**Project Abstract**

The BGC Equipment Tracker will be a web application used to help manage the BGC equipment inventory across three locations (Vancouver, Toronto, and Quebec). Our application is going to strive for user-end simplicity, allowing for the user to focus on their job rather than learning a complicated program. The BGC Equipment tracker will allow for Equipment Managers to scan the barcode of an item being lent out to a user, track which users have what equipment, as well as allowing them to view and request the equipment stored at the other locations. The Equipment Managers will also be able to look up specific users and see what they currently have signed out. When the user is done using their equipment, they can hand it back to the equipment manager, who will subsequently scan the equipment to sign it back into the inventory. The app will then automatically remove the equipment ID from the user's profile and update its status in the inventory. Equipment Managers will also be able to add, edit, and delete equipment entries. There will also be Admin users who have all the same abilities as the Equipment Managers, but they will also be able to create, edit, and delete user accounts.

**Customer**

Our customer is BGC Engineering. They are an international consulting firm that applies professional services in the applied earth sciences. Currently they’re using a generic app. Their current solution does not allow them to organise their equipment by the various locations (Vancouver, Toronto, and Quebec), and it does not have a way for users to request equipment that is stored at different locations. By having an in-app feature to request equipment, this can speed up workflow and users can see the status of their request (pending, approved, denied).

**Competitive Analysis**

The new BGC Equipment Tracker will be a superior version of their current app. It will have the same inventory management functionality, however we will improve on the shortcomings they have addressed in their old app. The BGC Tracker will allow for users with visual impairments to have an easier time navigating the BGC Tracker due to it having a simple layout with large text and bold buttons on it’s interface. We will also allow for the equipment database to be organised by different locations. Our option will also have other novel features such as Equipment Managers being able to generate ‘in-app’ requests for equipment at another location.

**User Story Roles:**

**Basic User:** Can sign out equipment, view inventory, and view equipment they signed out

**Equipment Manager:** Has all the same permissions as basic users. They can additionally manage (create, edit and delete) barcodes, manage (create, edit, and delete) equipment entries, signout /sign in equipment to other users, and can request equipment from other locations.

**Admin:** Has all the same permissions as Equipment Managers. Additionally they can also create, edit, and delete user profiles.

**User Stories:**

* **Iteration 1 - Story 1 - (Attempted login - FAILURE) :** A user attempts to login (figure 1). They input an incorrect email or password, and the app prompts them to re-enter the email and password. The user is unable to log in unless they input the correct details.
  + **Velocity points: 1**
* **Iteration 1 - Story 2 - (Basic User Login - SUCCESS) :** A basic user logs in and enters the homepage (figure 2). Here they have the option to look up equipment (implementation in progress).
  + **Velocity points: 1**
* **Iteration1 - Story 3 - (Equipment Manager Login - SUCCESS):** An Equipment Manager logs in and enters the homepage (figure 3). Here, they have the options to register equipment (implementation in progress), look up equipment and logout.
  + **Velocity points: 1**
* **Iteration 1 - Story 4 - (Admin Login - SUCCESS):** An admin user logs in and enters the homepage (figure 4). Here they have the same menu items as the Equipment Manager, except they can also register new users. They can specify the user’s name, email, password as well as the user role (basic user, equipment manager, or an admin).
  + **Velocity points: 1**
* **Iteration 1- Story 5 - (Admin Creates New User - SUCCESS):** After logging in the app, the admin selects “sign up user.” Here they enter the details for a new user, including the name of the user, the email, a password, and the user role. When they submit, the user can now use the new account to log in (figure 5).
  + **Velocity points: 2**
* **Iteration 1- Story 6- (Admin Creates New User- FAILURE):** While creating a new user, the admin accidentally leaves a section blank. The admin is now prompted to fill in that section and cannot submit unless this is done.
  + **Velocity points: 2**
* **Iteration 1 - Story 7 - (User is Inactive for 20 minutes):** A basic user logs into their account to view the equipment they have signed out under their name. After 20 minutes from when they log in, the user is logged out for security reasons.
  + **Velocity points: 1**
* **Iteration 1 - Story 8 - (User logout):** A logs into their account and enters the home screen. They then decide to logout. They click the logout button and then return to the sign in page.
  + **Velocity points: 1**
* **Iteration 2** **- Story 9**- **(Equipment Is added to the database- FAILURE)** - Equipment Manager receives new equipment and they wish to add it to the database. They select “add equipment” on the homescreen. Then they scan the barcode and input equipment details and press submit. If fields are not all filled out, the user will be prompted to fill them in. Now when users go to search for equipment, the newly added equipment also appears in the search.
  + **Velocity points: 2**
* **Iteration 2** **- Story 10**- **(Equipment Is added to the database- SUCCESS)** - Equipment Manager receives new equipment and they wish to add it to the database. They select “add equipment” on the homescreen. Then they scan the barcode and input equipment details and press submit. Now when users go to search for equipment, the newly added equipment also appears in the search.
  + **Velocity points: 2**
* **Iteration 2** **- Story 11**- **(Equipment Is deleted from the database)** - Equipment Manager wishes to delete a piece. They select “delete equipment” on the homescreen. They scan the barcode of the equipment they wish to delete. Now the equipment is removed from the database.
  + **Velocity points: 2**
* **Iteration 2 - Story 12- (User searches for equipment - FAILURE)** -User logs in and selects search for equipment on the homepage. The entire database is displayed initially. They then select the filters for their search (equipment name, location, id etc), and then they input their search. If nothing meets the search criteria, the list is displayed as empty.
  + **Velocity points: 1**
* **Iteration 2 - Story 13- (User searches for equipment - SUCCESS)** -User logs in and selects search for equipment on the homepage. The entire database is displayed initially. They then select the filters for their search (equipment name, location, id etc), and then they input their search in the textbox. The items meeting the criteria are displayed
  + **Velocity points: 1**
* **Iteration 2 - Story 14- (User searches for equipment by scanning - SUCCESS)** -User logs in and selects search for equipment on the homepage. The entire database is displayed initially, then they scan the equipment barcode, which then displays the equipment details on the screen.
  + **Velocity points: 2**
* **Iteration 2 - Story 15- (Admin Searches for Users)** -Admin logs in and selects search for users on the homepage. The entire database is displayed initially. They then select the filters for their search (name, email, etc. ), and then they input their search in the textbox. If there are users meeting that criteria, then they get displayed.
  + **Velocity points: 1**
* **Iteration 2 - Story 16- (User Refreshes the page)** -The user is on a page, and it is refreshed. The user does not have to log in again, as the token is reset and they return to the page they were previously on.
  + **Velocity points: 1**
* **Iteration 3 - Story 17- (Admin Searches for User - FAILURE)** -The admin logs in and selects on “Search Users.” From there they use the various filters to find the specific user they’re searching for. They can search by username, user id, etc. They will then see a list of

the user(s) that match the inputted search criteria. If a user does not meet that criteria then the list will be empty.

* + **Velocity points: 1**
* **Iteration 3 - Story 18- (Admin Searches for User - SUCCESS )** -The admin logs in and selects on “Search Users.” From there they use the various filters to find the specific user they’re searching for. They can search by username, user id, etc. They will then see a list of the user(s) that match the inputted search criteria
  + **Velocity points: 2**
* **Iteration 3 - Story 19- (Admin Edits User Information - FAILURE )** -The admin logs in and selects on “Search Users.” Initially all the users are displayed. From there, they use the various filters to find the specific user they’re searching for. They can search by username, user id, etc. From here they can select “edit” after finding the specific user. Then a popup appears where the Admin can edit the user's info. If the Admin leaves the text fields empty, they will be prompted to fill at least one field, and submissions won't be accepted until something is inputed.
  + **Velocity points: 3**
* **Iteration 3 - Story 20- (Admin Edits User Information- SUCCESS)** -The admin logs in and selects on “Search Users.” Initially all the users are displayed. From there, they use the various filters to find the specific user they’re searching for. They can search by username, user id, etc. From here they can select “edit” after finding the specific user. Then a popup appears where the Admin can edit the user's info. The admin then fills out the textbox they wish to change, and then press submit.
  + **Velocity points: 3**
* **Iteration 3 - Story 21- (Equipment Manager Edits Equipment Information)** - The Equipment Manager logs in and selects on “Search Equipment.” Initially all equipment is displayed. From there they use the various filters to find the specific item they’re searching for. They can search by item name, borrower, etc.
  + **Velocity points: 3**
* **Iteration 3 - Story 22 - (User Wants to View Their Equipment) -** The user is on the homepage and they select “Your Equipment.” Here they can see a list of all the equipment that is currently signed out under their name.
  + **Velocity points: 1**
* **Iteration 3 - Story 23 - (Equipment Manager wants to Delete Equipment) -** The Equipment manager is on the homepage. They then select “Search equipment.” From here they can search for specific equipment. There, they select the equipment they wish to delete and then delete its entry.
  + **Velocity points: 2**
* **Iteration 3 - Story 24 - (Equipment Manager wants to Export Data to an Excel spreadsheet) -** An equipment list is required for administrative purposes so the Equipment manager must generate it. The Equipment Manager logs in and enters the homepage. They then select “Search equipment.” From here they can search for specific equipment. They select search filters and then search for the equipment. The equipment specified appears on the list, and then the Equipment Manager presses “Export to Excel.” They now have an excel sheet displaying all the equipment that was being displayed with the filtered search.
  + **Velocity points: 3**
* **Iteration 3 - Story 25 - (User Requests Equipment) -** The user wishes to request a piece of equipment. They select “Request Equipment” on the homescreen. They then search the equipment using the appropriate filters. When they find the equipment they want, they select “request equipment” and an email is sent with the request details to the Equipment Managers.
  + **Velocity points: 2**
* **Iteration 3 - Story 26 - (Equipment Manager ACCEPTS Equipment Request) -** The Equipment Manager selects “Request Equipment” from the homepage. Here they see all the requests that are active. From here they can either ACCEPT or DENY the request. If they accept the request, the user is sent an email notifying them the request is ACCEPTED, and the status is then changed to CHECKED OUT. The user is now able to take the equipment.
  + **Velocity points: 2**
* **Iteration 3 - Story 27 - (Equipment Manager DENIES Equipment Request) -** The Equipment Manager selects “Request Equipment” from the homepage. Here they see all the requests that are active. From here they can either ACCEPT or DENY the request. If they deny the request, the user is sent an email saying the request is DENIED.
  + **Velocity points: 2**
* **Iteration 3 - Story 28 - (Email is Sent to Equipment Manager when Checking out Equipment) -** The User goes to the Equipment Manager to check out equipment. When the Equipment manager checks out the equipment, the Equipment Manager is then sent a confirmation email saying they have successfully signed out the equipment.
  + **Velocity points: 2**
* **Iteration 3 - Story 29 - (Email is Sent to User Requesting Equipment) -** After the User has requested equipment, they then wait to see if their request is approved/denied. When the decision is made, the User will receive an email when it’s APPROVED or DENIED
  + **Velocity points: 2**
* **Iteration 3 - Story 30 - (Equipment Manager Checks out Equipment to the User - FAILURE)** - When the Equipment Manager is checking out the equipment to the User, they scan the item barcode and fill in the user info as well as the “Checked Out” date and the “End Date.” If the Equipment Manager fills in an “End Date” that is before the “Checked out Date” they will be notified that the dates are incorrect and will not be able to submit the “check out request” until the dates are correct.
  + **Velocity points: 2**
* **Iteration 3 - Story 31 - (Equipment Manager Checks out Equipment to the User - SUCCESS)** - When the Equipment Manager is checking out the equipment to the User, they scan the item barcode and fill in the user info as well as the “Checked Out” date and the “End Date.” The information is successfully sent to the database, and the user can now take the equipment.
  + **Velocity points: 1**
* **Iteration 3 - Story 32 - (Email is Sent to User saying Equipment is Overdue)** - The User now has the equipment and is actively using it. They don’t realise that they have been using the equipment for so long that the “End Date” has now elapsed. The system detects the “End Date” has now elapsed and the status is changed to now being “OVERDUE.” When the status changes to “OVERDUE” the user is then sent an email notifying them of the new status, and they are reminded to return the equipment again.
  + **Velocity points: 2**
* **Iteration 3 - Story 33 - (User Returns Checked out Equipment)** - The User now has the equipment and is actively using it. They now realise that the equipment is almost at the “End Date.” They return to the Equipment Manager, and the Equipment manager opens the barcode scanner on the app. They scan the barcode, and the equipment status is changed from “Checked Out” to “Available.”
  + **Velocity points: 1**

**Velocity Discussion**

TOTAL VELOCITY POINTS: 56

AVERAGE VELOCITY POINTS PER ITERATION: approximately 19

The Average velocity of each iteration was 25, however, we noticed that the velocity of the first and second iterations are significantly lower (10, and 13) than the third iteration (33). This is due to the fact that in the first iteration many of the tools we were using were new to use, and we spent a lot of time doing research for what tools to use/how to use them.

A few examples of some of the things we had to learn for the first iteration were ReactJS, JSON web tokens, and the overall group dynamic of how to work together. By the time we got to the second iteration we got more comfortable coding using both ReactJS and JSON web tokens. And we were able to make more progress with those tools. This was also the iteration where we learned how to use the barcode scanning capabilities offered as a react component, as well as getting more experience managing the various user/equipment databases. Here we also learned how to use

bootstrap and this helped us standardise the styling for the pages in a simple and effective way. By the third iteration we felt comfortable using those previous tools, but this was the iteration in which we had to figure out how to generate emails and automate the sending of emails notifying users of things

such as confirmations for signing out equipment, accepting/denying equipment requests, etc. To do this we learned how to use Sendgrid and it allowed us to make our app more refined and allowed for a way to communicate various things to users by sending emails via the app.

A lot of time was consumed by conducting research into the various tools we used to make the app. By the time we reached later iterations, we felt more comfortable with the previous tools learned. Since fewer and fewer hours were spent conducting research, we were able to spend less time ‘figuring things out’ and more time actually creating the various features of the app. By the time we reached the final iteration, we could now use multiple APIs and skills to complete a single task. An example of this is the actual “Checking out” of the equipment. The scanning feature is available to both the Equipment Managers as well as the Admins, and this combines the skills we learned coding with ReactJS with the “barcode scanner” component, our active database management skills, as well as what we learned about using SendGrid for sending out emails. As a byproduct of learning more skills in later iterations, and features of later iterations requiring more tools/knowledge to create, there are not only more stories for Iteration 3, but also more complex stories which have more velocity points.