Deliverable: #4 - Project Validation and Acceptance
Title: CS 410: Advanced Software Engineering

Deliverable Overview:

Your project proposals provided an overview of the functionality your project aimed to provide; it also provided some insight into the non-functional properties you want your project to exhibit. The intent of this deliverable is to motivate, describe, and demonstrate the testing and validation strategy you used for your project. The best way to think of this deliverable is this: you are a freelance development team who has just completed your app for a customer who has commissioned three teams to build the same app. Your goal is to convince the customer to enter into a long-term relationship with your team. To do this, you need to convince them that a) your app provides the functional properties in your proposal; b) that the non-functional properties promised in the proposal have been delivered; and c) that the code underlying the app is of high quality (e.g., unexpected bugs are likely to localized and easy to fix);

System Validation Strategy:

Building your apps involved coordinating four developers over a three-month period working on a non-trivial technical artifact. Given the breadth of your apps and their dependence on external APIs (from the phone or otherwise), frameworks (UI, or otherwise), and tools (e.g., Cordova, react-native, etc.) there are a wide variety of things that could go wrong. To mitigate risks, and to demonstrate that your project is of high quality, you should have developed a testing strategy which, at a bare minimum, should involve some form of automated test execution (e.g., where you type a one line command and a host of tests are executed).

For your project: What kinds of testing have you performed, and why are these appropriate? How many automated tests have you performed (counts / LOC are appropriate here)? How do you concretely decide that you have tested sufficiently? Why is this measure appropriate for your project? Are different components in your system tested differently? Which components are more tested and which are less tested? Did you have to change the structure of your system at all to support your tests? How? This description can be at most 3 pages, not including any attached test reports.

If you have any generated reports from your test strategy, feel free to include the URL or attach the HTML file to the end of the PDF.

Functional Property Assessment:

Your project proposal included a list of functional properties that your app was supposed to fulfill. Describe the status of each of these properties. The easiest way to do this is to include a bullet-point list of functional properties from your proposal document and indicate whether each was completed, rejected, or modified. For features that were modified or dropped, please provide a brief rationalization for the change. If your project met any tricky, interesting, or deal-breaking constraints or problems, please include a discussion of these as well. This description can be at most 3 pages.

Non-Functional Property Assessment:

In your project proposal your team identified three non-functional properties that your system should support. Provide concrete evidence demonstrating that each of these non-functional properties has been met (or not). A NFP is only considered complete if evidence can be provided; if this is not possible for one of your NFPs, explain why and describe an approach that could be used in the future to measure whether the NFP has been met. This description can be at most 3 pages; including any diagrams (if appropriate).

Architecture and Design Reflection:

In this section, describe specific divergences between your concrete project architecture and design (aka your final system as implemented) from the conceptual architecture and design described in Deliverable 2.

Separately, reflect on the work required to support your pivot; was it hard/easy? How long did implementing the pivot take? Describe the specific changes required to your system to support the pivot. If any architectural or design changes were required, mention those here. This section should be no longer than 1 page.

Personal Contribution Statement:

Each team member will provide a 1 page summary of their technical contributions to the project that should be included at the end of the document. In the architecture and design document every team member took ownership of specific architectural components. This description is an opportunity to provide _specific_ examples of technical contribution to the project, including how the code you developed contributed to the architectural components assigned to you. Other technical contributions are also important: if you worked on infrastructure, tests, or other activities, please describe these as well. It is important that the contributions be concrete: summarizing your contributions and backing them up with links to specific commits and issues in Github are key here. If your contributions exceed a page, please focus on the key contributions and summarize the rest. Essentially we are trying to gain more context into your contributions to your project than are captured by the summary Github provides:
https://github.com/CS410-2015Fall/<YOURGROUP>/graphs/contributors

Written deliverable:

1. Metadata: App name and team member details (names and Github ids).
2. System validation strategy.
3. Functional property assessment.
4. Non-functional property assessment.
5. Architecture and design reflection.
6. Personal contribution statement (one per team member, appended to the end of the document).
7. Only one team member needs to submit the document using the online system (http://coast.cs.ubc.ca:1337/) by **0800 on November 30**. PDF only.

Project Acceptance Meeting:

1. Sign up for meeting slot online for December1/2/3/4: Make sure your whole team can attend. Meetings will be in TBD; time is tight, please be on time or your presentation grade may suffer. The signup is first come, first served; please sign up using your team name and do not take any slot that has already been taken.
2. The project acceptance meeting is 30 minutes. You will start by giving a 5-10 minute oral description of the differences between your proposal and architecture to your final project and a description of how you validated your system and can measure your NFPs. No slides can be used, but you are encouraged to make use of the whiteboard and can refer to the previously-submitted written documentation.
3. The remainder of the time will be used to discuss your test strategy and final project status. It is likely that you will be asked to execute your automated test suite and describe some of your individual tests (with their source code), so make sure you have a development machine with you. The TAs will also want to interact with your app, so make sure you have it installed on at least two phones present at the meeting. The discussion will also consider content from your proposal and your architecture and design document as the TAs work to understand the final status of your project.

Doodle poll link:

- Sign up for your design review meetings using the links below. Make sure you sign up for a meeting with the right TAs, that all your group members can attend, and remember to list your group # and project name when signing up for the meeting slot.
- Your group-to-TA map is available here:
  https://piazza.com/class/ie68gehaz93mk?cid=37
- Giovanni & Bruce:
  http://doodle.com/poll/pxcfp94p6m57nk72
- Daniel & Sohrab:
  http://doodle.com/poll/437it2ebu7rie8fr

Assessment:

This deliverable is worth 15% of your final grade. Everyone on your team should understand both the mechanics of how your project was validated along with the specific aspects of the tests they performed. Ultimately your grade will depend on your team being able to provide convincing evidence that your project was tested effectively and appropriately.