CSE 382
Fall 2022
Final Project

Background. You will soon complete Project #5, which is the last of this class' regular projects. A major purpose of Projects #1 through #5 was to allow you to develop smallish applications that utilize the techniques learned during the instruction portion of the course.

For this assignment you will develop a project that is larger and more sophisticated than the previous projects. Your final project will be highly polished and incorporate many of the aspects covered in this course. This project replaces a final exam as the class' culminating experience. Because of this, it is expected that the final project be allocated more of your time than the regular projects. To support you in your development of a high quality application, the following observations are made:

- Because there is no final exam, you are not required to spend time studying for a final exam or taking the final exam. It is expected that a portion of this savings be put into your final project.
- Overall, weeks 11-15 will include fewer coding exercises than normal. And, overall, the weekly volume of new instructional materials during the final weeks of the semester will be less than that covered in the early weeks of the semester. It is expected that students allocate the extra time toward their final project.
    - Some of the material covered during the later weeks will obviously be relevant to mobile apps but might not be incorporated into your final project. Nonetheless, you are expected to keep current on these materials.

Purpose. You will pick an application that you would like to design and implement. The purpose of this project is for students to create a highly polished app that they have personal interest in. The following summarizes the main points of this project:

- Allow students to implement an application that matches their personal interests.
- Provide extended time to allow students to develop a highly polished application that incorporates many of the techniques learned during the semester.

What application should you develop? You are free to create an application of your choosing. Here are some thoughts on that topic:

- Pick an application that you would like to see working and perhaps even use. The most successful ones have been those that the student had a personal interest in.

- The application must have a reasonable way to incorporate the required features and secondary features that are listed below. For example, your app must make use of a local database and have multiple pages, to name a few.
- Here are a couple nice examples of final projects that have graciously been provided by CSE 382 students from previous semesters. These can serve as examples that have a good level of polish and sophistication. Please note your application may NOT be modeled after these examples.
    - Akshita Lathar
    - Emma Gardner
    - Kyle Lierer
- Your selected application cannot be based on a tutorial that you have found online. You may, however, model your project after an existing, commercial, app but you must mention that app during your presentation.
- Some ideas that have been done in the past (which you can also do):
    - Meal tracker
    - Recipe storage, recipe creating, browser for external recipe webservice
    - Contact list
    - Task manager (to manage the things on your todo list)
    - Word games
    - Garden application that keeps track of the plantings in a vegetable garden
    - Symptom tracker (for individuals that want to gather data to take to their doctor)
    - Interfaces to existing web services
    - Keep track of favorite recipes
    - Unique and sophisticated calculators/converters
    - Display statistics of sports teams
- You should not simply extend the projects that we have previously developed in this course. That is, your final project should not essentially be a reproduction of one of the apps done earlier in the semester. Also, do not drop a calculator into your app simply to increase the size of the app. The app you select should utilize techniques learned during the semester but should also be applied to an application area that is new to our class.
- Some areas that are discouraged are those that need an external server that needs to be logged into and/or an external database that allows both reading and writing. We have not done those things in this course. Of course, this is appropriate if a student has the experience to perform these types of tasks.

Required Features. The application area can be anything you wish but the app must include features as described below. The features that you integrate into your application must add positively to the application's effectiveness; you should not simply drop in, say user preferences, if it does not enhance the application.

Required Features Must include all of these	**Secondary Features** Include at least three of these

● Local database
● Multiple pages
● ListView
● Images
	● Web Services (this is not a WebView or a web browser)
	● Meaningful user preferences
	● Layout changes based on:
	  ○ Phone vs tablet or
	  ○ Portrait vs landscape
	● Graphics
	● Sound
	● ‡ MVVM
	● Some other, instructor-approved, technique

‡ Topics to be covered in the coming days

Deliverables (see daily outline and Canvas for due dates)

- **10% <u>Project Proposal</u>.** You will develop 2-3 slides that describe the app that you will implement. You will record a presentation of your slides. Your goal is to provide the instructor with enough information to understand the app you plan to create, allow the instructor to provide feedback to improve on your idea, and ensure that the level of sophistication is appropriate.

  Your 2-3 slides should succinctly address the following:
  - Provide a top-level description of your application.
  - A list of the main pages of your application and the navigation approach you plan to use.
  - A mock-up of your pages.
  - How each of the main requirements (e.g., Local DB, etc) will be fulfilled.
  - Your tentative plan of how you plan to fulfill the secondary requirements: (e.g., web services, etc), if you have one.

- **10% <u>Midpoint demonstration</u>.** You will record a video (maximum of 4 minutes) demonstrating the current work that you have completed on your project. You should be about 50% complete. It is possible that one student spent a lot of time setting up data structures and the database and little time on the GUI. Alternatively, another student might have spent a lot of time on the GUI. Regardless of your approach, you should be about 50% done with the project. In your video:
    - State whether you are 50% complete.
    - Do a quick review of the source code that you have completed and what still needs to be done.
    - Run your code. Demonstrate as much as you can and identify what still needs to be done.
    - Your estimate if you are going to be able to finish your proposed functionality.
- **80% <u>Final deliverables</u>**
    - Push project source code to Git.
    - Submit your video demonstration to Canvas.

Proof of concept. The level of functionality and polish of this project should be about twice that of our regular projects (timewise). Because you do not have unlimited time to spend on this project you can make certain simplifications, as long as you provide the basic functionality. For example, if you are implementing a meal tracker, you can populate your database with 10 different foods, with the idea that 100s could be inserted if a production version of the app were to be created. For this, adding 100s of food items is a methodical task that could easily be accomplished with enough time.

Deliverables and scoring.

- **(10 pts) <u>Project proposal</u>.** You will be scored on:
    - Your organization and quality of your slide and presentation.
    - Clarity and organization
    - Your application is well-thought-out and has a level of sophistication appropriate for a final project of an upper-level CSE course.
- **(10 pts) <u>Midpoint demonstration</u>.** You will be scored on the following criteria:
    - 10: Outstanding progress. The demo shows excellent progress toward a highly functional and polished project. Note, the project does not need to be completed, or polished, at this point; it simply needs to demonstrate enough high-quality artifacts to indicate that effective work has been taking place and the project will be completed at the due date.
    - 9: Very good progress. The demo will instill confidence that the project is in a good state and that similar future efforts will result in a highly functional and

polished app. The professor should be confident that the project will be likely completed at the due date.
    - 8: Good progress. The demo shows solid progress but might be slightly behind schedule.
    - 6-7: Moderate progress made but clearly less than should be accomplished. Demos in this category will have raised doubts about whether the final product will be highly effective and highly polished.
    - 0-5: Minimal progress made. Scores in this range will be given to demos that indicate that little progress has been made and it is questionable whether the project will be successfully completed.
- **(80 pts) <u>Final deliverable</u>.**
    - **(70 pts)** Operation of the app should be complete, accurate, and highly polished. Some of the main items to be considered:
        - Your source code must be submitted to Git.
        - The functionality of your app should be at a level of sophistication that is approximately twice a regular project. The most obvious and important functionality relevant to the app must be included.
        - The program should not crash and should take steps to ensure the user avoids mistakes (e.g., using a Picker instead of entering text).
        - The app should use contrasting colors, appropriate text size, alignment, frames, etc to provide the user the ability to easily understand the information being presented to them. In particular, ListViews should be presented with icons, justification, font colors, font sizes, etc to promote readability.
        - The navigation patterns should be appropriate for the given task.
        - The application should include the required features and have the appropriate number of optional features.
    - **(10 pts)** Video demonstration. In this demonstration, you must:
        - Show the use of your application. While doing this, you explain what you are doing and how the project requirements are met. Make sure that your audio is turned on. While you are demonstrating your app, call out the underlying features that you used to support the required functionality and the secondary features. While doing your demonstration, show the common error situations and that your program does not crash.
        - Demonstrate error conditions and how your program handles them. In particular, if there are places the user can type in numbers, demonstrate that your application does not crash if non-numeric information is entered.
        - The length of your video should be at most 10 minutes long. Here are the deductions for going over 10 minutes:

- 1 minute too long: 0 point deduction
- 2 minutes too long: 1 points deduction
- 3 minutes too long: 2 points deduction
- 4 minutes too long: 3 points deduction
- 5 minutes or more too long: 5 points deduction