Final Project Submission (100 points)

This concludes the project. In this segment, you will make final revisions to your software, make a presentation to the class, and write a report about your project.

- Get Feedback – see details on next page (15 points)
  - Describe the feedback plan (7 points)
  - Describe the feedback plan execution and report the results (8 points)
- Refined Application (40 points)
  - Your application should be improved and refined from the initial working prototype. You should include a list of the changes that were made, with a brief justification as to why it was changed or refined (e.g. it was part of the initial design, was revealed as necessary by the feedback, etc.). As reflected by the point value for this portion of the final submission, it is expected that there will be significant improvement from the initial working prototype.
- Presentation & Demo (25 points)
  - Each team will have 20-25 minutes for their presentation including questions, so you should plan on taking no more than 20 minutes so there are a few minutes for questions.
  - Complete peer-review reports
  - You should cover the following points in your presentation:
    1. Name of the project and team members
    2. Statement of the problem and current competitive solutions
    3. Your approach to the solution
    4. Demo (or slides/video of application)
    5. How you improved your application based on the feedback you received
    6. Lessons learned, limitations of your app, future work, and why we should promote your app (for this portion you can treat the audience as an investor panel)
- Report (15 points)
  - This should be a standalone report that can be read on its own. You are free to copy and reuse elements from previous phases of the project.
  - Your report should include the following points:
    1. Introduce the problem
    2. Credits: describe which team member did what
    3. Describe your audience or target user population
    4. Describe competitive solutions and the deficiencies of those solutions
    5. Describe your solution
       a. Give an overview of your approach
       b. Present all screens you have designed with commentaries to explain (do screen grabs)
       c. Show transition diagram for the screens in your design (it is sufficient to use screen names as long as it is unambiguous what screens are being referred to)
       d. Describe your tutorial/help/any assistance you give to learners of your system
    6. Describe the feedback that you got on your initial working prototype: how many participants you had and their demographic characteristics (but don't include personally identifying information). Summarize the problems you discovered and describe what you changed in your final prototype to address those problems.
    7. Finally, describe open issues you are aware of, and what the next things you would do if you were to continue working on this project.

Submission

1. Add a link from your overall project page to this phase
2. Submit your final project code and report via the Canvas submission page

Get Feedback (15%)

Using the initial working prototype from the previous phase, you must design and conduct a usability test of your application in order to get feedback to improve your application. This deliverable has two parts: the feedback plan and a report of feedback results.

Feedback Plan

- Prepare tasks to ask participants to complete. These should be representative tasks, for example, if you developed a mapping application one task would be to find and navigate to a certain location.
- Describe the scenario for your application to your participants. Ideally the participants should be representatives from the target user population for the application.
- Give them a few (3-5) tasks to perform. Each task can include a simple sequence of items to do or more complex, open-ended, challenging tasks. Choose tasks that are clear enough so that users know when they are done and how to signal they are done.
- Create a brief survey or questionnaire. Depending on whether your application is intended to have a certain effect, you can do a pre- and post-questionnaire.
  - You may want a few pieces of personal data, such as gender, age, what kind of phone they have, how familiar they are with Android, number of years of experience in the domain of the application (e.g. if a campus oriented application, what year they are). Your may also need to verify that they have proper vision, hearing, mobility, etc. This is so you can report about who your participants were. Please refrain from personal identifying questions, that is, do **not** ask for name, social security number, address, etc.
  - You should ask 3-8 questions about their subjective satisfaction so you get an idea of what reactions they had, what problems they found, and general comments. You can use semantically anchored 1-7 Likert scale questions to ascertain how difficult-easy they found specific aspects of the interface, or ask them to describe the best and worst parts of the interface, what they liked or disliked, or ask for general comments. Your goal is to get information that will help improve your interface design, and determine if it is acceptable or in need of further improvements.

Execute the Plan and Report Feedback Results

- Conduct your feedback plan with 3-12 participants (this is on top of those that give you feedback in our class). You can get feedback from those in the class, and can report on it, but please get more feedback.
- Describe briefly the interactions participants had with your application. Analyze and summarize the findings from your survey. For each participant, give a one sentence description of who they are without personally identifying information (e.g. 24-year old female college student, who bicycles to campus 2-4 times a week), followed by a half page description of how they did (e.g. where they had difficulties), and what they said (e.g. “This is cool! I’d love to have this, especially if I could add my own schedule.”).
- Report on the survey results using descriptive statistics and bar charts, plus extracts of key comments made by users on open-ended questions like what it was they liked or thought was best, or what they disliked or thought was the worst part of the application.