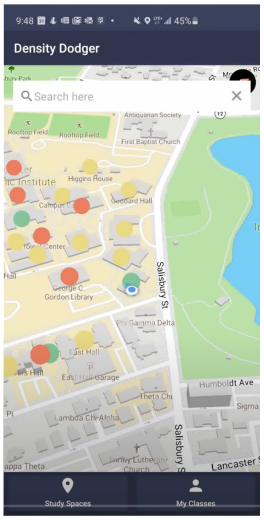
Term Project (800 Points)

For the term project, expectations are again based on two-person teams. Students may choose to complete the term project individually. Larger teams are permitted only with prior approval from the instructor and must submit an expanded proposal to ensure the project reflects a more ambitious scope and meets all course learning objectives.

We are approaching the end of the term, and I hope you have learned a lot about mobile computing and are ready to put what you have learned into this open-ended final project.

In this project, you will work on **a mobile application** of your choice by applying the concepts and mechanisms we have learned throughout this course. This week-long project provides an opportunity for you to fuse not only what you have learned in this course but also what you already know coming into this course. Simply put, the term project allows you to exercise the key techniques of this course in your personalized ways.

If successful, this project will add a unique experience to your portfolio that will help you to stand out in your future pursuits. *We encourage you to think big, be ambitious, but also importantly, to have fun with this project*. Here are a few screenshots from previous years' projects that I hope can spark your inspirations!



Overview

The term project is designed to provide students the opportunities to engage with each other, acting as both the potential users and the developers, and deliver a working prototype based on peer feedback. Broadly, the term project has two components:

- The **in-class component**: Student teams will participate in round-table discussions. Each team will take turns presenting their work, acting as potential users to provide peer feedback, and working on a feedback summary writeup.
- The **outside-class component**: Student teams will work to produce a project proposal, a minimal working prototype, and a final project teaser video and presentation.

There are many moving parts of this term project, which I take great care to simplify the process to make it a fun and rewarding process. Below, I summarize the key information for this term project. See Phase 1 and Phase 2 for more details.

Phase	Deliverables	In-Class Component	Points
Phase 1: Proposal & Feedback Summary	- Project Proposal (PDF or slides) - Peer Feedback Summary (signed)	Roundtable Presentation & Peer Review	215 pts
Phase 2: Final Prototype & Presentation	- APK file - Source code (cleaned up) - Teaser video (≤1 min) - Presentation slides	In-Class Demo & Presentation	585 pts

Specifically, the term project will consist of **two phases** to help teams earn points and make consistent progress. Here, we present a high-level description for each phase:

- **Phase 1: Project Proposal and Feedback Summary**. Each team will brainstorm to propose a term project relevant to this course. Upon identifying the project, the team will create a project proposal, which they will bring to the in-class round table discussions.
- **Phase 2: Minimal Working Prototype and Presentation**. Each team should complete a minimal working prototype and create a 1-min teaser video about the finished project. The video should clearly demonstrate the working of the main features. Each team will also create a presentation and present to the entire class to showcase their finished project. The presentation should clearly describe what the project is about, the key technologies used, and end with the teaser video.

Important: to earn points for the in-class components, you will need to actively participate in the corresponding activities.

Phase 1: Project Proposal and Feedback Summary (215 points)

The key goals of this phase are for teams (1) to create **a proposal document** that clearly articulates the proposed project, and (2) to engage with peers in discussion and provide constructive feedback, and (3) to identify ways to address the feedback for the next Phase 2.

The proposal document will be used by the team to present their project during the in-class round table discussion. You are free to choose the format for the proposal document between *a writeup PDF* and *PowerPoint slides*.

Your proposal document should clearly convey three key pieces of information: *(i)* **what** you are planning to do for the term project; *(ii)* **why** you are interested in doing the said project; *(iii)* and **how** you are planning to achieve the proposed features. As always, remember to include relevant citations.

To help you scope a term project with the right amount of workload and difficulty, we provide the following high-level baseline goals that each project needs to achieve. Specifically, these goals are grouped in three categories:

- **UI complexities**: Below are example requirements you can meet to satisfy this category. You may either implement these directly or propose more **demanding** alternatives.
    - Your app should have at least **two screens**. We recommend following the best practice of using Fragment, but you are welcome to use Activity.
    - Your app should use at least **two different ViewGroups** to arrange and manage widgets. For example, you can use RecyclerView and ConstraintLayout.
    - Your app should use intent extra to share at least **one piece of data** between the screens.
- **Mobile sensing and processing**: Your app should demonstrate interaction with the physical environment and/or external systems. You may fulfill this category using the examples below or propose more demanding alternatives.
    - Your app should use at least **one sensor**, e.g., a camera or a microphone, and present the corresponding information to the user.
    - Your app should be able to **fetch data from an external entity**, e.g., NFC with another mobile device or a Retrofit-based network component that communicates with either a local server or a managed cloud service.
- **Data persistence**: Your app should be able to remember relevant application states. You can meet this category using the examples below or propose more demanding alternatives.
    - Your app should properly handle the **screen rotations**, i.e., the relevant UI data should persist across screen rotations.
    - Your app should **save some app state** properly. When users come back to your app, the data should still be there. For example, you can use a long-term persistence strategy, such as using local filesystems or SQLite databases.

Your proposal document should clearly articulate your plan for completing the requirement for **all three categories** above. To clearly convey the design, teams often include UI mockup using either Layout Editor in Android Studio or third-party tools like Figma.

In-Class Discussion and Peer Feedback

To participate in the discussion, you will need to have your proposal document ready. The in-class discussion and peer feedback activities are designed with the following goals in mind.

1. Provide each team an opportunity to present technical contents in a small group setting. Such presentations are common requirements in professional workplaces.
2. Allow each team to act as potential users of other teams' proposed projects and think from the potential user's perspective to provide constructive suggestions on features.
3. Expose each team to a few other project ideas which might contain similar design and therefore allow teams to share insights and information during this phase.

Each team is expected to take detailed notes during the discussion and accurately capture peer feedback in the reviewers' own words. After the discussion, the team should identify common themes across the feedback and provide a concise summary. Finally, the team should generate a list of action items to help address the feedback received.

Please use the provided feedback summary template as a starting point for your write-up.

Phase 1 Deliverables

For this phase, you will submit the following files in a single .zip via Canvas after the in-class discussion.

- **The project proposal document**. If you receive a lot of feedback during the in-class discussion, you need to modify the project proposal by considering that feedback. You don't necessarily have to address all feedback, but you should clearly indicate which feedback you have addressed and the rationales for not addressing other feedback.
- **Discussion and feedback summary for your proposed project.** You need to have the other teams to sign off on the discussion and feedback summary, as indicated in the template.

Grading Rubric

The teaching staff will assign points to your proposal document and the feedback summary document based on the description below. Points for this Phase are roughly divided among the proposal documentation (done outside the class) and the feedback summary documentation (done during the class).

Requirement	Brief Description	Points total
Proposal document: problem statement	Clearly described?	15

Proposal document: motivation	Well motivated?	15
Proposal document: UI complexities	Clearly articulate how the proposed project will meet the baseline or more advanced UI feature goals	30
Proposal document: mobile sensing and processing	Clearly articulate how the proposed project will meet the baseline or more advanced sensing/networking goals	30
Proposal document: Data persistence	Clearly articulate how the proposed project will meet the baseline or more advanced persistence goals	30
Feedback summary	Nicely summarize the peer feedback?	60
Action items generation	Generate action items for next phase?	35

Phase 2: Minimal Working Prototype and Presentation (585 points)

The key goal of this phase is for teams to create a minimal working prototype for the proposed project. I encourage you to explore third-party libraries to build the minimal working prototype. The prototype should be implemented by incorporating the peer feedback from Phase 1.

To demonstrate the successful completion of this phase, you should create a teaser video that clearly demonstrates how the prototype satisfies each of the required categories and how the feedback was addressed.

Additionally, you also need to create a presentation to be used for in-class presentation.

In-Class Presentation

Each team will take turn to present their minimal working prototype using slides and the recorded teaser video.

Here are some more details about the presentation slides:

1. The presentation slides will be graded based on both content and delivery. Each team needs to **sign up via the Google Sheet.** This helps the teaching staff determine the presentation order and plan the in-class presentation logistics.
2. The signup sheet has pre-populated team IDs. When you sign up for a slot, also write down the team ID and make sure to include the team ID in your presentation slides.
3. The estimated presentation time per team is **3 minutes** based on the number of teams. This time also includes the time you should show the teaser video.
4. Teams should coordinate among themselves when delivering the presentation. All members should have an opportunity to talk during the presentation.
5. There are no limitations on how many slides you need to have, provided you can finish them within the allocated time.

Teaser Video Guidelines:

- Duration: ≤1 minute
- Format: .mp4, .mov, or .webm
- Includes: Audio narration describing the app; optional background music
- Tip: Browse past projects for inspiration *recordings of previous projects*.

Peer Recognition Awards (Optional but Encouraged)

To celebrate everyone's creativity and technical work, we'll also be hosting a **peer voting session** for the final presentations.

Each team will have a chance to vote for projects in the following fun categories:

- **Most Technically Impressive**
- **Best UI/UX**
- **Most Creative Concept**

Votes will be anonymous, and top-voted teams will be recognized. This is a great way to share appreciation for each other's ideas and effort—no extra work required outside the class!

Phase 2 Deliverables

When submitting this part, please include the following and compress all the files together as a single .zip. As with all projects, please only use standard zip files for compression; .rar, .7z, and other custom file formats will not be accepted.

- **The apk for the finished project**.
- **Presentation slides**.
- **The 1-minute teaser video**. If you have a large video file that can't be uploaded to Canvas, you can include a link where the teaching staff can download the video.
- **The project source code**. You are expected to perform reasonable cleanup, e.g., removing unused codes and organizing relevant files to packages, and include good documentations.

Grading Rubric

Requirement	Brief Description	Points total
Functional apk (required)	App can be installed and interacted with	10
1-min teaser video	Clearly demonstrates main features with narration	100
In-class presentation: content	Clearly explains project goals, design, and technologies	50

In-class presentation: delivery	Delivered on time; clear and well-coordinated	50
Finished project: UI complexities	Meets baseline or more advanced UI feature goals	100
Finished project: mobile sensing and processing	Meets baseline or more advanced sensing/network features	100
Finished project: data persistence	Meets baseline or more advanced persistence strategies	100
Finished project: end-to-end integration	App components are coherently connected and functional	50
Finished project: source code professionalism	Clean code structure, appropriate comments, organization	25

Misc

Term Project Ideas and Suggestions

As always, this term project, as well as this class in general, are meant to be low-stress and fun. If you want help, please feel free to ask for help via Canvas or Slack or to discuss your ideas with others.

If you already have a project idea but are not sure whether it fits into the term project scope, please don't hesitate to visit us during the office hours. If you need some inspirations, here are a few suggestions:

- You have probably used a lot of mobile apps (maybe not necessarily Android apps), so tap into this reservoir. Pick one app that aligns with your interest. For example, if you are interested in learning foreign language, you might find apps such as Tsurukame - For WaniKani a good inspiration. Just be aware that it will be challenging to reproduce the entire feature set of Tsurukame in a term, so you will have to prioritize.
- Another avenue to look for inspiration is from your popular websites. Are there any sites that you use frequently that you wish they have a mobile app companion? Again, the pro of going down this route is requirement engineering should be relatively easier (compared to starting from scratch). Sometimes, those websites will provide APIs (e.g., the Flickr API) you can use directly, but sometimes you will need to setup your own backend server for mocking up the data connection.
- You can also check out a list of apps developed by students from prior years. Even if you are not going with the exact same project idea, these prior projects can be informative regarding the project scope.

Optional Challenge Themes

If you're looking for inspiration or want to challenge yourself further, consider building a project around one of these broad themes. These are **not required**, but may help you focus your design and innovation:

- **Sustainability**: Apps that promote environmental awareness, reduce waste, or support eco-friendly habits.
  Example: a smart recycling guide that uses camera input and image classification.
- **Social Good**: Tools that support well-being, accessibility, or community outreach.
  Example: a journaling app that uses voice input and sentiment detection.
- **Tech for Daily Life**: Utilities or companions that improve daily routines.
  Example: a smart grocery planner or a task-based study timer.
- **Learning and Education**: Tools to support self-guided learning or skill building.
  Example: a flashcard quiz app with accelerometer-based gesture navigation.
- **Fun and Play**: Games, AR experiments, or creativity tools.
  Example: a step-based adventure game that tracks motion with sensors.

You're welcome to combine multiple themes or go beyond these ideas—the goal is to create something personally meaningful or technically exciting to you.

Some Common Questions

Q: can we reuse source code from our previous programming assignments?

- Absolutely.

Q: I am not sure if [XX] satisfies the requirements. What should I do?

- You are always welcome to check with the teaching staff. The project is open-ended, and we trust you have a good judgment of what will be considered a reasonable design. When grading, we leverage our experience and best judgment to give you the points.