# Milestone 2

## CPSC 304 Project Cover Page

Milestone #: 2

Date: Oct 15 2024

Group Number: 10

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---|---|---|---|
| Parham Alijani | 76133834 | h6w2b | Parham360321@gmail.com |
| Mulder Liu | 57346538 | k8n6v | mulderl@student.ubc.ca |
| Rafferty Ang | 28491272 | u2s6p | raffertyasher@gmail.com |

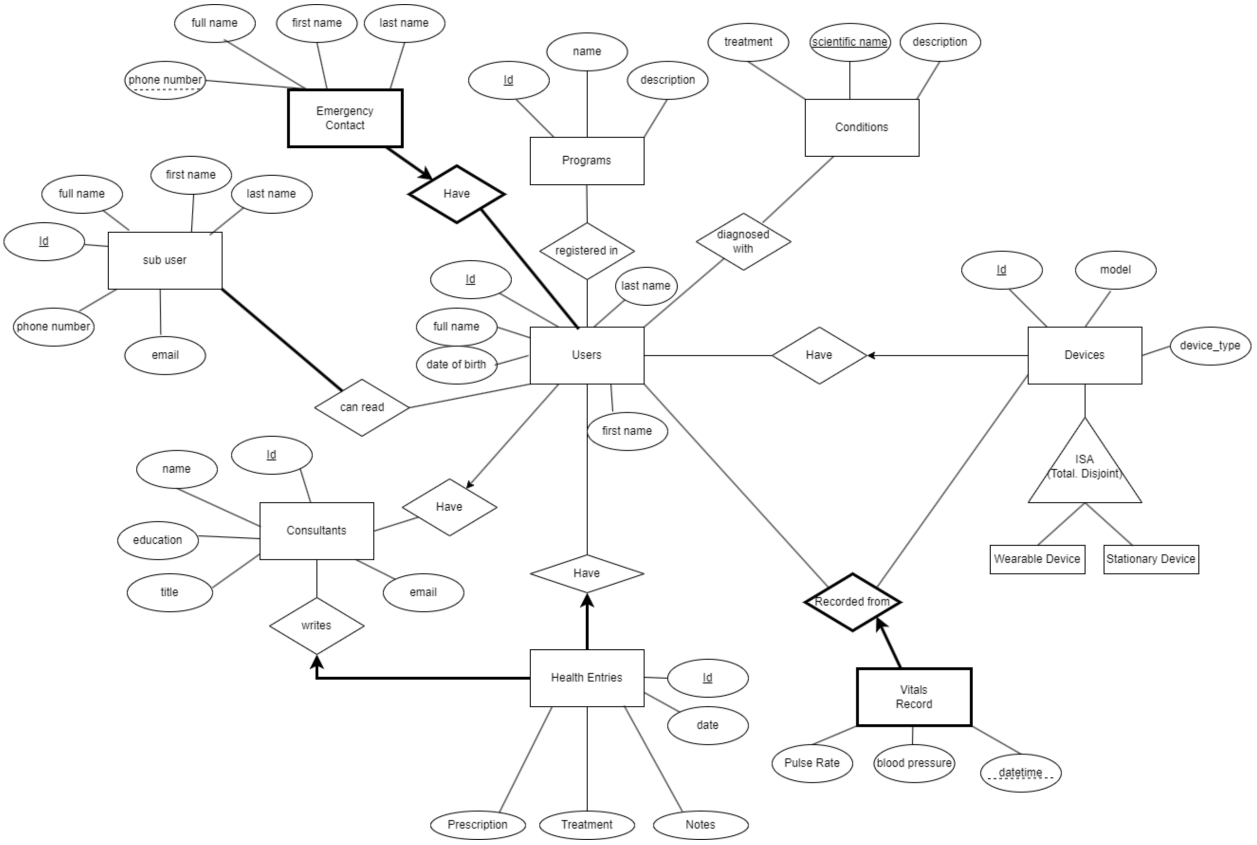
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.  (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

2)

The personal healthcare app helps users with health conditions, their families, and health consultants monitor and manage vital information. It enables users to track their health and follow programs, while health consultants can monitor and adjust patient care. A streamlined communication system ensures accurate, up-to-date exchanges between users and consultants.

3)



ER diagram changes:

- The ISA for conditions including diseases and injury was removed, and only condition is kept. This was done to simplify the design.
- Added ISA constraints to Device superclass
- Added attributes fullname, firstname, lastname to Users, sub user, emergency contact, for functional dependency
- Added attribute title to Consultants for functional dependency
- Added attribute device_type to device
- Added attribute email to consultant

4)

- Users(Id: int (PK, NOT NULL), fullname: varchar[101] (NOT NULL), firstname: varchar[50] (NOT NULL), lastname: varchar[50] (NOT NULL),dob: date, consultant_id: int (FK))

- Consultants(Id: int (PK, NOT NULL), name: varchar[101] (NOT NULL), education: varchar[50], title: varchar[50], email: varchar[50](CK, NOT NULL))
- EmergencyContact(phone_number: char[10] (PK, NOT NULL),user_id: int (FK, PK, NOT NULL), fullname: varchar[101] (NOT NULL), firstname: varchar[50] (NOT NULL), lastname: varchar[50] (NOT NULL),)
- HealthEntries(Id: int (PK, NOT NULL), entryDate: date (NOT NULL), prescription: varchar[100], treatment: varchar[100], notes: varchar[200], user_id: int (FK NOT NULL), consultant_id: int(FK NOT NULL))
- Devices(id: int (PK, NOT NULL), model: varchar[10] (not null), device_type: varchar[20] (not null), user_id: int (FK))
- WearableDevice(device_id: int (FK))
- StationaryDevice(device_id: int (FK))
- CanRead(user_id: int (PK, FK, NOT NULL), subuser_id: int(PK, FK, NOT NULL))
- SubUser(Id: int (PK, NOT NULL), fullname: varchar[101] (NOT NULL), firstname: varchar[50] (NOT NULL), lastname: varchar[50] (NOT NULL), phone number: char[10] (CK, NOT NULL), email: varchar[50] (CK, NOT NULL))
- RegisteredIn(program_id: int (PK, FK), user_id: int (PK,FK))
- Programs(Id: int (PK, NOT NULL), name: varchar[50] (NOT NULL), description: varchar[200])
- DiagnosedWith(condition_id: int (PK,FK), user_id: int (PK,FK))
- Conditions(scientific_name: varchar[50] (PK), description: varchar[200], treatment: varchar[100])
- User_VitalsRecord(device_id: int(PK FK NOT NULL), user_id: int(PK FK NOT NULL), DateTime: datetime (PK, NOT NULL), pulse rate: int, blood pressure: varchar[10])

5)

- User(Id, fullname, firstname, lastname, dob, consultant_id)
    - Id → name, dob, consultant_id
    - firstname, lastname→fullname

    - fullname → firstname, lastname
- Consultants(Id, name, education, title, email)
    - Id → name, education, title, email
    - education → title
    - email → id, name, education, title
- EmergencyContact(phone_number, user_id, fullname, firstname, lastname,)
    - user_id, phone_number → fullname, firstname, lastname
    - firstname, lastname→fullname
    - fullname → firstname, lastname
- HealthEntries(Id, date, prescription, treatment, notes, user_id, consultant_id)
    - Id → entryDate, prescription, treatment, notes, user_id, consultant_id
- Devices(id, model, user_id, device_type)
    - Id → model, user_id, device_type
    - model → device_type
- SubUser(Id, fullname, firstname, lastname, phone number, email)
    - Id → fullname, firstname, lastname, phone number, email
    - phone number → fullname, firstname, lastname, Id, email
    - email → fullname, firstname, lastname, phone number, Id
    - firstname, lastname→fullname
    - fullname → firstname, lastname
- Programs(Id, name, description)
    - Id → name, description
- Conditions(scientific_name, description, treatment)
    - scientific_name → description, treatment
- User_VitalsRecord(device_id, user_id, datetime, pulse rate, blood pressure)
    - device_id, user_id, datetime → pulse rate, blood pressure

6. Normalization into BCNF / 3NF

    - User
        - Closures
        - Id += {Id, fullname, firstname, lastname, dob, consultant_id}
        - firstname, lastname+= {firstname, lastname, fullname}
        - fullname +={fullname, firstname, lastname}
        - Decompose (firstname, lastname→fullname) to BCNF
        - Names(firstname(PK), lastname (PK), fullname(CK))
        - Users(Id (PK), firstname (FK), lastname (FK), dob, consultant_id(FK))
    - Consultants
        - Closures
        - Id += {Id, name, education, title, email}
        - email += {Id, name, education, title, email}
        - education += {education, title}
        - Decompose (education → title) to BCNF
        - Title(education (PK), title)
        - Consultants(Id (PK), name, education (FK), email (CK))
    - EmergencyContact
        - Closure
        - user_id, phone_number += {user_id, phone_number, fullname, firstname, lastname}
        - firstname, lastname+= {firstname, lastname, fullname}
        - fullname +={fullname, firstname, lastname}
        - Decompose (firstname, lastname→fullname) to BCNF
        - Names(firstname(PK), lastname (PK), fullname(CK))

    - EmergencyContact(user_id (PK, FK), phone_number (PK), firstname (FK), lastname(FK))
- HealthEntries
    - Closure
    - Id+ = {Id, entryDate, prescription, treatment, notes, user_id, consultant_id}
    - HealthEntries(Id (PK), date, prescription, treatment, notes, user_id (FK), consultant_id (FK))
- Devices
    - Closure
    - Id += {Id, model, user_id, device_type}
    - model += {model, device_type}
    - Decompose (model → device_type) to BCNF
    - Device_type(model(PK), device_type)
    - Devices(Id(PK), model(FK), user_id(FK))
- SubUser
    - Closure
    - Id += {Id, fullname, firstname, lastname, phone number, email}
    - phone number +={phone number, fullname, firstname, lastname, Id, emai}l
    - email +={email, fullname, firstname, lastname, phone number, Id}
    - firstname, lastname+= {firstname, lastname, fullname}
    - fullname +={fullname, firstname, lastname}
    - Decompose (firstname, lastname→fullname) to BCNF
    - Names(firstname(PK), lastname (PK), fullname(CK))
    - SubUser(Id(PK), firstname(FK), lastname(FK), phone number (CK), email(CK))

- Programs
    - Closure
    - Id+ = (Id, name, description)
    - Programs(Id (PK), name, description)
- Conditions
    - Closure
    - Scientific_name+ = (scientific_name, description, treatment)
    - Conditions(scientific_name(PK), description, treatment)
- User_Vitals
    - Closure
    - User_vitalsRecord+ = (device_id, user_id, datetime, pulse_rate, blood_pressure)
    - User-Vitals(device_id(PK) , user_id(PK) , datetime(PK) , pulse_rate, blood_pressure)
- List of every table
    - Names(firstname: varchar[50](PK), lastname: varchar[50] (PK), fullname(CK, UNIQUE): varchar[101])
    - Users(Id: int (PK), firstname: varchar[50](FK, NOT NULL), lastname: varchar[50](FK, NOT NULL), dob:date, consultant_id:int(FK))
    - Title(education:varchar[50](PK), title:varchar[50])
    - Consultants(Id :int (PK), name:varchar[50], education:varchar[50](FK, NOT NULL), email:varchar[50](CK))
    - EmergencyContact(user_id:int (PK, FK), phone_number: varchar[10] (PK), firstname: varchar[50](FK, NOT NULL), lastname:varchar[50] (FK, NOT NULL))
    - HealthEntries(Id: int (PK), date: date (NOT NULL), prescription: varchar[100], treatment: varchar[100], notes: varchar[200], user_id: int (FK NOT NULL), consultant_id: int(FK NOT NULL))

- Device_type(model:varchar[10](PK), device_type: varchar[20])
- Devices(Id(:intPK), model:varchar[10](FK), user_id:int(FK))
- WearableDevice(device_id: int (FK))
- StationaryDevice(device_id: int (FK))
- CanRead(user_id: int (PK, FK, NOT NULL), subuser_id: int(PK, FK, NOT NULL))
- SubUser(Id:int(PK), firstname:varchar[50](FK), lastname:varchar[50](FK), phone number:char[10](CK), email:varchar[50](CK))
- RegisteredIn(program_id: int (PK, FK), user_id: int (PK,FK))
- Programs(Id:int (PK), name:varchar[50], description:varchar[200])
- DiagnosedWith(condition_id: int (PK,FK), user_id: int (PK,FK))
- Conditions(scientific_name:varchar[50](PK), description:varchar[200], treatment:varchar[100])
- User-Vitals(device_id:int(PK) , user_id:int(PK) , datetime:datetime(PK) , pulse_rate:int, blood_pressure:string)

7. SQL DDL Create Table Statements

```
CREATE TABLE Users (
    Id INT PRIMARY KEY,
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50) NOT NULL,
    dob DATE,
    consultant_id INT,
    FOREIGN KEY (firstName, lastName) REFERENCES Names(firstN
    FOREIGN KEY consultant_id REFERENCES Consultants(Id)
);

CREATE TABLE Names (
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50) NOT NULL,
    fullName VARCHAR(101) UNIQUE NOT NULL,
    PRIMARY KEY (firstName, lastName)
);

CREATE TABLE Consultants (
    Id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    education VARCHAR(50),
    email VARCHAR(50) UNIQUE NOT NULL,
    FOREIGN KEY education REFERENCES Title(education)
);

CREATE TABLE Title (
    education VARCHAR(50) PRIMARY KEY,
    title VARCHAR(50),
);

CREATE TABLE EmergencyContact (
    user_id INT,
    phone_number CHAR(10),
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50) NOT NULL,
    PRIMARY KEY (user_id, phone_number),
    FOREIGN KEY (firstName, lastName) REFERENCES Names(firstN
    FOREIGN KEY user_id REFERENCES Users(Id)
);

CREATE TABLE HealthEntries (
    Id INT PRIMARY KEY,
    entryDate DATE NOT NULL,
    prescription: VARCHAR(100),
    treatment: VARCHAR(100),
    notes: VARCHAR(200),
    user_id: INT,
    consultant_Id: INT,
    FOREIGN KEY user_id REFERENCES Users(Id),
    FOREIGN KEY consultant_id REFERENCES Consultants(Id)
    )

CREATE TABLE Devices (
    Id INT PRIMARY KEY,
    model VARCHAR(10) NOT NULL,
    deviceType VARCHAR(20) NOT NULL,
    user_id INT,
    FOREIGN KEY (model, deviceType) REFERENCES DeviceType(mod
    FOREIGN KEY user_id REFERENCES Users(Id)
);

CREATE TABLE DeviceType (
    model VARCHAR(10) PRIMARY KEY,
    deviceType VARCHAR(20),
);

CREATE TABLE SubUser (
    Id INT PRIMARY KEY,
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50) NOT NULL,
    phone_number CHAR(10) UNIQUE NOT NULL,
    email VARCHAR(50) UNIQUE NOT NULL,
    FOREIGN KEY (firstName, lastName) REFERENCES Name(firstNa
);

CREATE TABLE Programs (
    Id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(200),
);

CREATE TABLE Conditions (
    scientific_name VARCHAR(50) PRIMARY KEY,
    description VARCHAR(200),
    treatment VARCHAR(100)
);
```

```
CREATE TABLE Users_Vitals (
    device_id INT,
    user_id INT,
    time_recorded DATETIME,
    pulse_rate INT,
    blood_pressure VARCHAR(10),
    PRIMARY KEY (device_id, user_id, time_recorded)
);
```

8. INSERT Statements

```
INSERT INTO Users (Id, firstName, lastName, dob, consultant_id)
(1, 'John', 'Doe', '1985-01-15', 1),
(2, 'Jerry', 'Smith', '1990-04-22', 2),
(3, 'LeBron', 'James', '1992-09-10', 3),
(4, 'Stephen', 'Curry', '1987-06-05', 4),
(5, 'Nikola', 'Jokic', '1995-12-01', 5);



INSERT INTO Names (firstName, lastName, fullName) VALUES
('John', 'Doe', 'John Doe'),
('Jerry', 'Smith', 'Jerry Smith'),
('LeBron', 'James', 'LeBron James'),
('Stephen', 'Curry', 'Stephen Curry').
('Nikola', 'Jokic', 'Nikola Jokic')

INSERT INTO Consultants (Id, name, education, email) VALUES
(1, 'Dr. Martha Stewart', 'PHD', 'martha.stewart@email.com'),
(2, 'Dr. Phil Wright', 'MD', 'phil.wright@email.com'),
(3, 'Dr. Michael Jordan', 'MSc', 'micheal.jordan@email.com'),
(4, 'Dr. Ron Artest', 'PhD', 'ron.artest@email.com'),
(5, 'Dr. Emma Black', 'PhD', 'emma.black@email.com');

INSERT INTO Title (education, title) VALUES
```

```
('PhD', 'Dr'),
('MD', 'Dr'),
('MSc', 'Mr/Mrs/Ms'),
('BSc', 'Mr/Mrs/Ms'),
('MBA', 'Mr/Mrs/Ms');

INSERT INTO EmergencyContact (user_id, phone_number, firstName,
(1, '1234567890', 'John', 'Doe'),
(2, '7787787789', 'Jerry', 'Smith'),
(3, '6047786034', 'LeBron', 'James'),
(4, '4676041234', 'Stephen', 'Curry'),
(5, '6056047780', 'Nikola', 'Jokic');

INSERT INTO HealthEntries (Id, entryDate, prescription, treatmen
(1, '2005-11-01', 'Ibuprofen', 'Sleep', 'Felt a headache after
(2, '2012-12-09', NULL, 'Rest', 'Felt nauseous while driving ho
(3, '2016-05-22', 'Ibuprofen', 'No fried food', 'Felt a stomach
(4, '2019-02-11', NULL, 'Sleep with a pillow', 'Woke up with a
(5, '2020-12-04', 'Ibuprofen', 'Exercise', 'Woke up with severe

INSERT INTO DeviceType (model, deviceType) VALUES
('AB12345', 'Heart Monitor'),
('BC34567', 'Heart Monitor 2'),
('CD45678', 'PPG'),
('DE56789', 'ECG'),
('EF67890', 'Fitness Tracker');

INSERT INTO Devices (Id, model, user_id) VALUES
(1, 'AB12345', 1),
(2, 'BC34567', 2),
(3, 'CD45678', 3),
(4, 'DE56789', 4),
(5, 'EF67890', 5);



INSERT INTO SubUser (Id, firstName, lastName, phone_number, ema
(1, 'Peter' 'Doe', '1233214567', 'peter.doe@email.com'),
(2, 'JR' 'Smith', '7781238901', 'jr.smith@email.com'),
(3, 'Savannah', 'James', '1234560000' 'savannah.james@email.com
(4, 'Seth' 'Curry', '01203405678', 'seth.curry@email.com'),
(5, 'Danilo', 'Jokic', '1122334455' 'dan.jokic@email.com');

INSERT INTO Programs (Id, name, description) VALUES
(1, 'Strength Training', 'Program designed for building muscle s
(2, 'Cardiovascular conditioning', 'Program focused on improving
(3, 'Yoga', 'Program focused on developing the skills of Yoga'),
(4, 'Weight Loss', 'Program designed to help with weight loss ch
(5, 'Endurance Training', 'Program designed to increase your sta

INSERT INTO Conditions (scientific_name, description, treatment)
('Diabetes', 'High blood sugar level', 'Insulin therapy'),
('Asthma', 'Inflammation of the airways', 'Inhalers'),
('Arthritis', 'Inflammation of the joints', 'physical therapy'),
('Hypertension', 'high blood pressure', 'medication and lifesty
('Obesity', 'Excess body fat', 'Diet, exercise');

INSERT INTO User_Vitals (device_id, user_id, time_recorded, puls
(1, 1, '2024-10-07', 72, '120/80'),
(2, 2, '2024-10-09', 75, '130/85'),
(3, 3, '2024-10-10', 69, '110/70'),
(4, 4, '2024-10-11', 65, '115/75'),
(5, 5, '2024-10-12', 80, '140/90');
```