**A logo for a university

Description automatically generated**

**LAB 3**

Department of Software Engineering

SOFE 4650U: Mobile Application Development

Teaching Assistant: Austin Page

Prepared by:

**1. Activity Screens**

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Default Note Edit Screen

Loaded Note on Edit Note Screen

Saved Note on Home Screen

Add Image Functionality

**2. Code Functionality**

A screen shot of a computer

Description automatically generated

Setting an onClick listener for the image button on the Edit Note screen to call the imageOptionsMenu function.

A computer screen shot of a program code

Description automatically generated

Creating an AlertDialog object upon calling the function with the option to take a photo, choose a photo from the gallery, or cancel the photo. The option clicked calls the correlating functions takePicture, selectImage, or dismisses the dialog menu.

A computer screen shot of a program code

Description automatically generated

The functions takePicture and selectImage which create their respective intents using ACTION\_IMAGE\_CAPTURE and ACTION\_PICK respectively and sending a request code with the intent which opens the camera to take a photo or select an image from the image gallery.

A computer screen shot of a program code

Description automatically generated

The overrided function that acts according to the request code parameter. If the request code is the GALLERY\_REQ\_CODE, the URI data is retrieved and used to set the image button’s image to display the selected photo. Otherwise if the CAMERA\_REQ\_CODE is the request code, the photo taken is converted to a Uri using to bitmapToUri function, and is then used to set the imageButton’s image to the photo taken.

A screenshot of a computer program

Description automatically generated

The bitmapToUri function that temporarily creates a file to store the bitmap to which is needed to convert the file to a Uri.

A screenshot of a computer program

Description automatically generated

Upon saving, this if-else block is used to detect if the note is associated with an ID. If not, the system understands that the note is a newly created note and is saved as a new note in the database via the database.insert command. If the note is associated with an ID already, the note’s row is instead updated.

A screen shot of a computer

Description automatically generated

On the NoteAdapter.java class (class which provides functionality for all notes displayed on the Home screen), a setOnClickListener function is used to detect when a note is clicked.

A black rectangle with white text

Description automatically generated

On the Home.java class, the Adapter is implemented, and the onItemClick method opens the New Note (or Edit Note) screen.

A screen shot of a computer program

Description automatically generated

The beginning of the loadNote function, which checks if a note is being loaded in. It references the database schema and uses a database.query function to point the cursor at the desired note.

A screen shot of a computer

Description automatically generated

The end of the loadNote function, which sets the views of the Edit Note page to be filled in with the database’s information of the note specified.

**3. Lab Experience Report**

The goal of this lab was to load notes from the Home page to the Edit Note page and provide the Edit Note page with functionality to add images to the note and save to as well as load from the database. I created the image display on the edit note page by clicking the adding an image button to the corresponding XML file, doubling as a button for image options and as an image display. Next, I used a setOnClickListener() with the image button to call the imageOptionsMenu() function which creates an AlertDialog UI for the user to select if they would like to upload an image from the gallery, or a live photo from the camera. Depending on their choice, the appropriate intent would be launched in the takePicture() and selectImage() functions. The onActivityResult() function would read the code sent by the intents to understand which intent was called upon it and sets the image button’s display image to the photo accordingly (since live photos are bitmaps and the gallery photos are URI’s, the process to have them displayed using URI data is different). When saving the note, the URI is stored in the database. When loading a note, the cursor is pointed to the note selected, and each view is filled in with the column data associated with it.

The challenges of this lab all had to do with the coding aspects of what was required, as I had to research every bit of functionality I wanted to add. Such examples include finding out how to invoke the gallery and camera apps of an Android phone, how to update the display image for an image button, and how to store and load images onto the display image for the image button. Loading the data from a note also proved a bit difficult as I did not initially know I had to interface the Home screen with the NoteAdapter to give every displayed note item their individual onClick listener.