Deliverable 6 - Final Presentation & Arch + Design Document

SE2: Software Design and Architecture (CS446, ECE452, CS646)

Description

Conclude your project with a final presentation that demos your app and an written document that details the architecture and design of your app.

Final Presentation Demo

The format of the demo is 6 minutes presentation followed by 4 minutes Q&A.

Requirements:

- The demonstrations should both demonstrate what the system looks like from the user's perspective and describe some of the technical underpinnings / challenges you faced creating your system. The demo should show the major scenarios your tool supports (including the ones you proposed at the outset of the course).
- Please include all team members in the demo. The members who cannot make it to the assigned presentation slot can pre-record a video that will be played by other members in class. Each member is expected to speak either during the in-person demo presentation, in the pre-recorded video, or during the Q&A session. Those members who are present in the class during the demo should be prepared to answer all questions about the app (i.e., excuse like "another person worked on this part and I don't know" is not acceptable).
- You are allowed to pre-record a 3–6 min video and play it during the presentation. The same video can be uploaded to Youtube and earn the 2 bonus points.
- Your presentation, including any pre-recorded video, is limited to 6 minutes. The instructor will time your presentation and will give a hint when you have 1 minute left and 0 minute left. The instructor will stop your presentation if it goes 30 seconds over time.

Logistics:

- You can demonstrate your app from a mobile phone, tablet, laptops, or simulators on PC.
- The projector in the classroom supports HDMI connection, and the instructor will bring a HDMI to USB-C adaptor. If you need a different adaptor, please bring it yourself.

Arch + Design Document

Your project proposals provided an overview of the functionality your project aims to provide; they also provided some insight into the non-functional, or quality, requirements you want your project to exhibit. In class we have described a variety of architecture styles and design patterns, each of which has both strengths and weaknesses. The intent of this deliverable is to identify, describe, and justify the architecture and design of your project. Do not be constrained by the individual styles and patterns we described in class; it is expected that a heterogenous set of styles and patterns (including ones we may not have covered in class) may best suit your project.

In this document you will describe your project architecture (5 pages) and design (5 pages).

Requirements for architecture description:

- Describe how your system supports the functional requirements and non-functional requirements proposed in your project proposal document.
- Non-functional requirements should be specific enough to enable us to verify whether or not your app supports them in a measurable way.
- Clearly identify *2 architecture styles* used in your project and where they are implemented in the code.
- Use diagrams to aid in the description of your architecture. Include at least *a component view* and *a deployment view* of your system. You should aim for *3 pages of text and 2 pages of diagrams*.
- Note: a two-box server-client component diagram will be insufficient here. For example, every phone-based/external service you interact with (e.g., NFC, Geolocation, Contacts, etc.) should be clearly identified and can be treated as individual components.

Requirements for design description:

- Describe the design of your system, and its rationalization, such that a junior programmer could implement some subset of the system and integrate it appropriately. Your design should include a clear description of the structure of the classes and their externally-visible interfaces. Rationale must be provided documenting why you selected your design. The applicability of your design compared to alternative designs should also be referenced in this discussion.
- Highlight key patterns, classes, abstractions, and data structures / algorithms that are critical to the successful implementation of your system.
- Provide a mapping between the design-level entities (e.g., the classes) and the components described in the architecture is also required (or should be depicted on the diagrams).
- Clearly identify *2 design patterns* (except for Singleton) used in your project and where they are implemented in the code.
- Analyze how your design minimizes coupling and could accommodate future requirement changes. Think critically about how you could envision your system being altered and discuss how your design would support or inhibit evolving to meet those changed requirements; discuss at least *one example* of such changes.
- Use diagrams to aid in the description of your design. Include at least *a class diagram* that shows all of the classes and public APIs for your system and how they interact, and *a sequence diagram* that captures how your system behaves for each of the user scenarios. You should aim for *3 pages of text and 2 pages of diagrams*.
- In your diagrams and text, please clarify the physical location where the classes will reside if your are using a server-client architecture, and clarify any external API your system will use.

Your document should be a PDF document in letter size. The page limit is **10 pages** (excluding acknowledgements and references; see above for the suggested breakdown of these pages). The font size of the main text should be 11pt. The document should be uploaded to Learn; only one team member needs to upload this document. The file name should be
`cs446-d6_<group-number>_<project-name>.pdf` (use - instead of space in project name).

Assessment

The presentation is worth 5% of your final grade, and the document is worth 15% of your final grade.