CS180: Project and Lab Guidelines

Points: 50 points (out of 100 points)

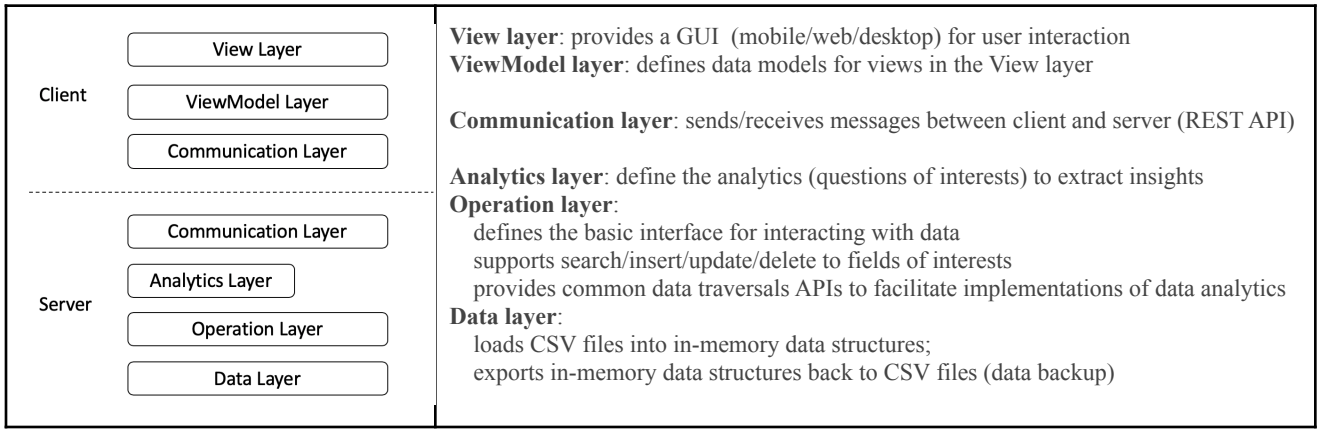
Description

This is a **team** project, with 4 to 5 students per team (**5 student teams are preferred**). The goal is to build an **in-memory data store** with a **layered client-server architecture** from scratch. Your data store will provide predefined data analytics of your choice, for gaining insights from an interesting data set. Initially, the data store will be populated with some public dataset in CSV format. The TA will disclose a list of project topics (public datasets), from which each team can pick a topic (dataset) and design data analytics around the topic.

Take the dataset "Uber Pickups in New York City" as an example. Each team needs to propose its own data analytics features for their dataset. For the Uber dataset, these might be some interesting analytics:

- *What are the peak hours for Uber pickups during a day?*
- *What are the top-3 popular streets that get most pickup requests?*
- *How do popular streets change overtime during the day? (This could help drivers find the right place to go at the right time)*

The project should follow a layered client-server architecture, as illustrated below:



Notes:

- Students **cannot** use any third-party libraries or frameworks in their project, with the exception of *JSON parsers / serializers* (for the REST API) and *visualization libraries* (for visualizing analytics results). In particular, CSV parsers and in-memory databases (e.g., the Pandas library for Python) are **disallowed**. If you are thinking of using a library and unsure if it is allowed, you must contact the instructor and get explicit permission before using it.

- Students are free to choose languages and platforms (mobile/web/desktop for the client) based on their skill sets.

- The client and server can be deployed **on the same machine**; the server does not need to be hosted separately. That said, using the REST API, deployment on separate machines should work.

Schedule

The project will follow the **Scrum** software process, with six week-long sprints (development cycles) to implement all its features. At the end of each sprint, the team needs to demo a piece of **working software** with some new feature(s). During development, the team needs to submit **test cases** and maintain an up-to-date **Task Board sheet**.

The "Artifacts to Submit" column lists documentation that must be completed by the *end* of the corresponding lab. For tools, we recommend keeping things simple and sticking to Word, Powerpoint, Google Docs / Slides, etc. The TA will show an example of how this documentation should look.

Lab	Activities	Artifacts to Submit
Before Lab 1 (Sep. 30)	Form a team of 4-5 members (otherwise, TA will assign a team in Lab 2); Select a project topic from a topic list disclosed by the TA;	Team name, members, & their emails; Project topic;
Lab 1 (Sep. 30)	Demo: have setup a channel (e.g., Slack) where all members have joined Planning meeting: - what data analytics / features to implement? - what languages and platform (mobile/web/desktop) will be used?	Platform for client and HTTP Server; Programming languages; Feature list (questions of interests)
Lab 2 (Oct. 7)	Sprint-1: Retrospective meeting: any updates since last meeting? any issues? Demo: client can connect to server (i.e., send/receive dumb messages) Planning meeting: - what are the fields of interests to search? - breakdown the goals into the actionable tasks - define task completeness criteria (test cases if needed)	Feature: - fields of interests to search GUI design and User test cases: - for testing the search feature Taskboard: - Done list of last sprint - ToDo task list for the next sprint
Lab 3 (Oct. 14)	Sprint-2: Retrospective meeting: what tasks have completed last week? any issues? Demo: search operation for multiple fields in the data Planning meeting: - what update/insert/delete features to implement? - breakdown the goals into the actionable tasks - define task completeness criteria (test cases if needed)	Feature: - fields to update/insert/delete GUI design and User test cases: - for testing update/insert/delete - for backup/import Taskboard: - Done list of last sprint - ToDo task list for the next sprint
Lab 4 (Oct. 21)	Sprint-3: Retrospective meeting: what tasks have completed last week? any issues? Demo: update/insert/delete features; data backup/import Planning meeting: - what analytics feature(s) to implement next week? - breakdown the goals into the actionable tasks - define task completeness criteria (test cases if needed)	Feature: - analytics to implement GUI design and User test cases: - for testing the analytics Taskboard: - Done list of last sprint - ToDo task list for the next sprint
Lab 5 (Oct. 28)	Sprint-4: Retrospective meeting: what tasks have completed last week? any issues? Demo: new analytics feature (s) Planning meeting: - what analytics feature(s) to implement next week? - breakdown the goals into the actionable tasks - define task completeness criteria (test cases if needed)	Feature: - analytics to implement GUI design and User test cases: - for testing the analytics Taskboard: - Done list of last sprint - ToDo task list for the next sprint
Lab 6 (Nov. 4)	Sprint-5: Retrospective meeting: what tasks have completed last week? any issues? Demo: new analytics feature (s) Planning meeting: - how to support incremental analytics? - breakdown the goals into the actionable tasks - define task completeness criteria (test cases if needed)	Feature: - incremental analytics User test cases: - for measuring the performance of incremental analytics Taskboard: - Done list of last sprint - ToDo task list for the next sprint
Lab 7 (Nov. 18)	Sprint-6: Retrospective meeting: what tasks have completed last week? any issues? Demo: performance improvements with incremental analytics	Taskboard: - Done list of last sprint

Demos (maximum of 3-4 mins using the pre-defined test cases) should be recorded offline and submitted every Wednesday by 11:59pm Pacific. Artifacts are due by the end of each lab.

Sprint Evaluation:

- **Sprint-1**: 4 points for demo and 2 points for artifacts; **Sprints 2-5**: 6 points for feature demo and 2 points for artifact submissions; **Sprint-6**: 6 points for demo and 1 point for artifacts. **Post-Sprint-6** cleanup, documentation, testing: 5 points.wh
- New feature(s) will be evaluated based on ***completeness*** and ***complexity***. Please contact TA if you are not sure about the complexity of your features is sufficient for one sprint;
- New feature(s) need to be demonstrated from the **users' perspective** (i.e., interacting only with the GUI);
- Features can be reevaluated (if needed) ONLY ONCE in the next sprint with a 15% late penalty.

After Sprint 6: You will be giving your final in-class demos several days after the final sprint evaluation (Lab 8 on Nov. 18). Your final demo should include some further polish / improvement over the version of your project demo'd in Lab 7. Further, the final 5 points of your project grade will be awarded based on a final review of your code and documentation done by the instructor. These points will be awarded based on the quality of your design, clarity of documentation, and presence of unit tests / use of continuous integration. All changes to your code must be complete and committed by **December 3 at 5pm.**

Grading

Individual Contribution (IMPORTANT)
To recognize individual contributions, the number of tasks performed by each team member will be summarized based on the Scrum sheets and converted into **percentages**. The final grade depends on the individual contribution of team member ***X*** , which can be calculated as follows:

$$IndividualContribution\ (X) = \frac{Num.\ of\ tasks\ completed\ by\ X}{Total\ num.\ of\ tasks}$$

The task statistics will be extracted from the Scrum spreadsheet.

The Github activities will also be used as a reference for individual contribution evaluation. An ideal contribution proportion would be *100% divided by team size* from each member, but *we prefer your actual contribution, rather than ideal*. Each student's contribution will be factored into that student's project score. **PLEASE NOTE:** it is highly important that Scrum sheets accurately reflect the work done by each team member and that the team agree on the information recorded there.

How Are Project Scores Computed? (IMPORTANT)
Each student's project score will be based on (1) that team's project score (*TeamScore*), which is the same for all team members, (2) the student's contribution to the project (*IndividualContribution*). The formula is as follows:

$$Multiplier = max\{0.8,\ min\{1.2,\ IndividualContribution \times TeamSize\}$$
$$IndividualScore = min\{45,\ TeamScore \times Multiplier\}$$

So, each individual's score will be within 80% and 120% of the team score, with a maximum of 50 points. **NOTE:** the instructor reserves the right to make exceptions to this formula in extreme cases, e.g., if one team member does vastly less work than other team members.

Frequently Asked Questions:

Q: What if I do not work well with my teammates?
A: You need to inform the TA & Instructor ASAP. After the development starts, we will not be able to accommodate any changes to team assignments. And you will be responsible for building good relationships with your teammates. Team skills are part of the training of the course project.

Q: What if one of team members is not contributing enough hence dragging the whole team down?
A: While the score computation formula is designed to account for such cases, early resolution is highly desirable