University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 01

Date: September 27th, 2022

Group Number: 92

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Katrina Huynh	33661737	k2e8u	Katrinahuynh2002@hotmail.com
Jaren Agujo	36384543	k6l0t	jaren.agujo@gmail.com
Shubhankar Vakde	48454482	m8c8l	sh2002vk@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

---

Q2. A brief project description answering these questions:
   a. What is the domain of the application? Describe it.
   b. What aspects of the domain are modeled by the database?
   a) The domain of the application will be an organization (named "JKS") that controls multiple branches of the amusement parks.
   b) Ticket sales are updated when a new passenger arrives, staff are tracked by ID and Name, the rides they operate, and the food stands they work at. Rides have restrictions on which passengers can ride it, etc. All these relations are modeled by the Database

Q3. Database specifications/functionality
   - Can be used by the parent organization of the amusement parks to monitor and analyze stats of how their holding are doing
   - The *Park-Manager* is an all-in-one solution for "Enjoyment Empire", a Vancouver-based Amusement Park company. It manages ticket sales, staff allotments, food stands, and separates them by different branch locations.

Q4. Application Platform
   a) We plan to use PHP for the project
   b) We also may use Java for the project, and Oracle for the database.

Q7.
   - We are making the assumption that the staff can work at various different rides/games/food stands.
   - We are assuming all customers have bought a ticket

Entity-Relationship Diagram containing the following elements:

- **Restrictions** (weak entity) with attributes: Height, Weight, Health Issues
- **Have** (identifying relationship) connecting Restrictions to Rides
- **Rides** entity with attributes: Name, ID
- **Operates** relationship connecting Rides to Ride Operator
- **IsAt** relationship connecting Rides to Branches
- **Staff** entity with attributes: Name, ID
- **ISA** (Disjoint + Total) hierarchy from Staff to: Ride Operator, Games operator, Food stand person
- **Games** entity with attributes: ID, Name
- **Operates** relationship connecting Games to Games operator
- **IsAt** relationship connecting Games to Branches
- **Works at** relationship connecting Food stand person to Food Stands
- **Food Stands** entity with attributes: ID, Name
- **Have** (identifying relationship) connecting Food Stands to Food
- **Food** (weak entity) with attribute: Name/Type
- **IsAt** relationship connecting Food Stands to Branches
- **Branches** entity with attributes: ID, Address, Name
- **Hosts** relationship connecting Branches to Events
- **Events** entity with attributes: Date, Name, ID
- **Customers** entity with attributes: ID, Points, Name
- **Have** relationship connecting Customers to Tickets
- **Tickets** entity with attributes: Date, ID
- **ISA** (Disjoint + Total) hierarchy from Tickets to: GA, Senior, Child