CS 520
Final project description

Final projects will be completed in teams of 3 or 4 students. Each team is responsible for a single project.

You should select a team and a project by **Tuesday, March 21, 2023, 11:59 PM (a little before midnight)**.

Your mid-point presentation will be due **Tuesday, April 18, 12:00 PM (noon)** and will take place **Tuesday, April 18, 4 PM**.

The final project presentation will be **Tuesday, May 16, 2023, 4:00 PM**. The final project deliverables will be due about a week later.

There are three topics for a final project[1] (each team will do one):

1. Research
2. Experimental evaluation
3. Development

The first section provides more details about each of the three topics for the project and the second section describes what needs to be included in the mid-point presentation for that project.

Topic selection

Research

The Mining Software Repositories conference runs an annual challenge in which they provide a dataset and ask you to answer research questions about the dataset. Read the description of the past and current years' datasets, research questions, and challenges here:
https://conf.researchr.org/track/msr-2022/msr-2022-mining-challenge#Call-for-Mining-Challenge-Papers
https://2021.msrconf.org/track/msr-2021-mining-challenge#Call-for-Mining-Challenge-Papers

Experimental evaluation

Replication study A replication study takes an existing software engineering research paper, replicates its experiments on the same data, and then extends the experiments to expanding that data set on which the experiments are run. For this project, we highly recommend selecting a paper with publicly available dataset and code to execute the experiments. The project involves a write up describing the process of replicating the experiments, deviations in the achieved results from the original ones reported in the paper, and lessons learned from applying the experiments to new data.

Here is a list of several papers that are good candidates for replication:

1. Are mutants a valid substitute for real faults in software testing?

   Paper: https://homes.cs.washington.edu/~mernst/pubs/mutation-effectiveness-fse2014.pdf

   Source code and dataset: https://github.com/rjust/defects4j

[1] In unusual cases, it is possible to convince the professor to do a self-defined project.

2. ExceLint: automatically finding spreadsheet formula errors

   Paper: https://dl.acm.org/doi/10.1145/3276518

   Source code: https://github.com/ExceLint/ExceLint-addin

   Dataset: https://dl.acm.org/do/10.1145/3276934/full/

3. SOSRepair: Expressive Semantic Search for Real-World Program Repair?

   Paper: https://people.cs.umass.edu/~brun/pubs/pubs/Afzal21tse.pdf

   Source code: https://github.com/squaresLab/SOSRepair

   Dataset: https://github.com/squaresLab/SOSRepair-Replication-Package

   Other candidate papers will be posted on Piazza.

Tool comparison This project would involve comparing/contrasting two different software development toolkits on the same benchmark(s) or dataset(s). The final deliverables are a presentation (and write up) of the lessons learned and experimental evaluation as well as any development artifacts (e.g., documentation, version control repository).

Here are some development toolkits. There are many others available.

- Design mockup tools, e.g., Marvel (https://marvelapp.com/), invision (https://www.invisionapp.com/)
- ML development tools, e.g., Neptune.ai (https://neptune.ai), Weights & Biases (https://wandb.ai/site)
- DevOps tools, e.g., https://www.atlassian.com/devops/devops-tools

Development

Your software development project must involve the following SE topics:

- Requirements
- Architecture
- Design
- Implementation
- Evaluation

The project needs to be under version control and written in an OO programming languages (e.g., C#, Java, javascript, python).

The Elevation-based navigation system (EleNA) is described below. This system applies many of the SE topics covered in this course.

Alternatively, you may propose your own development project. You must share your proposed project (in person or as a private Piazza post) with the instructor or a TA and they need to sign off on it.

Elevation-based navigation system (EleNA) Navigation systems optimize for the shortest or fastest route. However, they do not consider elevation gain. Let's say you are hiking or biking from one location to another. You may want to literally go the extra mile if that saves you a couple thousand feet in elevation gain. Likewise, you may want to maximize elevation gain if you are looking for an intense yet time-constrained workout.

The high-level goal of this project is to develop a software system that determines, given a start and an end location, a route that maximizes or minimizes elevation gain, while limiting the total distance between the two locations to x% of the shortest path.

Components:

Your software system will most likely have four main components:

1. Data model that represents the geodata.
2. A component that populates the data model, querying, e.g., OpenStreetMap.
3. The actual routing algorithm that performs the multi-objective optimization.
4. A component that outputs or renders the computed route.

While all components are necessary for a working prototype, you may choose to focus on some of them in greater detail. For example:

- If you focus on developing and experimenting with several routing algorithms, it is sufficient to have a simple interface for entering the start and end location and a simple output that represents the route.
- If you focus on a sophisticated UI with proper rendering of the computed route, it is sufficient to have a basic data model and routing algorithm.

Resources:

- The A$\star$ algorithm: `https://en.wikipedia.org/wiki/A*_search_algorithm`
- Dijkstra's algorithm: `https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm`
- OpenStreetMap wiki: `http://wiki.openstreetmap.org/wiki/Main_Page`
- The following paper, in particular Section 2, provides a very accessible introduction and overview of metaheuristic search algorithms:
  `https://pdfs.semanticscholar.org/9c83/752460cd1024985981d4acaa7bc85e15c0f7.pdf`