

| IS 448 Project  Our Company:  Retriever Web Solutions |
| --- |
| | Russell Poon, Shane Severe, Jude Kim, Matthew Smith | 02/14/24 | IS 448 | | --- | --- | --- | |

# Team Charter

**Team purpose:** For this project, we are employed by UMBC and we are tasked to build a web application to solve a problem at UMBC. We will all be responsible for the final deliverable of the project, but for assigning individual responsibilities, we have volunteered for the project-specific roles.

**Team Roles:** Russell Poon (Lead Analyst)

Shane Severe (Project Lead)

Jude Kim (Test Lead)

Matthew Smith (Lead Programmer)

**Duration and Time Commitment:** We will work together either through discord video chat or with face-to-face meetings for about 2-4 hours a week. Depending on how the project is going, we might change the amount later on. We will also work on some tasks individually and bring our ideas and code to our meeting or through discord to get feedback. We are planning to have everything done by the specified deadlines. The last part of the project will be due on May 8th and we hope to have it done possibly a week early to test, redo, and practice the code and presentation.

**Team practices strategies and processes to accomplish work together, specifically:**

Our team will work together by using our collective knowledge, skills, and resources to provide the best product to serve UMBC’s needs. By using Discord to message, chat, and meet with each other through the app and in person, we will be able to work efficiently and in a timely manner to get the job done. Some strategies we plan to use include everyone coming up with ideas and preparing code outside of our meeting times so that it is ready to share, discuss, and implement.

We also plan to be ready for group discussion by implementing a little bit of code at a time and running it to see if it works. If it does, we will move onto the next block of code and if it doesn’t we know where we messed up and can fix it quickly. Depending on how our progress is going at first, we expect to meet 1-2 days a week for a total of 2-4 hours. We can always meet for longer than we anticipate and will probably be increasing the number of days and hours as we progress to more difficult code. Dividing the code at first will initially depend on the skill level of individual group members. Some may be farther along than others and will be writing the code, but will be getting ideas, guidance, and help throughout the project. As each team member gets better, more and more could be divided to ease the workload for others.

**Team Performance Assessment:** Some key areas will be communication, working well with the other team members, and working quickly and efficiently. We will measure our progress as each week progresses and what stage we’re at. The amount of code we have and if it meets the requirements for the week will greatly help us each week. Adjusting the number of days and hours we meet face-to-face could also be a measure of our progress. After coming up with what we will create, there are requirements for each deliverable. We will follow the requirements and then make sure our code runs smoothly. As each requirement is completed and after testing our code, we will meet to determine if our goals are met each week. The policy we expect to follow is simple: everyone puts their best foot forward and helps the team complete its goals each week. Whether you know little or know a lot, providing ideas and suggestions is crucial for success. eight eyes are better than two and someone might miss something that someone else will find and point out. Being able to work well together is key for a successful deliverable and project grade.

**Communication strategies:** We will be communicating through Discord using the messaging and video chat capabilities. We will also be meeting before, during, and after class when time permits in order to actively discuss and share ideas to make the best product possible.

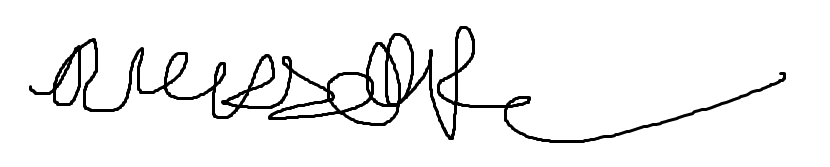
**Decision making and accountability:** Group will establish feasible project milestones that each group member will turn in by deadline. Decisions will be made between all team members and communicated either in our meetings or through messages in Discord. If there is any disparity in workload distribution, all members will discuss a more evenly distributed workload between members whether it's adding or subtracting some responsibilities between members. If there is any problem with a team member’s lack of following proper decorum and respect or the inability to meet deadlines, someone in the group should pick up the slack and finish the one part to get full points on the project deliverable. We will then have a discussion about why that member or members aren’t doing their share in the work and make a decision. If it persists, we might have to go to the professor to intervene or make a decision. The workload would shift to the other members in the group, adding a larger workload and responsibility so everyone else gets the full amount of points for the deliverable.

**Resources:** Textbook, Online resources like w3schools, PowerPoint presentations and resources provided by the professor on Blackboard.

Signature page:









# Project Proposal

**Proposal One:** Web Page for Studying Abroad Opportunities at UMBC

**Application Need:** Students who may want to study abroad may feel intimidated or confused by the process of applying to study abroad. Currently, the only way to learn about what options are available to a student is by attending a “Study Abroad 101” Session and by the information on the Education Abroad Office webpage (<https://studyabroad.umbc.edu/>). This provides an opportunity for innovation because a student could enter their major, their number of completed credits, and classes they are taking the next semester and the webpage could immediately let a student know the options they have available to them by displaying opportunities based on their entered criteria. This could reduce wait times for the “Study Abroad 101” session and aid the Education Abroad advisors because students would know what studying abroad opportunity they want. It also assists the advisors by informing them of what a student may need to complete before they can apply for the opportunity.

**Incorporation:**

**HTML:** HTML will provide the basis for how we add content to our website. This could include forms to gather user input, we may use tables for the purpose of displaying available studying abroad opportunities. Images may also be included to show what the country looks like that a student may be traveling abroad to.

**CSS:** CSS will be used to dictate how the content on our webpage will look. This may include ensuring that all text on the web page follows the same text font and color for consistency. We would include various selectors such as generics and class selectors.

**Javascript:** We will be using Javascript for our client-side programming. Javascript will assist us in gathering user input (students information in our case).

**PHP:** PHP will be our server side counterpart to Javascript.

**MySQL:** MySQL will store all the information used by students to register for the opportunity to study abroad. This will be used in part with Javascript, taking information input into Javascript and then stored in databases within MySQL

**Ajax:** Ajax will be used to help retrieve the data and help format the users page asynchronously, making it so that we, as the developers, do not need to refresh the page whenever an update is sent out from the web application.

**Use Case 1:** User inputs major, number of credits, upcoming classes, System takes user input and checks database for travel opportunities based on that criteria

**Use Case 2:** User inputs travel sponsor request information, System takes user input and checks database for travel sponsors of traveling abroad students

**Use Case 3:** User inputs country they would like to visit, System takes user input and shows all available travel opportunities based on that country

**Use Case 4:** User inputs Travel Abroad Request ID, System takes user input and assigns them to an Education Abroad Advisor

**Proposal Two:** Web Page for Study Room Reservations

**Application Need:** The web page for study room reservations as it currently exists is hard to navigate around. To get to where reservations can be made is somewhat difficult, as it is hidden under other banners and dropdown menus. By creating a new dedicated page for study room reservations, students would be able to reserve rooms easier and without much searching. Students/teachers would be able to reserve rooms for specific dates and times, check to see if the room is available, and make it easy to access and see all the features the rooms would provide. This would allow anyone wanting to reserve a room to pick the best time and room for their needs and make it easy to use without any confusion or mixups.

**Incorporation:**

**HTML:** HTML will provide the basis for how we add content to our website. This could include forms to gather information from the person making the reservation, images of the rooms they will be reserving to get a look at what they will be getting, and information about the room, what it includes, and how long you can reserve a room.

**CSS:** Allow for simplification of the menus and designs and will be primarily used for page/pages esthetic.

**Javascript:** We will be using Javascript for our client-side programming. Javascript in this proposal will be our mechanism for gathering user input such as building, occupancy, and time the study room is needed.

**PHP:** PHP will be our server side counterpart to Javascript.

**MySQL:** MySQL will keep a database of rooms currently being occupied, rooms that were occupied and that have freed up, or rooms that are already open. Time slots will be accounted for through the database.

**Ajax:** Due to HTML and CSS unable to be dynamically updated, AJAX will be a method in order to dynamically update the page on the availability of the study rooms.

**Use Case 1:**  User inputs time of reservation. System takes user input and returns available Study Rooms rooms across campus at that time.

**Use Case 2:** Students inputs building ID. System takes user input and returns available Study Rooms in that building

**Use Case 3:** Students inputs study room ID and the System returns whether that study room is reserved or in use at that time.

**Use Case 4:** Student inputs hardware needs of the study room (audio visual system, computer), System takes user input and returns the conference rooms that have those devices.

**Proposal Three:** Web Page for Interactive Campus Map

**Application Need:** New or current students might not know where certain buildings are for classes, where to get food, where they will be living, parking situations, etc. UMBC has a map that shows where everything would be on campus, but it takes a while to find what you want. Having an interactive campus map to highlight buildings, food, parking, other transportation locations, etc., to find specific locations or where all the food locations or parking locations are with a click of a button would help students be able to find the building they are looking for or all the options possible to decide where they would like to go.

**Incorporation:**

**HTML:** HTML will provide the basis for how we add content to our website. This could include adding buttons to display all the buildings and services when clicked, having a drop down menu to select certain buildings and parking locations, images to show what the building looks like and to have the interactive map so people can click on the map to show more information about the building or service.

**CSS:** CSS will be used to dictate how the content on our webpage will look. This would include ensuring that the text on the web page follows the same font and color for consistency and so that all users will be able to read the text on the screen. We would also make the web page fit the window and add colors to make it stand out by using class selectors.

**Javascript:** We will be using Javascript for our client-side programming. Javascript will assist us in having our action buttons when clicked open up extra information about the building or service and an image to show what it looks like.

**PHP:** PHP will be our server side counterpart to Javascript.

**MySQL:** MySQL will store all the information used by campus to identify landmarks, nearby parking options, and data about the building such as name and department within. This will be used in part with Javascript, taking information input into Javascript and then stored in databases within MySQL

**Ajax:**Due to HTML and CSS unable to be dynamically updated, AJAX will be a method in order to dynamically update the page on the links for each building.

**Use Case 1:** User inputs name of building they want to find. System takes user input and returns location on the map

**Use Case 2:** User inputs type of building they are looking for (dormitory,educational, commons). Systems takes user input and shows only the buildings that meet the type requested.

**Use Case 3:** User inputs parking lot number/name. System takes user input and returns location on the map.

**Use Case 4:** User inputs the class ID of a class being held that day. System takes the class ID and returns the building on the map where the class is being held.

# Initial Design

Link to our Google Slides for Initial Design: [Group 5 Initial Design](https://docs.google.com/presentation/d/1DX4DOpfuFBxOSdNl-3Bt7nFVnScEd4LGc94dlh0fY68/edit?usp=sharing)

# HTML and CSS Code

| **USE CASES** | **WHO OVERSAW EACH USE CASE** | **HTML/CSS FILES** |
| --- | --- | --- |
| Use Case 1 | Matthew Smith | main.html, res\_conformation.html, projectcss.css |
| Use Case 2 | Russell Poon | projectcss.css, reservation\_room.htlm |
| Use Case 3 | Jude Kim | reservation\_info.html |
| Use Case 4 | Shane Severe | cr\_confirmation.html, cr\_checkin.html,  cr\_checkout.hmtl |

# 

| **HTML/CSS DOCUMENTS** | **WHO WROTE THEM** |
| --- | --- |
| cr\_checkin.html | Shane Severe |
| cr\_checkout.html | Shane Severe |
| cr\_confirmation.html | Shane Severe |
| main.html | Matthew Smith |
| projectcss.css | Russell Poon/Matthew Smith |
| res\_conformation.html | Matthew Smith |
| reservation\_info.html | Jude Kim |
| reservation\_room.html | Russell Poon |

# Demo Slides

Link to our Google Slides for Initial Design: [Group 5 Demo Slides](https://docs.google.com/presentation/d/1AowySfw80tTx0sER3yBi2SS5RCNig9m91sixBskY97w/edit?usp=sharing)

# MySQL Code

| **USE CASES** | **WHO OVERSAW EACH USE CASE** | **DATABASE TABLES USED IN USE CASE** |
| --- | --- | --- |
| Use Case 1: Provides the user with an entry page with a list of available buildings and study rooms that they can then decide if they would like to make a future reservation or check into a room. | Matthew Smith | Building study\_room |
| Use Case 2: Users begin the study room reservation form. User selects the building, room, time of reservation, and equipment needed. | Russell Poon | Study\_room Study\_reservation Equipment building |
| Use Case 3: Continuing from UC2, User provides additional data to confirm the reservation such as their Student ID, # of individuals using the room, etc. | Jude Kim | Student, Study\_reservation study\_room |
| Use Case 4: Provides a Check In / Check Out system that can be used to identify that they are using a conference room and identify when they are done with it. | Shane Severe | Study\_room Study\_reservation building |

# PHP

| **USE CASES** | **WHO OVERSAW EACH USE CASE** | **PHP FILES USED IN USE CASE** |
| --- | --- | --- |
| Use Case 1: Provides the user with an entry page with a list of available buildings and study rooms that they can then decide if they would like to make a future reservation or check into a room. | Matthew Smith | get\_filtered\_rooms.php |
| Use Case 2: Users begin the study room reservation form. User selects the building, room, time of reservation, and equipment needed. | Russell Poon | Reservation\_room.php, reservation\_room\_getbuilding.php |
| Use Case 3: Continuing from UC2, User provides additional data to confirm the reservation such as their Student ID, # of individuals using the room, etc. | Jude Kim | Reservation\_room.php, reservation\_room\_getbuilding.php |
| Use Case 4: Provides a Check In / Check Out system that can be used to identify that they are using a conference room and identify when they are done with it. | Shane Severe | Reservation.php |

# 

| **PHP FILES** | **WHO WROTE THEM** |
| --- | --- |
| Reservation.php | Matthew Smith, Jude Kim |
| reservation\_room.php | Russell Poon, Shane Severe |
| reservation\_room\_getbuilding.php | Matthew Smith, Jude Kim |
| get\_filtered\_rooms.php | Russell Poon, Shane Severe |

# JavaScript/Ajax

| **USE CASES** | **WHO OVERSAW EACH USE CASE** | **JAVASCRIPT FILES USED IN USE CASE** |
| --- | --- | --- |
| Use Case 1: Provides the user with an entry page with a list of available buildings and study rooms that they can then decide if they would like to make a future reservation or check into a room. | Matthew Smith | N/A |
| Use Case 2: Users begin the study room reservation form. User selects the building, room, time of reservation, and equipment needed. | Russell Poon | reservation\_room.js |
| Use Case 3: Continuing from UC2, User provides additional data to confirm the reservation such as their Student ID, # of individuals using the room, etc. | Jude Kim | reservation\_room.js |
| Use Case 4: Provides a Check In / Check Out system that can be used to identify that they are using a conference room and identify when they are done with it. | Shane Severe | N/A |

| **JAVASCRIPT FILES** | **WHO WROTE THEM** |
| --- | --- |
| reservation\_room.js | All team members |

**Explanation:** reservation\_room.js is a javascript file that reservation.php can reference that adds an alert upon return. If the user’s input does not meet the input validation. an alert message will be shown, and the form submission will be prevented. Otherwise, if all fields pass validation, the form will be submitted. This ensures that invalid data doesn't reach the server. This maps to Use Case 2 & 3 because it checks the Student ID, number of individuals, and other fields to ensure valid data is being input.