**Term Project Final Report Paper**

**COMP 370 - ON1**

Group Members:

1. Jannine Gemmell
2. Naomi Ugwuoke
3. Shruti Verma

**Link to the GitLab for the UFV Interactive Map code:** [**https://cisgitlab.ufv.ca/Naomi/COMP371-UFVMAP**](https://cisgitlab.ufv.ca/Naomi/COMP371-UFVMAP)

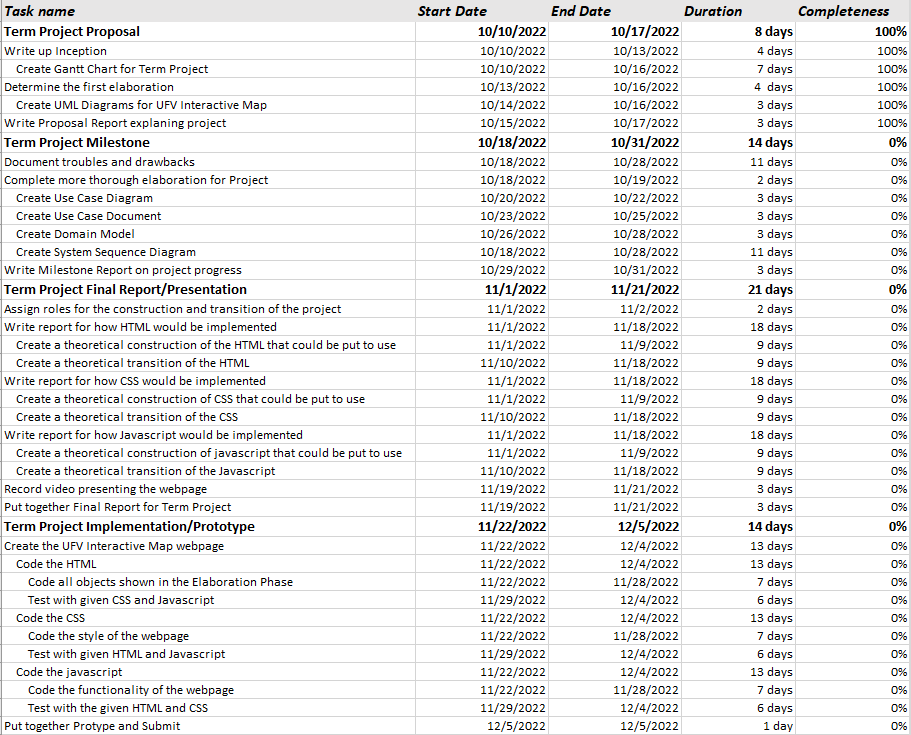
The original objective of this project was to create an interactive UFV map so that visitors and first-time students could easily navigate the Abbotsford campus.

The UFV Interactive Map has been designed to be a webpage similar to the UFV map currently found at <https://www.ufv.ca/maps/abbotsford-campus/>. Originally, the UFV Interactive Map had many functions proposed. Such functions included allowing the user to click on different areas of the map for more information and allowing the user to input directions into a search bar on the map to find specific rooms on campus.

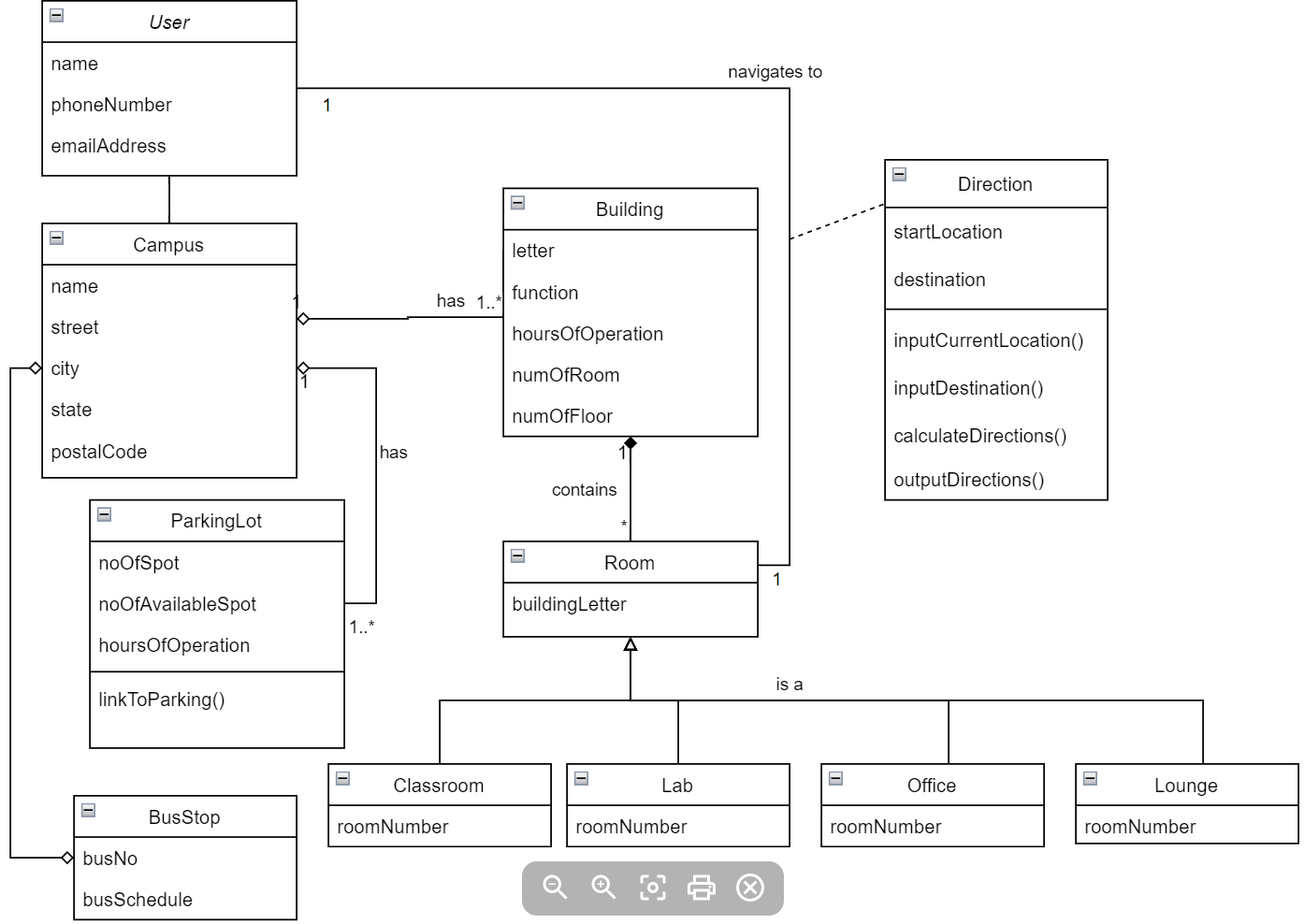
The proposed plan was to give the user access by clicking on buildings, parking lots and bus stops for information on each. The information to be displayed about buildings were hours of operation of the selected building, floor numbers, and room numbers and buildings with fast food places would display information on what fast food place is housed in that building, its hours of operation, and the menu as well (once selected by the user on the information screen). The information to be displayed about parking lots were how many stall numbers there are, the time of day parking is more likely available, the rate per hour that parking costs, and a link to the UFV webpage displaying parking information (<https://ufv.ca/parking/>). The information to be displayed about the UFV shuttle bus were hours of operation, what route the buses are on at that moment, and a link to the SUS Campus website (<https://ufvsus.ca/campus-shuttle>) or a link to the SUS Campus website (<https://ufvsus.ca/campus-shuttle>). The information to be displayed about the public bus stop were the bus schedule, as well as a link to the external website containing information on the bus and its routes (<https://transitfeeds.com/p/bc-transit/686/latest/stop/107190>).

The proposed plan also contained the idea of adding a search bar that allowed the user to input a starting point and an endpoint for locating different places on campus. The initial idea was that the start and end points could specify to the building letter and room the number of the building the user wished to go to, in order to ensure accuracy for directing the user. An example of such a query would be Start Point: Building B B121 and End Point: Building S 2103a. Once the user has inputted their start and end locations, the map would generate directions displaying a solid blue line (similar to that found on Google maps) that would lead the user through the campus to where they need to be. The initial plan was to create (or reuse) multiple maps of the campus to generate the line of direction, twisting and turning with the pathways and hallways on the campus.

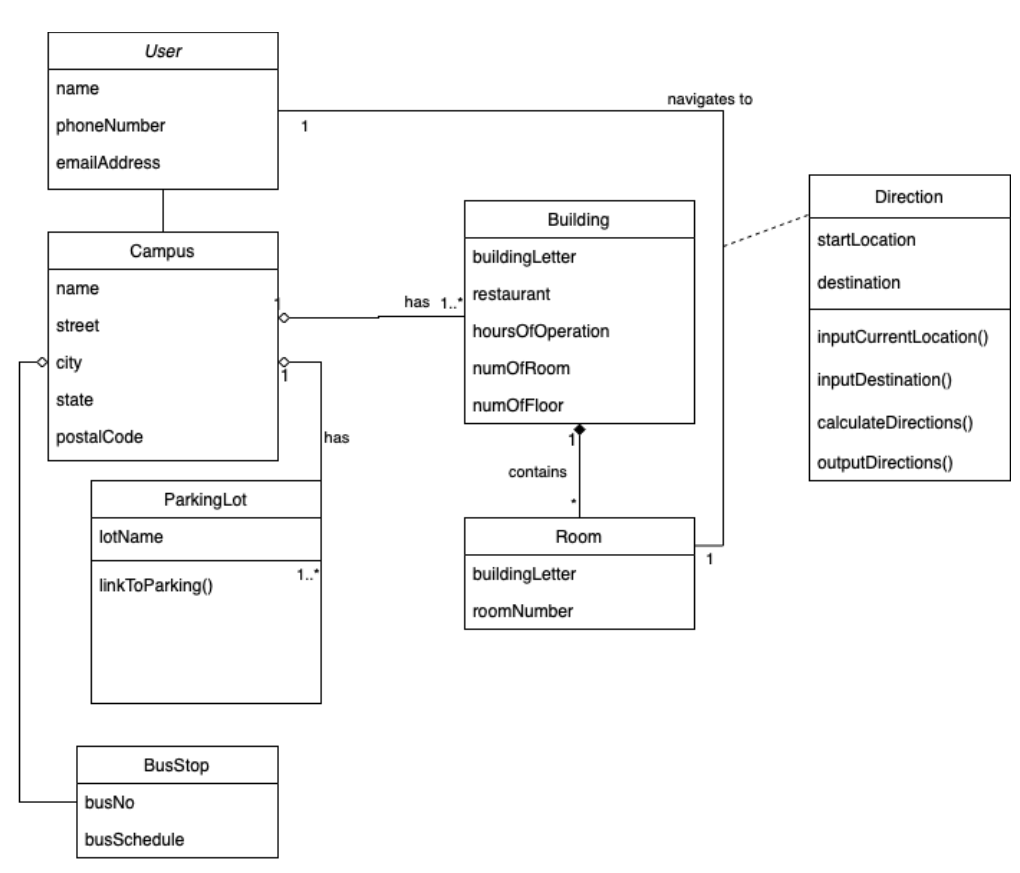
To avoid the use of the waterfall method and to keep to a Unified Process, a Gantt Chart was created during the inception of the project. According to the textbook, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development 3rd edition* (hereby referred to as the textbook), Page18, “[The] waterfall values [promote] big up-front speculative requirements and design steps before programming…the waterfall is strongly associated with the highest failure rates for software projects…”, and “The [Unified Process, or] UP, combines commonly accepted best practices, such as an iterative lifecycle and risk-driven development, into a cohesive and well-documented process description.” (the textbook, Page 18). “A Gantt chart is a Network Diagram. The key features contained in a Gantt chart are tasks, duration of the tasks, completeness, and the signature horizontal bar to indicate the duration length and the order of the task.” (COMP371 Lab #1, Page 1). The Gantt chart created for the project took into account the inception, elaboration, and construction of the project. Because this is a school project, the tasks were ordered by the deliverables and their due dates and the proposed plan for what should be done throughout the semester to stick to a Unified Process and get the project done on time.

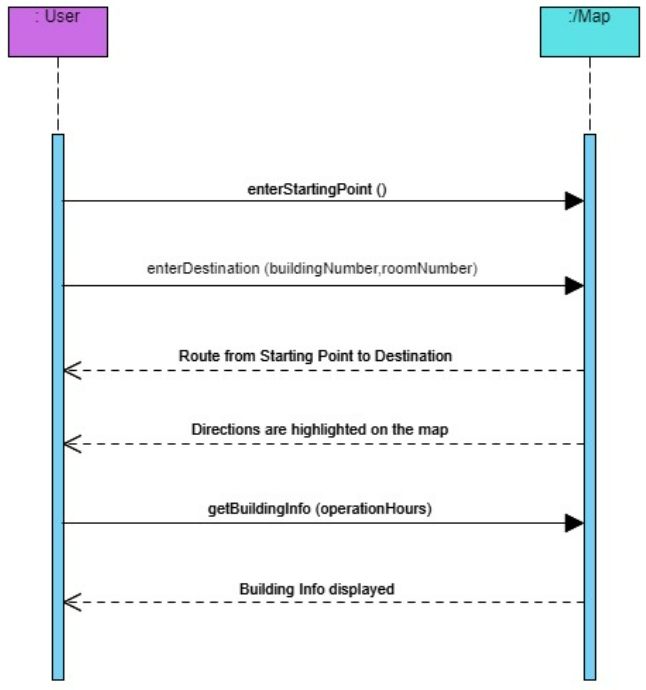


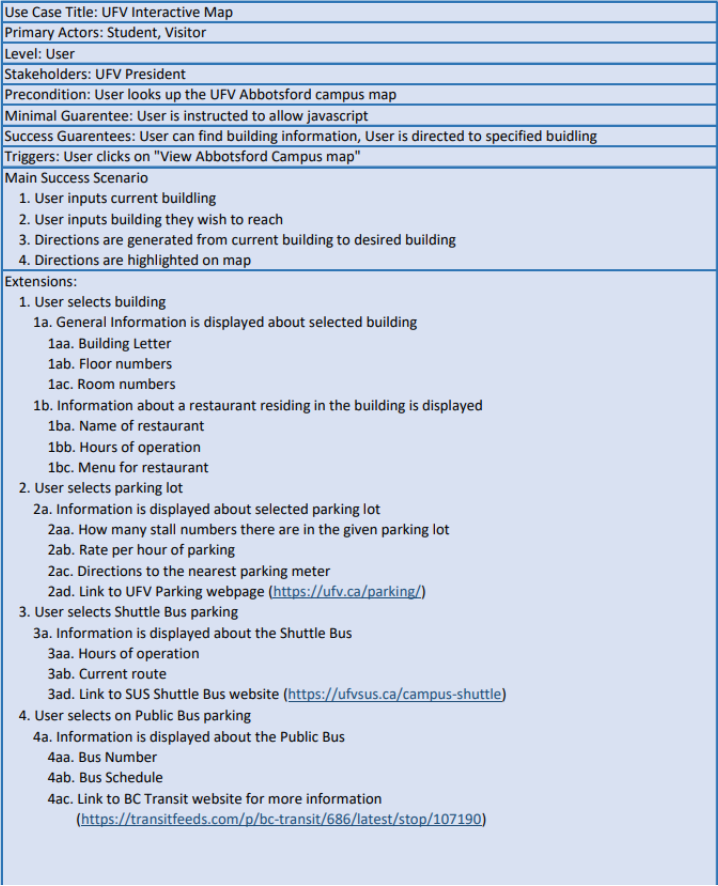
As part of the inception phase and to give us a visualization of the project, a Domain Model was also created to display the classes the project would contain along with the attributes proposed.

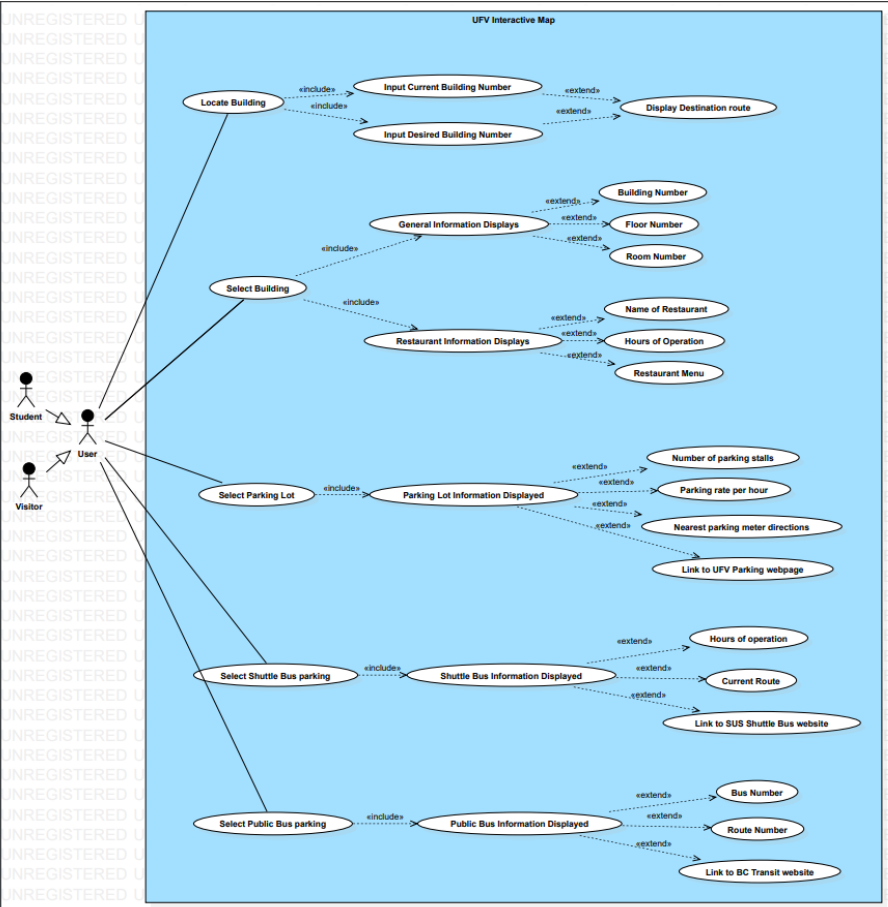


When doing the elaboration, UML diagrams were created for the UFV Interactive Map. Such UML Diagrams created were a domain model, a system sequence diagram, a use case document, and a use case diagram. According to the textbook, Page 131, “A domain model is the most important…model in OO analysis. It illustrates noteworthy concepts in a domain. ”, “the term “Domain Model” means a representation of real-situation conceptual classes, not of software objects.” (the textbook, Page 134), “A [system sequence diagram, or] SSD shows, for a particular course of events within a use case, the external actors that interact directly with the system, the system (as a black box), and the system events that the actors generate. … Time proceeds downward, and the ordering of events should follow their order in the scenario.” (the textbook, Page 175), “[A use case document is also known as fully dressed use cases.] All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees.”,“ Fully dressed use cases show more detail and are structured; they dig deeper” (the textbook, Page 67), and “[A use case diagram is also known as a UML Diagram.] the UML is the de facto standard diagramming notation for drawing or presenting pictures (with some text) related to software - primarily OO software.” (the textbook, Page 11). Each of us in the group created a UML diagram for the UFV Interactive Map.









Unlike the other three added UML diagrams, the Domain model was edited from the proposal phase to be updated so that the model was more precise since labelling each room as its own class is not a software engineering best practice when they’re all instances of the same class, ‘Room’. All UML diagrams created in the early phases of the elaboration phase were based on the proposal. This meant that everything proposed in inception was treated as objects and relationships between those objects were drafted in the four different UML diagrams created. Each UML diagram gave us different perspectives on how the project was shaping up. The Domain Model gave us a rough idea of how the objects in the project linked together and how they utilized one another to get the tasks done. The System Sequence Diagram displayed how the system would run given what a user could do with the system. The Use Case document documented all such possible use cases for the UFV Interactive Map with the main success scenario involving the user finding their way on campus. The Use Case diagram displays visually what the Use case document explained.

Once the UML diagrams had been created, rough coding to start implementing the project began. This rough coding involved using a screen capture of the UFV map displayed on UFVs website (<https://www.ufv.ca/media/assets/maps/2022-UFV-Campus-map-Abbotsford-with-legend.pdf>) to get an idea of the map we were going to use, and creating external HTML pages to display information about each building, parking lot, shuttle bus, and public bus.

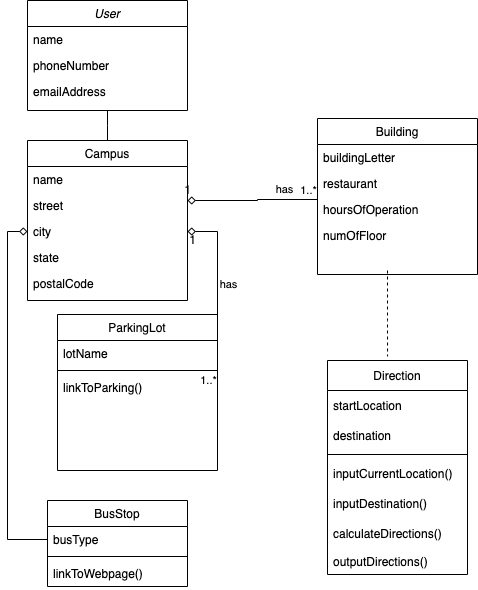
The plan was to have the information displayed as a pop-up tag from each of the buildings being selected and having the information further accessible by a scroll-down menu implemented in javascript, but creating external HTML pages was just to get a rough idea of how implementing information for the UFV Interactive Map was going to be like. As it turns out, there was a lot more information to add than initially anticipated, such as listing the number of rooms in each building. The number of rooms in each building is not listed on the UFV website, nor are the rooms listed by any numbering sequence, so each room had to be hand counted from the Abbotsford campus floor plans (<https://www.ufv.ca/operations/floor-plans/#d.en.938742>).

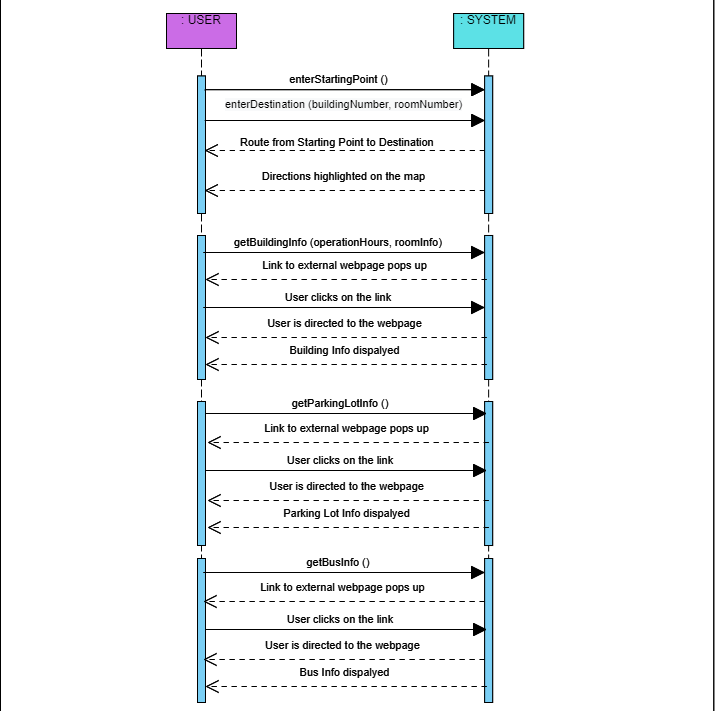
For the rough coding, allowing the user to click the buildings to display further information was implemented via an HTML image map element, found on W3Schools (<https://www.w3schools.com/html/html_images_imagemap.asp>). This image map element allows the programmer to specify a shape and when the user clicks on the shape specified an action could happen. I specified the shape to be a polygon and used Microsoft Paint to get the pixel measurements of the buildings. For the rough coding, image maps were only created for Building A and Building S. This turned out to be a very time-consuming task as well.

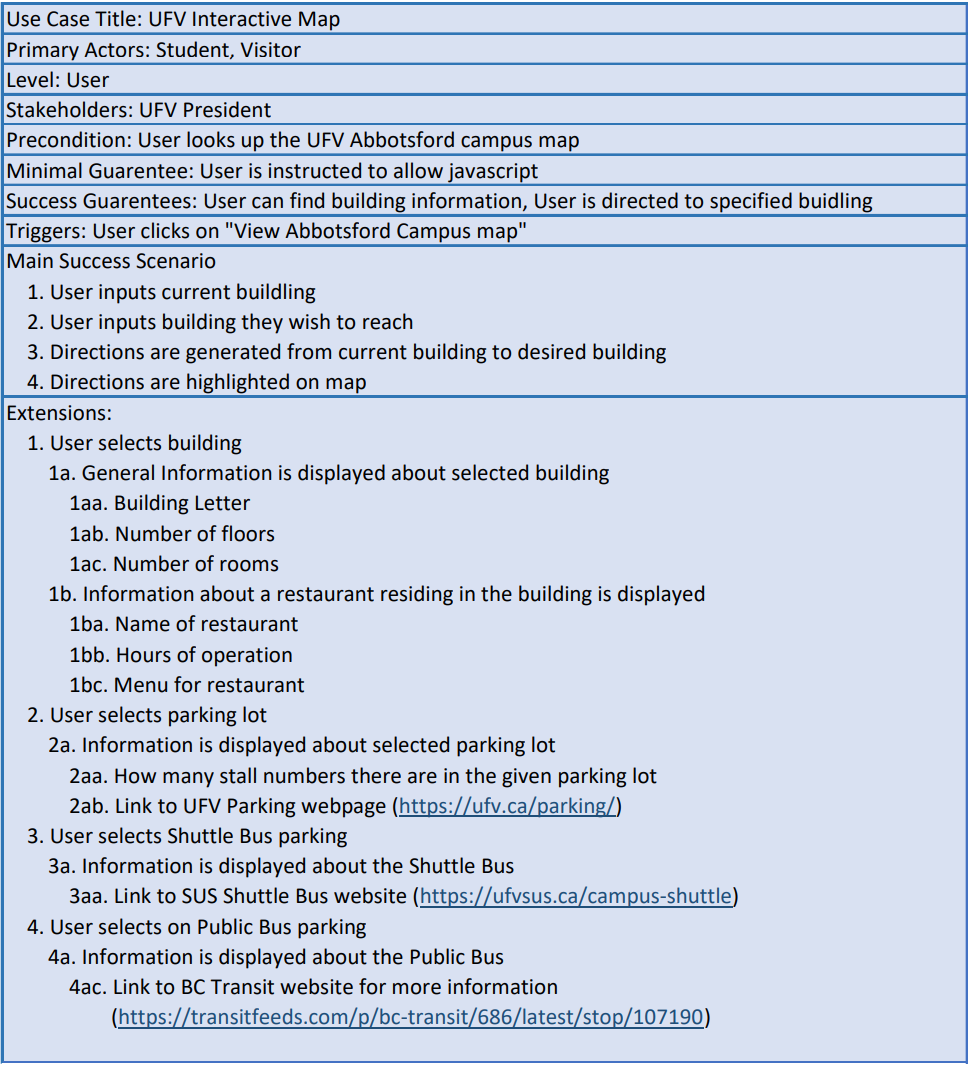
A better approach to creating the UFV Interactive Map was created by Naomi. Naomi created the map on the Google cloud platform. This platform allowed for the map to be styled using UFV colours and fonts and for the labels of each building to be cleared to simplify the look of the map. After designing the map, an API key and a map key were generated to allow the webpage to function. The first thing that was done to display the map was applying the Maps Javascript API and map keys to the javascript portion of the website. The default loaded view of the map was set to the coordinates of latitude: 49.02884701852467 and longitude: -122.28481317802066 with the map zoom set to 16.7. These adjustments allow the default view to focus on the UFV Abbotsford campus region. Next, markers for the building letters were created. The goal was to get the webpage to display a map that resembles the UFV map image as shown on UFV’s website (linked previously). The markers were created using PNG icons downloaded from <https://icons8.com>, a free-access icon provider. These markers were set at specific coordinates that correspond with each UFV building so that they can be clicked to display more information about the specific building. As per the original plan, they have been set up to show pop-up tags that will contain information about each of the buildings.

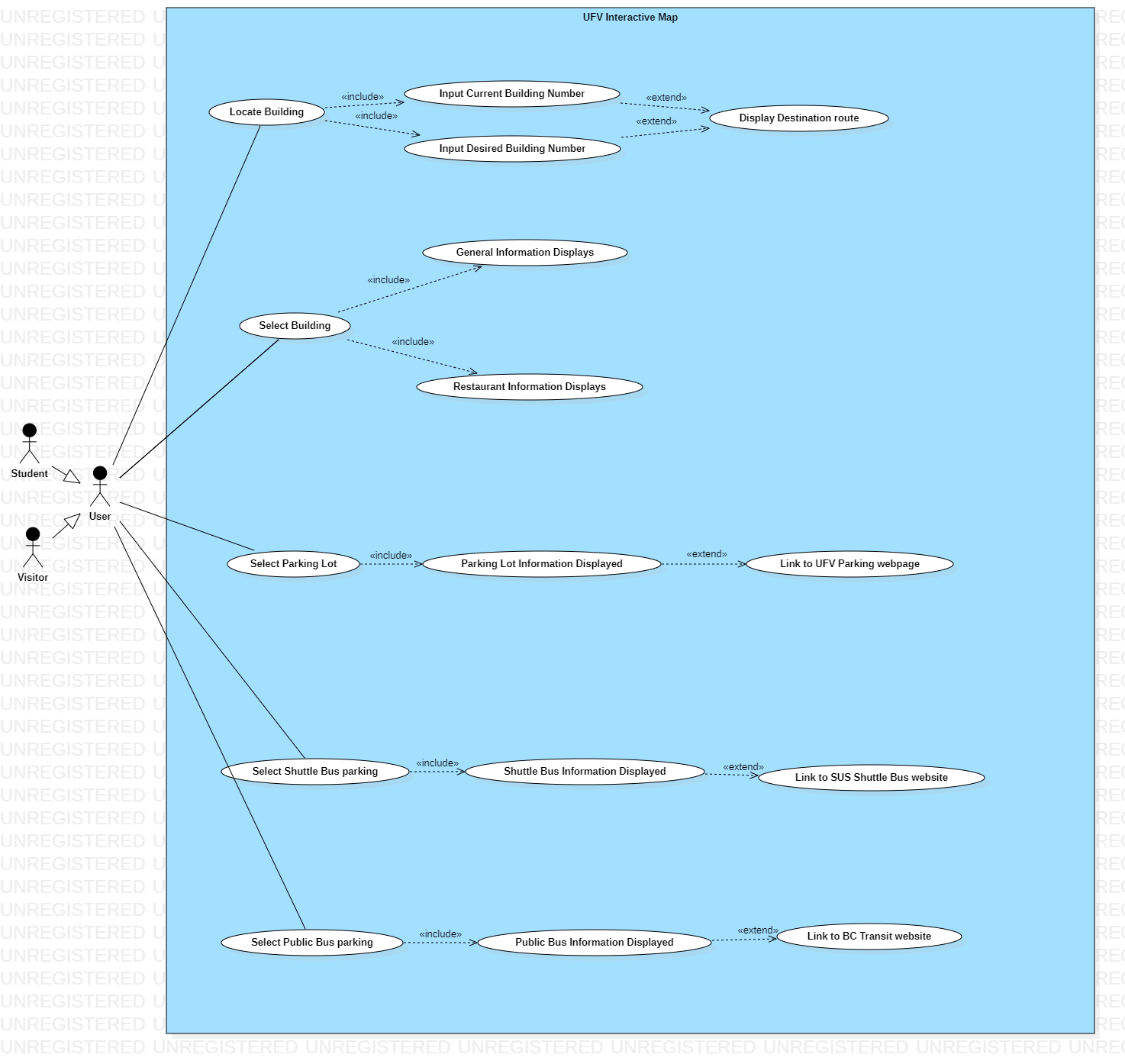
As the elaboration process proceeded many adjustments had to be made to the project's UML diagrams and code. A quick change made to the UML Diagrams is instead of “function”, “restaurant” is used to display information about any restaurants/food joints in a particular building. One of the problems encountered with implementing the map came from adding the Google cloud platform map to the webpage. Loading the webpage, the layout of UFV can be seen initially. However, the user can zoom out on the map allowing the user to visit other parts of the world map that is not the UFV Abbotsford Campus. Since this is a map of UFV, the map should only be able to display UFV. The original plan was to search for ways to lock the explorable area of the map to rectify this issue. However, we ended up letting the map load on the UFV Abbotsford campus. This was to support the “Where am I?” function which was added to help the user find their location on the map. Another problem that came up is that in order for the user to be able to get directions to a specific room in a specific building, multiple maps would be needed to display the internals of the buildings. This would lengthen the process of getting the project done and because we only have one semester to do this project, that was not an option. A fix to the issue was changing the UML diagrams to specify that only the buildings can be searched for and directed to on the map. In part of the compromise, after a user reaches the building, written instructions would display and this would guide the user on what floor to get to and what to do to find their room number. Overall, this meant eliminating the option for specifying the room numbers from the search query, which leads to a similar issue encountered. When the user inputs a destination and a start location on the Google cloud platform map, the user can enter any address. A proposed solution to fixing the issue was adding a function that restricts what the user can search by defining all the data that can be entered into the search query. Such data that the user would be restricted to entering is a building letter that is attached to each building by the corresponding latitude and longitude value. Using these values would then allow the Google maps API to calculate the route for the user.

As we approached the project's deadline, it became apparent the whole project was not going to be fully implemented. Changes were made to the UML diagrams to reflect what had been completed and what has yet to be implemented (given that we have more time to work on the project). Changes made to the Use Case Diagram were eliminating the extensions to the building information such as building number, floor number, and room number. This change was made because “General Information Displays” refers to the external HTML page which is a separate object of the webpage, whereas the specific info to be displayed is text within the “General Information Displays” object. Similar adjustments were made for the rest of the objects. Links to external web pages were kept as separate objects however because they refer to web pages beyond the scope of this project. Changes made to the Use Case Document were taking out info in the Extensions section that was no longer going to be implemented such as the “rate per hour of parking” and “current route” of the buses. Changes made to the System Sequence Diagram were taking out the “Building Info Displayed” action from the original system sequence diagram and creating a second system sequence diagram to display the interaction between the user and the system when the user selects different parts of the map. Changes made to the Domain model were taking away attributes of the classes that were no longer going to be implemented such as “busNo” and the Room class.









A prototype of the UFV Interactive Map has now been completed with Naomi’s map and is displayed in the video presentation.

Working with UML diagrams and using them to design an app has been a good experience for us. This project has exposed us to the Unified Process of coding development which are valuable skills we can use beyond the classroom. Creating the UFV Interactive Map introduced a new challenge of proposing an idea, elaborating on the idea, doing some rough coding to get the idea going, and taking a step back to re-elaborate on the idea once realizing certain challenges of the coding for the proposed idea. We have gained invaluable knowledge in doing this project and in this course, the knowledge that we can take on with us into our future jobs and careers.