CSCE 361 course project details

CSCE 361 Capstone Project Choices

In the capstone project, you will demonstrate the good software engineering practices learned during the semester to develop a software project on a team of 3-4 students, using the Scrum development process that you will learn about in a few weeks.

Timeline

- Partnering requests due: March 31, 2021
- Project preferences due: March 31, 2021
- Team & Project Assignments will be completed by April 4, 2021

There will be 2 sprints, each 2 weeks long.

- Sprint 1: Ends 4/16/2021
- Sprint 2: Ends 4/30/2021
-
- Presentations/Demonstrations:
    - Presentation slidedeck and recording due: May 4, 2021 at 11:59 pm CDT
- Minutes Due: 11:59pm on the day of the corresponding meeting
- Peer Assessments Due: at end of each sprint
    - 11:59pm on April 16, 2021
    - 11:59pm on April 30, 2021
    -

Project Preferences

This 0-point assignment is your opportunity to identify your preferences for which project you would like to work on. You are not required to identify your preferences. If you do not identify your preferences, we will interpret that as you having no preference.

If one member of a pre-formed team identifies preferences, we will understand that to be the preferences of the full team.

Beyond partner requests, we will form teams of 3-4 students based on project preferences, based on performance in prior assignments, and based on time zones.

- We will strive to honor students' project preferences by placing them on a team assigned to their 1st- or 2nd-ranked preference. We absolutely do not guarantee that students will be working on their 1st- or 2nd-ranked project: the likelihood of being able to honor all such requests depends on the actual rank-ordered preferences of all students in the course.

- We will strive to balance the teams' collective mastery of the course material, so that if a student is weak in some particular area, other team members will be able to help them out.
- Except as noted below, we will assign at most one student per team whose prior performance indicates they are a habitual poor team member, either by not contributing an equitable share, or by failing to coordinate with other team members.
- If there are enough poor team members, we will create a team or teams consisting only of poor team members.

Projects

Questions to clarify a project's requirements should be posted to Canvas on the Project Discussion Board so that all teams assigned that project may benefit. I encourage students to discuss (on the Discussion Board) those questions. If students collaboratively arrive at a reasonable interpretation that does not contradict another requirement, we will likely accede to that interpretation. If students cannot arrive at a reasonable consensus, then we will provide the interpretation to use.

Online Store App:

E-commerce is the future of shopping. Your team is ready to jump on this innovative bandwagon and build an online store of whatever products are trending/useful for your target market.
The Online Store is a web application for users to shop for products. The user should be able to login to the website and be able to add products to their cart. They must be able to view products by category or by searching. The user’s cart must persist between sessions and the user must be able to provide shipping and payment details when checking out. The Online Store will have sales and the website must show the category or the products that the sale applies to. The back-end must provide product information to the front-end and handle business logic. The information and logic must be handled correctly and securely. The database will provide accurate information about products, sales, and users. The website must be able to provide this information about a product: name, product images, manufacturer information, description, dimensions and weight, a product rating, and a SKU. Sales must also have the following information: a start and end date, and what products or categories it applies to, as well as the dollar amount or percentage that applies to the sale. Users should not be able to see other users’ information or carts.

Boggle

Procrastination Pastimes wants you to develop a computer game of Boggle. The program should implement the rules of the pencil-and-paper version of Boggle as faithfully as is possible.

- https://www.hasbro.com/common/instruct/boggle.pdf

The sixteen dice have the following letters:

RIFOBX .IFEHEY .DENOWS.UTOKND

HMSRAO .LUPETS.ACITOA .YLGKUE

QuBMJOA.EHISPN.VETIGN .BALIYT

EZAVND .RALESC.UWILRG .PACEMD

The system shall be playable by a single player or by multiple players on networked computers. You may allow a limited or unlimited number of players; however, at least four players must be able to play against each other. You do not have to do network discovery but may instead use manually-entered address:port information similar to SocketChat (you will need to set up a separate socket between the host and each client).

When all connections have been made and the host instructs the system to start a game, each player will be shown a three-second countdown and then the dice shall be displayed. The dice can be in any permutation of the sixteen die positions, and each die can have any of its six faces showing. The timer shall then begin. You are not required to show a visual representation of this timer counting down.

Each player should then be able to enter as many words as they can until the timer runs out. If the user enters a word that is too short or is a duplicate of a previously-entered word, the program should silently allow the error to occur, so as not to distract the player.

After the timer has run out, the program should remove duplicates words, words that are too short, and words that do not appear in the corpus. The program shall then calculate and display each player's score and declare the winner of the round.

After each round, the host player may initiate another round. If the players play multiple rounds then after each round the system shall display not only each player's score for the round but also each player's running score.

The system may be implemented in text-mode or GUI-mode.

Card Games

Procrastination Pastimes (PP) wants you to develop a computerized card game that use the standard "French" 52-card deck (or multiple standard decks).[1]

The system shall offer at least one solitaire game and at least one game played with multiple players. The multi-player card game(s) must accommodate any number of players up to 6.

- If you implement a GUI, your multi-player mode may be multiple-player on a single computer, or it may be multiple-player across a network.
- If you do not implement a GUI, your multi-player mode must be multiple- player across a network.

Voting System

The city of Pacopolis wants you to develop an electronic voting system. While the mayoral race between Pat Mann and Dawn Keykong is the most visible election, there are other elections and issues that will also need to be decided. And, of course, the electronic voting system must be usable in future elections, too.

The system shall allow a voter to identify themselves through authentication, after which they shall be presented with the ballot. After making their selections, the voter shall be offered the opportunity to review and change their selections. Once submitted, the voter's choices shall be recorded. The system shall prohibit a voter from voting more than once in the same election. At the end of the voting day, the system shall determine the winner of each election and the outcome of each issue.

While it shall be possible for a voter to later view their recorded vote to confirm that it was recorded correctly, and it shall be possible for a third party to determine *whether* a particular voter voted, it shall be impossible for a third party to determine *how* any particular voter voted.

The system shall allow an unlimited number of voters to vote from their own computers or from a shared computer at a polling location.

- The system may be implemented in text-mode or with a GUI.

Package Tracker

Mo's Drones is a courier service using small unmanned aerial systems (SUAS) to deliver packages within the Lincoln and Omaha, Nebraska, areas. When a customer needs a package to be delivered, a SUAS is dispatched from a nearby depot to pick up the package.

Because the SUAS has a limited range, there are depots every ten miles along I-80 between Seward and the Missouri River. There are also depots in Lincoln at the intersection of O Street and 27th Street, at the intersection of O Street and 84th Street, and at the intersection of 84th Street and Nebraska Highway 2. When a SUAS with a package arrives at a depot, the package is handed off to another SUAS which will carry the package to the next depot or the destination (if the destination is within range).

A customer (both the sender and the receiver) should be able to observe the status of a delivery, to include the point of origin and the destination, when the SUAS was dispatched, when the SUAS picked up the package, when the package was handed off to another SUAS at each depot visited, and when the package was delivered. A customer should also be able to generate a delivery request, which will cause a SUAS to be dispatched automatically to pick up the package.

The Mo's Drones staff should be able to observe where each SUAS is, whether at a depot, between depots, en route to/from a customer, or at a customer's location. The staff should be able to observe which package is aboard which SUAS. Just as the customers can, the staff should be able to observe a package's status. While dispatching a SUAS to pick up a package is automatic, the staff should be able to dispatch an empty SUAS from one depot to another.

Current information about the SUAS & package locations & destinations must be recoverable after a power outage.

The system may be implemented in text-mode or GUI-mode.

Airline Crew Scheduler

Cornhusker Airways (CHA) is a small regional airline that flies out of Lincoln, Nebraska; Iowa City, Iowa; Evanston, Illinois; and West Lafayette, Indiana. CHA needs a crew scheduling system to keep track of which employees are on (or are scheduled to be on) each flight. The crew positions are:

- *Captain* - One of two cockpit positions. The captain must be a qualified pilot for the particular aircraft.
- *First Officer* - One of two cockpit positions. The first officer must be a qualified pilot or co-pilot for the particular aircraft.
- *Flight Attendant* - Crew member(s) who are responsible for the safety of passengers in the main cabin.

CHA operates two types of passenger aircraft. **GBR-10** aircraft can carry 45 passengers, and **NU-150** aircraft can carry 75 passengers. You may assume both types of aircraft have enough range to fly between any two airports that CHA services and, under ideal conditions, will average 500 mph when flying between airports. You do not need to keep track of the maintenance condition of the aircraft.

Each flight must have a flight number, an aircraft (designated by aircraft type and registration number), a captain, a first officer, plus one flight attendant for every fifty passengers (rounded up). A flight has an origin airport, a destination airport, a scheduled takeoff time, an estimated takeoff time, an actual takeoff time, a scheduled touchdown time, an estimated touchdown time, and an actual touchdown time. (For our purposes, a "time" combines the date and the time of date.)

Each airport must have a full standby crew for each type of aircraft, though the standby crew members may be assigned to flights individually rather than as an atomic crew.

Updates to the schedule must be maintained in an electronic log that can be searched by flight, crew member, airport, and/or date range. Each update will generate a unique update number. The update number format must include semantically-significant information that is interpretable by humans, such as including the date of the change and/or the date of the flight as part of the update number. The possible updates are omitted here for brevity and will be provided later.

Any time the schedule is updated, the software must check those constraints. If any of the constraints are violated, the software must alert the user to the violation(s).

Current information about past & future flights and crew assignments must be recoverable after a power outage.

The system may be implemented in text-mode or GUI-mode.

[1] A standard deck of cards consists of four suits (clubs(♣), diamonds(♢), hearts (♡) and spades (♠)), with each suit having an Ace, a King, a Queen, a Jack, and numbered cards 2-10. Some games may make use of two Joker cards. ↩