

CPSC 304 Project Cover Page

Milestone #: M3

Date: 11/1/23

Group Number: 81

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Bob Pham	44606424	z4j2v	bobpham@student.ubc.ca
Jason Wang	52783859	x5y6z	jason.wang014@gmail.com
Stevan Zhuang	57167090	m4y2u	stevan.zhuang@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Summary:

Our project is a Pokemon Go journal that aims to allow players to keep track of their activities and interactions with both in-game and outside of game elements. Our database models different in-game aspects that players may want to keep track of, in addition to actions that players may do in-game. As such, players can keep track of what they have accomplished within the game, as well as compare with other players.

Application Platform

For our project we are planning on using Oracle as our database management system and PHP as our backend (leaning towards Laravel or Symfony, but will go with what the course teaches). We will likely just use PHP with vanilla HTML and CSS, but are open to exploring TypeScript/React.js as our frontend framework. For version control, we'll use Git/Github.

Backend end

Everyone:

- A copy of the schema and screenshots that show what data is present in each relation after the SQL initialization script is run

Stevan:

- SQL Script:
 - Creates all the tables
- Aggregation with groupby
- Aggregation with having
- Input sanitization

Jason:

- Update Operation
- Selection
- Insert Operation
- Nested Aggregation with Group By
- Division

Bob:

- SQL Script:
 - Inserts all of the data
 - Delete
 - Projection
 - Join

Frontend

Stevan:

- Aggregation with groupby
- Aggregation with having
- Input sanitization

Jason:

- Update Operation
- Selection
- Insert Operation
- Nested Aggregation with Group By
- Division

Bob

- Initial Setup
- Inserts all of the data
- Delete
- Projection
- Join

Timeline:

Nov. 6th:

- (Stevan) Creates all the tables

Nov. 8th:

- (Bob) Inserts all of the data

Nov. 17th:

- Backend should be done (Soft deadline, we'll extend into the frontend timeline if required)
 - Delete
 - Projection
 - Join
 - Update Operation
 - Selection
 - Insert Operation
 - Nested Aggregation with Group By
 - Division
 - Aggregation with groupby
 - Aggregation with having
 - Input sanitization
- Discuss what the front end should look like
 - Get a standard css going

Nov. 29th:

- Front end + Back end should be done (Hard deadline)
 - Delete

- Projection
- Join
- Update Operation
- Selection
- Insert Operation
- Nested Aggregation with Group By
- Division
- Aggregation with groupby
- Aggregation with having
- Input sanitization