CS130: Project Part A

Guidelines for Application Proposal

Deadline: Thursday, 10/24 11:55pm

Objective

In this project, you will use the skills acquired through lectures and labs to develop a new application. It can be any kind of application (e.g. web application, mobile app), it can be implemented in programming language of your choice (e.g. Java, Python, ...) and using any framework of your choice (e.g., ReactJS, Node, Ruby on Rails, ...). Our goal is to help you gain insights into the process of software design and development by working on this project.

First, you will start by identifying the requirements from a user perspective. Then you can design a solution in the form of an application to answer the user requirements and create a working prototype of the proposed application. Finally, you may identify key issues around and improve the prototype into a useful product.

Setup: Team Formation and GitHub Repositories

For this project, you will form teams of five or six students (preferably 5). The team should be formed before Part A is due. If your team needs a different size, please consult with your TA by the end of the second week. The team size cannot be changed in the middle of the project. However, the instructor can make an exception under extremely necessary circumstances, so please come and talk to us if exceptional situations arise.

Your team needs to set up a GitHub repository for your project and send an invitation to your section's TA as a watcher (read permission only). GitHub repositories are an essential means to team collaboration and communication, as well as recording the progress of development. So, make sure to commit your code into repository regularly in Part B and C. Your TA will be watching your repository's statistics in order to track the progress of your project.

Part A: Propose Your Idea

After your team is formed, you need to come up with an idea for your application that is both appealing and feasible to pursue. The feasibility of your project is crucial, since your team is supposed to implement the proposed idea and produce a working and test-able application for Part B and C in order to receive any grade for those parts.

Thus, when proposing an application in Part A, your team must investigate whether the application is feasible to implement and whether your team has the capability to finish the product within time limits (5 weeks). Please make sure to propose a set of requirements such that its implementation could be completed in the given time and you are able to demonstrate its progress in live demos during the Part B and C presentations. Thus, it is important to find a good balance between creativity and feasibility!

You must write a report to propose your idea and justify its feasibility. Also, you need to create mockup designs to demonstrate your idea and include its screenshots in the report. You are going to present your idea in your discussion section *on* Week 4. Each team is scheduled for a 6-minutes presentation followed by a 3-minutes Q&A. In order to reduce the time spent on presentation setups and transitions, your TA will download their teams slides in advance on their computer. So please submit the slides that you wish to use for your presentation by the deadline.

Submission Instructions

Make sure to include your GitHub repo URL in your report. Please submit presentation slides and the report (each as a single PDF file) on CCLE before the due date. **Only one person should submit the PDF files per team.**

Report Structure

Your report should include the following sections:

Title Page

Project Name:
Team Name:
Team Members (names and UIDs):
GitHub Repository URL:

Motivation

You must include a description on why the proposed application is needed by users, what the problem is, and what your vision is for creating the application.

Feature Description and Requirements

1. You must include a description of all of the proposed features and their requirements. Each feature should be explained verbally separately. You should also come up with different user stories or usage scenarios demonstrating how users might face each of the requirements in real life and how they can use your application in response to that need. You need at least 3 usage scenarios for your report.
2. You must include a UML use case diagram to specify the functional requirements. You may also include state chart diagrams to specify behavioral requirements. You may specify other non-functional requirements such as performance, persistence, etc. in English if necessary.
3. Please include mock-up screenshots of the proposed application and describe the associated user interaction scenarios in detail, so that the teaching team can understand the proposed features and assess the difficulty level of the proposed implementation as best as possible (e.g., similar to how user manuals are written.)

Feasibility

You must include a detailed description of how it is feasible to implement the proposed application. Here are several common factors that could lead to software development failures:

- A lack of familiarity with APIs
- Unable to deliver on performance
- Excessive cost
- Already existing project
- Too many features
- Features that are not useful
- Third party APIs/services that may not be reliable to use

1. You should justify how your application is offering a new functionality such that no other already-existing application can do so. If there are any critical non-functional requirements, you should explain how your design and implementation is going to comply with those requirements.

2. You must describe your preliminary design by including a UML class diagrams showing your design classes and their relationships. If your team is using any of existing libraries or technologies to implement the proposed features, please include specific descriptions of those libraries and technologies and specify the API which is to be used and integrated into your application.

Capability

You must include specific, detailed descriptions of how your team is capable of implementing the proposed application in 5-week time limit. Please include relevant prior internship/personal projects or class experiences to substantiate and justify your claims about each individual's capability and their estimated role in the team.
In your report, you must also specify each member's contribution on this part of the project.

Make sure your report has:

- **Evidence of novelty:** What purpose does your application serve? If related applications already exist, how is yours different? Are these changes useful or solve an important problem?

- **Documentation of UML diagrams:** Use case diagrams, Statechart diagrams and Class diagrams of the system.

- **UI Mock-ups:** you can use Sketch, Figma, InVision (or any other tool of your choice) to make UI Mock-ups to give us an idea of what the application will look like.

- **Report Length:** The report should be around **7-8 pages in length**, inclusive of diagrams. Having more pages will not result in better grades! In fact, conciseness and clarity in the report is appreciated.

Part A Checklist

- ❑ Form a team and update Sheets with your team information
- ❑ Finalize an idea for your application
- ❑ Set up a repository on GitHub and add your TA (
  amrutha: https://github.com/amrutha95 ,
  prateek: https://github.com/prateekmalhotra,
  mihir: https://github.com/Mihirmathur)
- ❑ Create mockup designs (e.g., screen snapshots)
- ❑ Write Part A Proposal Report
- ❑ Create slides and prepare for the presentation
- ❑ Submit Report & slides on CCLE (Only one per team)