**CS3733-D21 Project Part B**

**Practice Sprint B**

**Requirements and Functional Model**

This project assignment consists of 3 main parts that should be done in parallel as much as possible.

1. Practice Sprint 1 where your team will create user stories for software **prototypes** in a sprint planning meeting and then will write the programs during the sprint.
2. Gathering requirements for the full Brigham & Women’s Hospital pathfinding application.
3. Creating a functional model for just the **first real development sprint (the minimal application)** that is to be implemented in Sprint 1.

This is a complex project assignment and you will want to use Trello, Jira, or Airtable to keep track of all of the things that need to get accomplished, what state they are in, and who is working on it.

Hospital maps and data can be found in Assignments -> Project -> Project Maps and Data.

**Agile Methodologies Preparation**

1. Everyone should have read the Scrum Reference Card (Module M 2-1) and Rubin chapter 2.
2. Scrum masters should also read Rubin chapters 10 and 19.
3. Product owners should also read Rubin chapters 6, 7 and 9.

**Part 1: Practice Sprint 1 (Software Development Prototypes)**

Everyone should be working to get the entire team ready for software development. Apply the Agile methodology and do a Scrum planning meeting where you will create Poker sized and prioritized user stories (just the user story cards are required although you may include the conversation and/or confirmation if you created them) using Trello, Jira, or Airtable of the following prototypes and tasks:

1. CSV files and test parking lot map image. You will edit the hospital map of floor 1 to create 10 parking spaces in the parking lot to the left of the Shapiro building. Note that the parking lot on the right of the map below the E and F elevators already has parking spaces drawn out. The undirected graph of the locations and paths will include the 10 parking spaces in the left parking lot, 15 parking spaces in the right parking lot (you can select which 15 to use), 75 Lobby entrance, and the Emergency room entrance. Treat the streets as sidewalks for now, and create the intermediate nodes to permit paths between the parking spots and the two entrances so that patients would not walk through walls. Do not do pathfinding inside the building as the paths will change substantially in a future project submission. Look at the attached Word file, CS3733-D21 Map Data Entry Instructions to see how the node and edges CSV files are created.
2. **Database developers**. Create a command line program named “Xdb.java” where **X** is your team letter. The program always runs by creating an embedded (client-server will be later) Apache Derby database with a nodes table and an edges table and populating them with the same data that is in the CS files above. The Java JDBC program will permit a user to access, modify, and display the tables in the database. In your program, use the username admin with a password of admin to connect to the database.
3. When your program is executed without any arguments, e.g.,  
   > java **X**db   
   Then the program should output a message with the following options, and then terminate:

1 – Node Information

2 – Update Node Coordinates

3 – Update Node Location Long Name

4 – Edge Information

5 – Exit Program

1. When the program is executed with an argument 1 as follows:  
   > java **X**db 1  
   The program displays the list of nodes along with their attributes
2. When the program is executed with an argument 2 as follows:  
   > java **X**db 2  
   The user enters the ID of the node and is then prompted for the values of a new x and y coordinates
3. When the program is executed with an argument 3 as follows:  
   > java **X**db 3  
   The user enters the ID of the node and is then prompted for the new long name for the node
4. When the program is executed with an argument 4 as follows:  
   > java **X**db 4  
   The program displays the list of edges along with their attributes
5. When the program is executed with an argument of 5, the program exits.
6. When you finish writing and debugging the program, generate the JAR file using Gradle . You may need the assistance of your team coach if you encounter problems.
7. **Algorithm specialists**. Write a text-based program that uses a depth-first search algorithm on the test parking lot map data. Then begin writing the A\* algorithm. You do not need to complete the A\* algorithm in project B but will submit your IntelliJ project.
8. **Everyone else!** Have one of the lead or assistant lead engineers create the initial JavaFX app, **X**app.java where **X** is your team letter, by mirror cloning   
     
   github.com/mjclements/CS3733-Starter-Code  
     
   Instructions on how to create a mirror clone, which will permit you to make your repository private (forking will not), is included in the starter code. This starter code includes example JavaFX code on how to best switch between windows. Everyone who is not working on the database or A\* should create their own service request JavaFX page with several UI components on it – labels, textboxes, checkboxes, dropdown lists, etc. Create a default page that lists each service request and where each person will include a button to switch to their own service request page. And on their own service request page, they should create a button that switches to the default page. Do not worry about making the other UI components on the page do anything in this prototype. This is a good time to get people working with GitHub. When you finish creating the application, generate the JAR file using Gradle. You may need the assistance of your team coach if you encounter problems.

**Part 2: Requirements Gathering for the Full Application**

Because gathering requirements will take longer than the length of our sprints – one week, we will incorporate an **architectural spike** into our process before we begin the weekly application development sprints. For this part of the architectural spike, your team will perform requirements gathering for the entire final application. Please do not survey or interview health care professionals this term because most of them are too busy dealing with the pandemic and vaccinations to be distracted from their work.

1. See the attached Project Description for the high-level requirements of the full application.
2. Also, see on our Canvas site, Assignments -> Project -> Project Maps and Data
3. Use Zoom, and Google Docs or Miro (miro.com – sticky notes with real time collaboration), to brainstorm online as a team to come up with a set of application features (refer to the PowerPoint slides on how to conduct a brainstorming session!). Do not forget to include enhancements to the pathfinding, map editor, and service request components. Given what is currently taking place, see if you can brainstorm some ideas that take into account the current COVID-19 pandemic.
4. Create a survey of potential patient/visitor pathfinding enhancements and send them to 35 potential users – fellow students, family, friends, etc. **Do not specify Brigham & Women’s Hospital. Instead, state that the survey is about features that they may want if they were visiting a hospital**. Do not survey map building features because locating people with this background will be difficult. Have a question at the beginning of the survey asking if the participant is under 18 years old. If so, the survey should immediately end and thank the participant.
5. Present your survey questions. You will submit your survey results and findings including tables/charts in the next project submission, Project Part C.
6. Include your interview protocol (see Requirements Gathering PowerPoint slides). By Project Part C, you will need to have interviewed individuals on what they might want to have in a hospital pathfinding and service request application and how they would want to use it. Make it clear to them that this will be a combination of a **desktop application and mobile app**. No one under 18 years is to be interviewed.

**IMPORTANT**: Because the surveys and interviews fall under human subjects research (although not an IRB), the data collected are to be anonymized. You should state this at the beginning of the survey and interview. Respondents must be 18 years old or over and residing in the United States. Although you may collect demographic information (and will probably need to), it should not be so precise that individual people can be determined from the anonymized data. Instead of asking for exact ages, you can give ranges of 5 years or 10 years. For example, 18-30, 31-40, 41-50, etc.

**Part 3:** **Functional Model for the Minimal Application**

Note: You will write the code for this minimal application in sprint 1 (April 14 to April 20)

Create a functional model as a team, rather than divvying up each part to individuals, for the following iteration 1 minimal application that

1. using the A\* algorithm, draws a path on the parking lot map of Brigham & Women’s Hospital from a starting location to a destination specified by the user. The starting and ending locations are the 25 parking spots on the map and the two entrances. Do not create the paths for inside the building. You do not need to draw a path across more than one map.
2. includes a floor map editor for managing (add, edit, and delete) nodes, edges, and locations on a floor map.
3. includes one working service request component. See Project Description for some possibilities. You may come up with your own service request component but must get my approval for it.

The functional model should include scenarios, epics, user story cards, use case diagrams (textual use cases are optional), and UI mockups that are created using Figma or Adobe XD. These user stories do not need to be sized yet. Storyboards are optional. However, include storyboards in your submission if you created them. Remember to ask for feedback from the quieter members of the team by giving them time to think before asking them for their input.

**Submission**

The project manager will submit the following in a single compressed zip file:

1. The Xdb JAR file and IntelliJ project
2. The A\* algorithm IntelliJ project along with your simple map. You do not need to submit the depth-first search pathfinding program.
3. Xapp JAR file
4. Test parking lot map and accompanying CSV files
5. A single Word (using the Project B upload button) organized with clear sections that include the following
6. Cover page including
   1. CS3733-D21 Prof. Wong
   2. Team name
   3. Your team coach
   4. In a table, list each person, their position, and their GitHub account name. Keep in mind that the roles over time should become more flexible like in actual Agile Scrum teams in the industry.
   5. GitHub link to your organization. Make sure repositories are viewable to me and to your team coach!
7. List all software, project, and communication tools that the team plans to use.
8. List of sized user stories for the 3 prototype programs and who worked on them.
9. Application Requirements Gathering. Results of your brainstorming. Present your interview protocol and list of survey questions.
10. The functional model for the minimal application. UI mockups are to be pasted as images into the document.
11. Mention what you did this past week as a team to promote a healthy and congenial team culture. Spend time to discuss as a team, giving everyone a chance to give their feedback on how the team is operating with respect to task assignments, meetings, communications, etc.
12. There is a deliverable salary of $30.00 WPI dollars to be distributed among the team members (do not assign fractions of a cent) which will go towards the individual portion of your project grade. Determine a base salary for everyone who is participating, typically around 85%. If everyone is working roughly equally hard then the base salary may be higher. The remainder is to be assigned anonymously by individuals to those who did outstanding work. Meet as a team where every person will say what they worked on and accomplished for the past week before salary deliberations take place as a group exercise. Please note that the salary is only a small fraction of your final individual project grade. If the entire team does well as a whole, then the range of individual grades may go from a medium A to a high A. Concentrate on contributing to the team, not as much on the salaries! Only if someone does nothing or almost nothing will they receive a salary of 0 or 50%.
13. Acknowledge significant contributions by your team members in a personal manner. Also, think about how to best motivate all of the team members without disenchanting anyone early in the project. Justify your **rationale and method** for the final salary distribution. List the team members and their salary in **alphabetical order by last name** using the following table format:

Last name, first name Salary Rationale

1. When you have completed a document that includes all of the above, submit it to the Project Part B link in Canvas.

Remember to have fun!