CS 422/522
SOFTWARE METHODOLOGIES I

COURSE PROJECT

The practical component of this course consists of a software development project. In this project, you will practice the concepts learned in the lectures. You will have lots of fun, although it will not be easy. Let's start!

1. PROJECT DESCRIPTION

You may choose your team for the course project, but I may make changes at the end of the proposal stage (in agreement with team members) to make team sizes uniform. All groups must have four students. The main idea is to learn team organization, i.e., human resources, workload distribution, deadlines, collaboration, etc., as the project develops. It is not optional to be part of a team.

You have the first week to form your team, brainstorm, and propose ideas for the project topic, extension, limitations, resources, constraints, etc. You may know some of your peers; you have one week to make new acquaintances and integrate your team. During this week, the team members must gather (in-person or video meeting), propose, brainstorm, discuss, and develop a project idea.

Each team must have a team representative or leader who will be the primary communication channel for the team. For instance, the team leader will email the team name and members, upload submissions to Canvas, etc.

The team leader must submit the project title description, ConOps, main features, and team members and roles (see Section 3). This description must indicate the project's complexity. We will analyze it and provide feedback; your proposal may be ambitious or straightforward. Based on the feedback, the team decides whether or not to reduce or increase its feature set.

You may use the programming language(s) of your preference in accordance with the application field. You must document this decision.

2. DUE DATES

All project activities are already scheduled; the file `project-schedule.pdf` on Canvas contains the presentations, progress review meetings and their due dates. We may slightly adjust dates or times, according to the team's schedules.

3. PROJECT CHARACTERISTICS

The course project must include the following minimum set of features:

(a) State your ConOps (Concept of Operations), i.e., explicitly and clearly define what problem your project will solve.
(b) List all the project requirements (functional and non-functional).
(c) Start with a clear architectural design diagram – this diagram will be the basis of many of your design choices.
(d) Include front-end and back-end components.
(e) You must store data in a database – you may choose the database management system of your choice. The database must be in the third normal form (3NF) or higher. Your database design cannot have a straightforward structure. For example, you may have `books` and `customers` relations and a `borrows` relationship that indicates which customers are borrowing what books. See us if

you need help or want us to verify your database design. Create comprehensive documentation, including the database schema, data dictionary, and query examples.

(f) If the front end is a web app, make it responsive (it works for devices with different viewport geometries).
(g) Do your best on the front-end design – if you have artist or designer friends, have them help you.
(h) Provide a proper UX design – document your design using storyboards or any other UX description mechanism.
(i) You must use Git as a version control system and software development concurrent collaboration tool. Document your design decisions and configuration parameters.
(j) You must use a Project Management tool (e.g., ClickUp, Jira, or Trello).

4. THE PROJECT PROPOSAL

The project proposal must include:

(a) SRS Concept of Operations.
(b) SDS system overview
(c) Software architecture diagram (a general UML component diagram)
(d) List of technologies you will use.
(e) Project timeline and distribution of activities among team members.

Each of these items must be on a separate page of the Proposal. You are not committing to exactly what you write in the Proposal (projects evolve). Nonetheless, the Proposal must identify a clear initial idea of what your group intends to do in the project. All writing for this project must follow "Produce Good Writing" as discussed in the course syllabus.

5. YOUR REQUIREMENTS ANALYSIS MUST BE VALID AND SOUND

Build systems that address real human needs. Do not build systems that rely on your guesses regarding (a) human needs and (b) technology that might help address human needs. Do not make stuff up. Your Concept of Operations, Use Cases, and all other aspects of your SRS must be based on a correct understanding of real processes. For example, (a) interviews, (b) peer-reviewed research papers, and (c) newspaper articles from reputable sources (such as the New York Times or the Washington Post).

Assertions require support. There are few ideas that you can state as assertions without providing support. Support usually comes from empirical data (direct observation) or citing of the research literature. Note how the example Project's description includes a form of a requirements analysis and builds the case for the need for a Time Series Repository. Note how the project description develops an understanding of peoples' needs and the technology currently available to meet those needs. Studying articles from reputable peer-reviewed journals, conference papers, and technical documentation resulted in an understanding of the process. This study, plus conducting user interviews, is the approach you should take to develop your Project SRS Concept of Operations.

6. TECHNICAL REQUIREMENTS

This section presents a minimum set of technical requirements. Your projects should generate more than just these.

i. The delivered system must be complete. Even if you deliver a subset of the required functionality, the provided system must execute without major bugs. The graders do not have time to debug your software.

ii. Systems should use standard libraries as much as possible. Follow the guidelines to document your code and mention what each library provides.

iii. Installing and running the system. Provide the required instructions to install, execute, and use your system.

iv. If you are programming in Python, use virtual environments (pip or conda) and include a `requirements.txt` file in your submission so that graders can reproduce the exact conditions under which you developed the library.

v. If you develop a Web App, the customers (graders in this case) will not require installation, and you will not need to produce a requirements file or installation instructions. However, you will provide and maintain everything on the back end.

vi. If you produce a Mobile App, we need to test it. You are responsible for providing a testing environment.

7. TEAMS, MEETINGS, AND DELIVERY

You will be working in groups of four students. Use the team organization concepts to get organized, divide work, etc. At the end of the project, every student will turn in a peer evaluation form. Keep that in mind when accepting your part of the work.

In a real-life project, you need to meet with your customer or a designated stakeholders committee for the following reasons, among others:

i. Initial meeting – this event is where the customers specify their needs.

ii. Questions and concerns – developers may have questions about the system's processes or its environment (its ecosystem).

iii. Quality specifications –required by the customer or proposed by the developer.

iv. Specifications of standards – standards represent general knowledge of the application field.

v. Changes to specifications – procured by the customer. These better happen before you start any implementation or detailed design. You must sign a contract before the project begins.

vi. Progress reports – either previously agreed and scheduled, requested by the customer, or proposed by the developer team.

In this course project, you may have a client, or your team can play the client role. We can schedule meetings to discuss your interaction with your client.

Project delivery is the final presentation of the project to the customer. For more extensive projects, several deliveries may occur at the end of each planned project stage.

An initial presentation and progress report meetings will occur at the designated dates and times. The final delivery will occur during class (see `project-schedule.pdf` on Canvas). In this project, the final delivery will occur during lectures and in the form of a brief presentation, followed by questions and suggestions from peers and instructors.