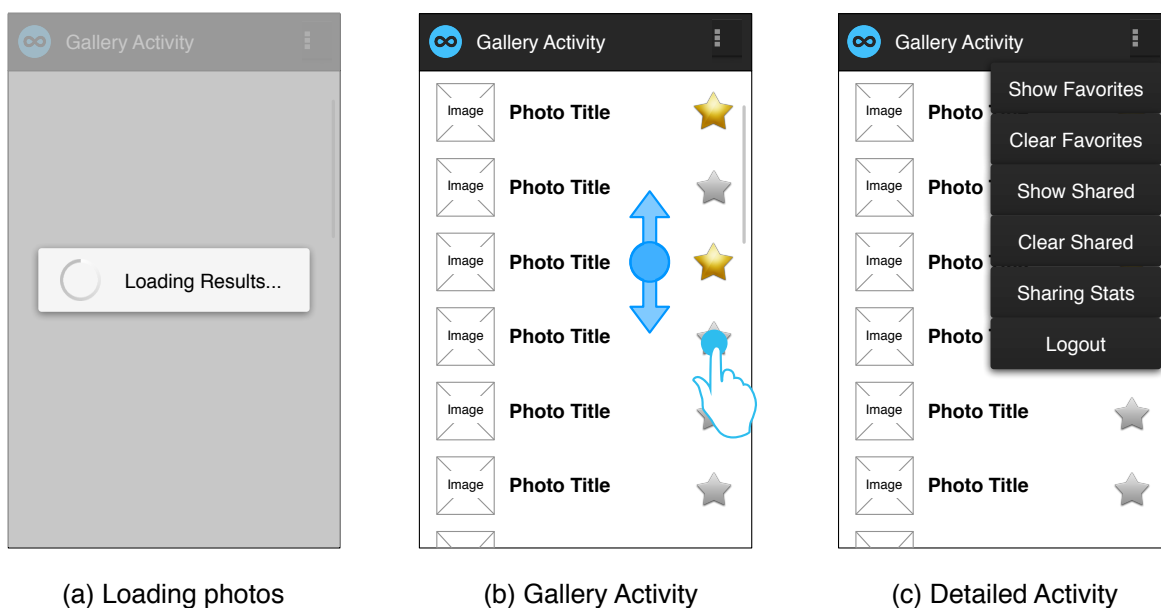
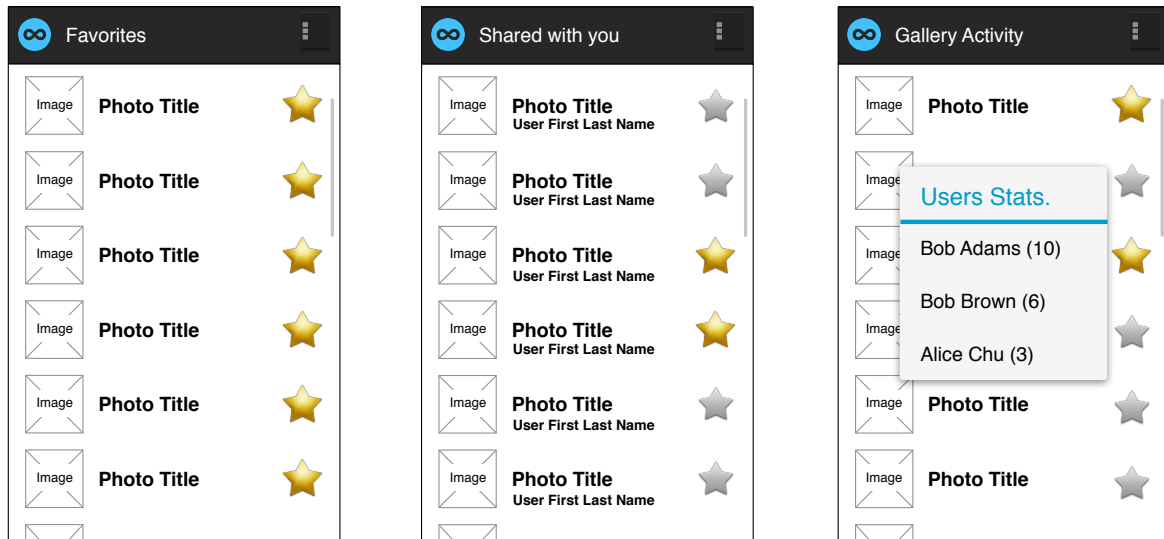


Figure 3, Gallery Activity

9. **(NEW)** Menu Items: The below menu items should be added to the Gallery Activity:
 - **Show Favorites:** Tapping this item should show the list of the photo items that were favorited by the current user. This information should be retrieved from parse.com. As shown in Figure 4(a). Change the action bar title to "Favorites".
 - **Clear Favorites:** Tapping this item should delete favorites photo items information maintained for the current user on parse.com and should reload the displayed list if needed. Show a toast message indicating that the favorites have been cleared.
 - **Show Shared:** Tapping this item should display the list of photos that were shared with the currently logged in user. In addition each list item should indicate the first and last name of the user that shared the photo item with the current user. As shown in Figure 4(b). Change the action bar title to "Shared with you".
 - **Clear Shared:** Tapping this item should delete all the shared photo items information that were shared with the currently logged in user on parse.com and should reload the displayed list if needed. Show a toast message indicating that the shared have been cleared.

- *Sharing Stats: Tapping this item should display an alert dialog containing the list of users who shared photos with the currently logged in user ordered in descending order based on the number of times they did share items with the currently logged in user. See Figure 4(c), Bob Adams has shared 10 photos with the currently logged in user.*
- *Logout: Tapping this item should logout the user, end the activity and show the login activity.*



(a) Showing Favorites List

(b) Showing Shared with you

(c) Showing Sharing Stats

Figure 4, Gallery Activity and Details Activity

Part D: DetailsActivity (20 Points)

This activity should display the photo details, which includes the photo title, owner name and the photo itself. The DetailsActivity should receive the photo information from the GalleryActivity. Image Download should be done in a child thread. Figure 5(a) shows the DetailsActivity. The implementation requirements include:

1. Use a thread pool (or AsyncTask) to retrieve the selected photo. If there is no photo image, display a default image of your choice to indicate no photo found.
2. Pressing back should end this activity and go back to the Gallery activity.
3. **(NEW)** Tapping the "Share" Button should do the following:
 - (a) Display an Alert Dialog listing all the users that are currently stored on your parse.com. The users should be sorted in ascending order by first name then last name.
 - (b) Selecting a user from the list should share the current photo item with the selected user, which should save all the required photo item information on parse.com in addition the current user and the user the photo is being shared with. See Figure 5(b).
 - (c) Display a toast message, see Figure 5(c).

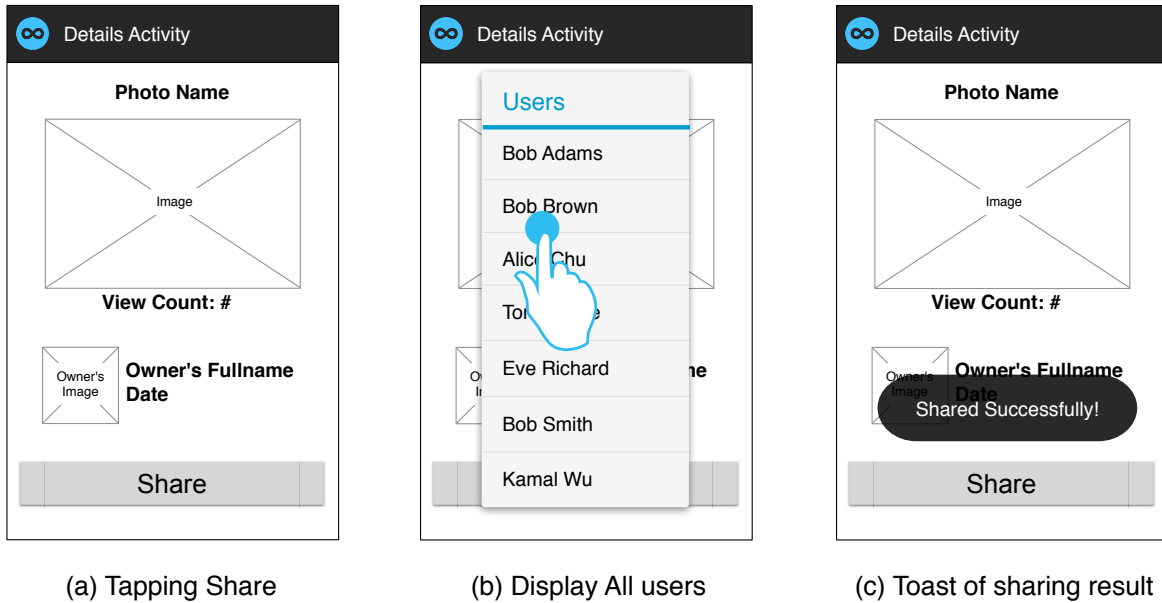


Figure 5, Share workflow

Note: Below is the grading rubric, there will be no partial credit granted for incorrect implementations. Please use your time wisely and select the functionalities that you are able to complete during the exam time.

| Percentage (%) | Rubric Item |
|----------------|--|
| 10% | Gallery Activity — The favorite indicator (filled or unfilled star) correctly displayed for the favorites and not favorites photo items respectively. This should work in all the lists displayed. |
| 10% | Gallery Activity — Add to favorites: Clicking on favorite icon correctly adds item to favorites on parse.com and toggles the favorite icon accordingly. |
| 10% | Gallery Activity — Remove from favorites: Clicking on favorite icon correctly removes item from favorites on parse.com and toggles the favorite icon accordingly. |
| 10% | Gallery Activity — Show Favorites Menu Item: Shows the list of photos that was marked as favorite |
| 10% | Gallery Activity — Clear Favorites Menu Item: Clears all photos that were marked as favorites from parse.com and refreshes list accordingly. |
| 10% | Gallery Activity — Show Shared Menu Item: Displays photos shared with current user, and displays the list items correctly. |
| 10% | Gallery Activity — Clear Shared Menu Item: Deletes all the items shared with the current user from parse.com , and refreshes list accordingly. |
| 10% | Gallery Activity — Sharing Stats Menu Item: Displays the alert dialog containing user sharing stats and sorted correctly. |
| 10% | Details Activity — Share button shows list in alert dialog of list of users |
| 10% | Details Activity — Selecting a user from dialog shares current photo item and saves the required information correctly. |