CST2335 Graphical Interface Programming – Draft

Final Project Assignment

Although this is a group project, you must only work on your part of the project that you have selected. You must write all of your own code that makes up your part of the project. If you use code that someone else, including your teammates, then that is considered plagiarism and you will get a mark of 0 for the final project.

Purpose:

The Project is assigned to give you experience in:

- Developing software in a group environment, including using Github to merge code into 1 project.
- Dividing workload to meet deadlines.
- Designing modular software that allows for that division of work.

Part 1 – Choosing your team

This is a group project for groups of 4 members and can be from any lab section in this course. If you would like to be assigned to a group, then send an email to your lab instructor. If you have chosen partners yourself then please email the names of the group to your lab instructor. Once you have a team for the final project, each member of the group should select a different one of the given topics. You should then fill out the Group Activity Worksheet together to exchange contact information amongst your group (Algonquin student email addresses at a minimum). Each member of the group must then upload the Excel worksheet on Brightspace, under the FinalProject assignment link on or before ***July 5th, 2024. This is worth 5% of your final project mark***

Beginning Steps

- One person should create a new project for the team and then upload it to github using the menu option "VCS" -> "Import Into Version Control" -> "Share project on GitHub".
- That group member must then invite the other group members to contribute. This is done by clicking on the "Settings" tab in Github, then click "Collaborators" on the left side menu, and search the group member names to add them to the project. Other team members should then clone that project to their computer and start making branches for their work. From AndroidStudio, select "File" -> "New" -> "Project from version control" -> "Git" and then paste the git URL from the main github repository from the previous step. ***You will not be able to integrate your work if you do not start by first cloning the project!***
- Then write your own code on your own branch and then merge that branch on Github (after each requirement is finished). Don't try to merge the code only on the last week.

Part 2 – Programming your application. Requirements:

In your group project, you will have a main page that shows once your application is launched. From this page, there should be 4 buttons that launches a page that is the landing page for your part of the project. In your part of the project, you must implement the following requirements:

1. You must have a ListView that lists items that were inserted by the user.
2. There must be a TextField along with a button that lets the user insert items into the ListView.
3. You must use a database to store items that were inserted into the ListView to repopulate the list when the application is restarted.
4. Selecting items from the ListView should show details about the item that was selected. On a phone would use the whole screen to show the details but on a Tablet or Desktop screen, it would show the details beside the ListView.
5. Each activity must have at least 1 Snackbar, and 1 AlertDialog to show some kind of notification.
6. Each activity must use EncryptedSharedPreferences to save something about what was typed in the EditText for use the next time the application is launched.
7. Each person's project must have an ActionBar with ActionItems that displays an AlertDialog with instructions for how to use the interface.
8. There must be at least 1 other language supported by your part of the project. If you are not bilingual, then you must support both British and American English (words like colour, color, neighbour, neighbor, etc). If you know a language other than English, then you can support that language in your application and don't need to support American English.All activities must be integrated into a single working application, on a single device or emulator. You should use GitHub for merging your code by creating pull requests.
9. The interfaces must look professional, with GUI elements properly laid out and aligned.

10. The functions and variables you write must be properly documented using JavaDoc comments. You must create the JavaDocs in a JavaDocs folder like you did in the labs.

Grading Guide

- Each student is graded on his or her application separately. ***This counts for 85% of your project mark.***
- Week 13 All code is merged on your group's github repository You should only be doing merge fixes and fixing crashes the final week.
- Week 14 of the semester - Project demonstration during your scheduled lab demonstration found on the demonstration schedule. You will show each of the 13 requirements from the list. Arrange a single submission of the group deliverable by one of the group member's computer on behalf of the entire group. You must be in the lab in person to answer questions about your work. Code submitted on Brightspace will not be marked.

Part 3 – Submit your source code on Brightspace

- Before your group demonstration, each member of the group must submit their final code as a record of what was finished at the end of the project. From your github repository, there is a link for "Clone or Download". Select the Download option and save your code as a zip file on your computer. Then upload that zip file to Brightspace using the FinalProject link. ***This is worth 10% of your final project mark.***

The Application Topics

Each of the applications requires similar programming techniques. Each application takes information from the user and stores it in a database. They can then view the data saved to a list of favourites and delete items from that list. Beyond that you are free to get creative. There are 4 topics listed below so each member of your group should choose a different topic to implement the requirements listed in part 2 above.

Customer list page

This page will be for adding new customers to a database of customers. You will be able to add, view, update, and delete customers.

- Your application should have a button for adding a new customer. When the user presses this button, there is a page that lets the user enter the customer's first and last name, address and birthday. You should check that all fields have a value before letting the user submit the new customer.
- Once a customer is added, they should appear in a list of customers, as well as be inserted in a database. Selecting a user from the list should show the customer's details in the same page as when creating the customer. However now instead of a submit button, there should be an Update and Delete button. The update button just saves the customer's updated information and delete removes the customer from the list and database.

When a user adds a new customer, the user should have a choice to copy the fields from the previous customer or start with a blank page. You should use EncryptedSharedPreferences to save the data from the previously created customer in case they want to copy the information.

Airplane list page

This page will simulate an airline keeping a list of airplanes that it has in the company. You should be able to add new planes to your company's list of planes, view current planes or delete planes that were sold or are too old to fly anymore.

- Your application should have a button for adding a new airplane. When the user presses this button, there is a page that lets the user enter the airplane type (Airbus A350, A320, Boeing 777, etc), the number of passengers, the maximum speed, and the range or distance the plane can fly. You should check that all fields have a value before letting the user submit the new airplane type.
- Once an airplane is added, they should appear in a list of all airplanes available to a company, as well as be inserted in a database. Selecting an airplane from the list should show that plane's details in the same page as when creating the airplane. However now instead of a submit button, there should be an Update and Delete button. The update button just saves the airplane's updated information and delete removes the airplane from the list and database.

Flights list page

This page will simulate an airline keeping a list of flights between two cities. You should be able to add new flights to your company's list of routes, view current flights or delete flights that are no longer offered.

- Your application should have a button for adding a new flight. When the user presses this button, there is a page that lets the user enter the departure and destination cities, as well as the departure and arrival times. You should check that all fields have a value before letting the user submit the new airplane type.
- Once a flight is added, they should appear in a list of all flights available to a company, as well as be inserted in a database. Selecting a flight from the list should show that flights' details in the same page as when creating the flight. However now instead of a submit button, there should be an Update and Delete button. The update button just saves the flight's updated information and delete removes the flight from the list and database.

Reservation page

This page will simulate an airline booking a customer on a flight. You should be able to add new reservations for customers on a given flights.

- Your application should have a button for adding a reservation. When the user presses this button, there is a page that lets the user select an existing customer onto an existing flight, on a certain day. Assume that a flight only happens once per day, and the flights repeat every day, meaning that you don't have to worry about the case where flights are only available on certain days, like Wednesday and Fridays. Assume that flights happen every day. If you want multiple flights between the same cities, but at different departure times, then those would have to be entered as different flights (for example AC 456, AC 457, AC 345 could all be travelling from Toronto to Vancouver but have different departure times)
- Once a reservation is added, they should appear in a list of all reservations made with a company, as well as be inserted in a database. Selecting a reservation from the list should show the reservations details like customer name, departure and destination cities, as well as departure and arrival times.