University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 1, 2024

Group Number: 4

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Angela Felicia	51190304	r5e5u	angelafc@students.cs.ubc.ca
Qinying Li	10959491	o3e0n	carina00@students.cs.ubc.ca
Leo Shang	52598174	c3r3z	lshang04@students.cs.ubc.ca

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

University of British Columbia, Vancouver
Department of Computer Science

Summary
A brief summary about the project is present.

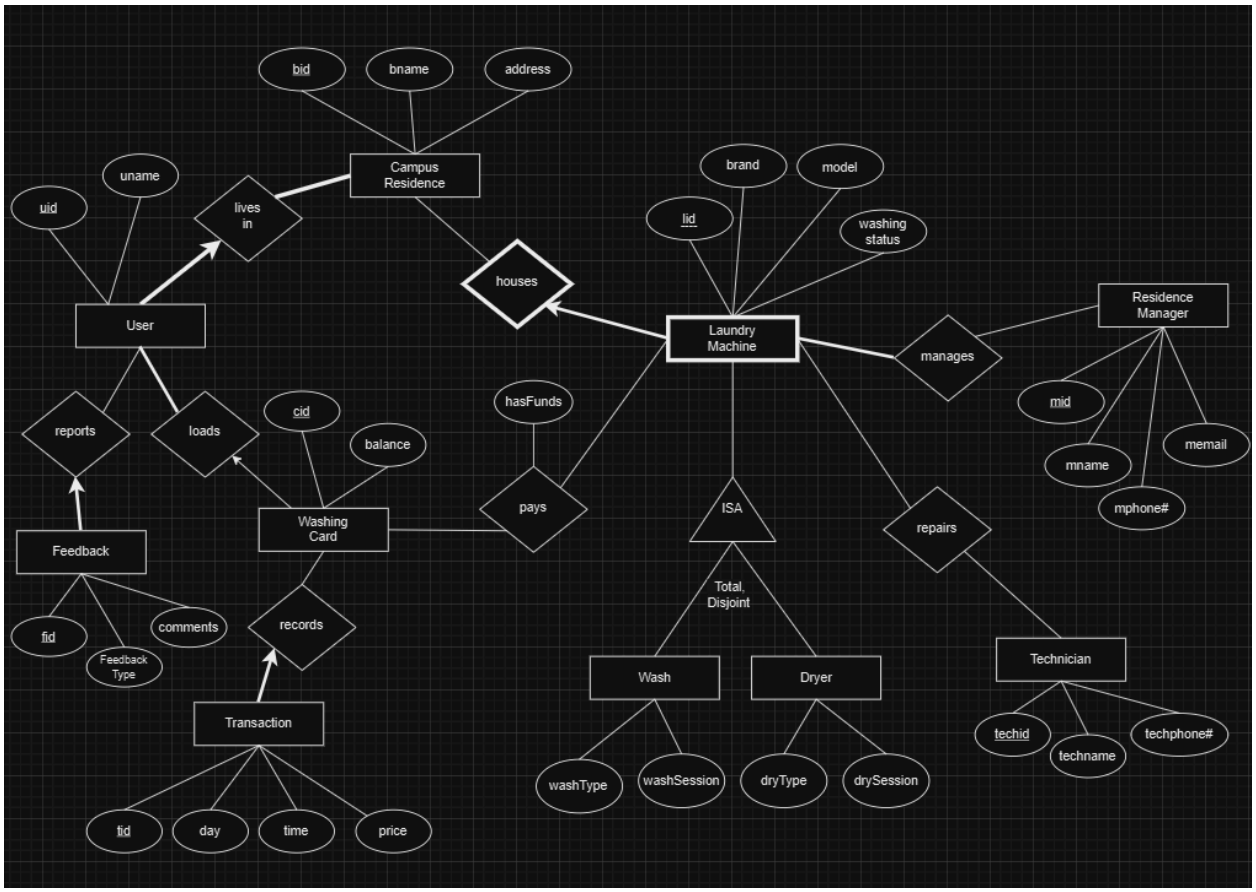
The field of the application is Campus Laundry Machines/Laundry Machine Management and covers various aspects of the laundry services in a university campus setting

The database will primarily track and display the real-time status and availability of laundry machines in campus residences. This allows users to easily check machine availability, plan their laundry sessions, and monitor the progress or completion status of their laundry, providing a more convenient and efficient laundry experience. The database also handles user management, enabling users to register and link their accounts to specific residences. Additionally, it tracks washing cards for machine payments, technician repairs to ensure accountability, and residence managers responsible for machine oversight. This model can be applied in real-life situations like university dorms, where students need easy access to laundry services, and maintenance teams need clear machine and repair tracking systems to ensure smooth operation.

ER Diagram
The ER diagram you are basing the relations on. If the ER diagram has changed from milestone 1, you must include a description of how the diagram has changed and why the changes were made. If you did not incorporate your TA's suggestions from milestone 1, please indicate why.

https://app.diagrams.net/#G1mB_rT2vNTpyHJYiDXJLDfOE5jZdDcDTL#%7B%22pageId%22%3A%22C5RBs43oDa-KdzZeNtuy%22%7D

University of British Columbia, Vancouver
Department of Computer Science



Relational Schema

List the table definition (e.g., Table1(attr1: domain1, attr2: domain2...). Specify all PKs, CKs, FKs, and other constraints that the tables have to maintain.

***NOTE**: primary keys are underlined and foreign keys are bolded*

Note: For primary keys, I did not put them under not null, unique, and candidate key since they are implicitly candidate keys, not null, and unique anyways.

UserLivesIn (uid: INT, **bid**: INT, uname: VARCHAR[20])

- **bid** references CampusResidence.
- uname default = “User”
- NOT NULL:
  - bid
  - uname

CampusResidence (bid: INT, bname: VARCHAR[20] , address: VARCHAR[50])

- CANDIDATE KEY:
  - bname
  - address
- UNIQUE:
  - bname
  - address
- NOT NULL:
  - bname
  - address

ResidenceLaundryMachine(**bid**: INT, lid: INT, brand: VARCHAR[15], model:VARCHAR[20], washing status: VARCHAR[10])

- **bid** references CampusResidence(bid)
- brand default = “Coinamatic”
- NOT NULL
  - washing status

NOTE: Wash and Dryer violated FDs so additional tables were introduced as seen in the normalization section

Wash(**bid**: INT, **lid**: INT, washType: VARCHAR[20], washSession: VARCHAR[20])

- bid, lid REFERENCES ResidenceLaundryMachine(bid, lid)
- NOT NULL:
  - washType
  - washSession

Dryer (**bid**: INT, **lid**: INT, dryType: VARCHAR[20], drySession: VARCHAR[20])

- bid, lid REFERENCES ResidenceLaundryMachine(bid, lid)
- NOT NULL:
  - dryType
  - drySession

LoadsWashingCard(cid: INT, **uid**: INT, balance: INT)

- balance default = 0
- uid references UserLivesIn

NOT NULL:

- balance
- uid

RecordsTransaction(tid: INT, **cid**: INT, day: DATE, time: Char[10], price: INT)

- cid references WashingCard(cid)
- NOT NULL
    - cid
    - day
    - time
    - price

ResidenceManager (mid: INT, mname: VARCHAR[20], mphone: VARCHAR[15], memail: VARCHAR[30])

NOT NULL:

- mname
- memail
- mphone

Candidate Key

- memail
- mphone

Unique

- memail
- mphone

Manages(**bid**: INT, **lid**: INT, **mid**: INT)

- bid, lid references ResidenceLaundryMachine(bid,lid)
- mid references ResidenceManager(mid)

Technician (techid: INT, techname: VARCHAR[20], techphone: VARCHAR[15])

- techphone DEFAULT="250-807-9000"
- NOT NULL:
    - techname
    - techphone

ReportsFeedback (fid: INT, **uid**: INT, feedbackType: VARCHAR[50], comments: VARCHAR[50])

- UID references UserLivesIn(uid)
- NOT NULL
    - uid
    - comments

Repairs (**bid**: INT**, lid:** INT**, techid:** INT)

- bid, lid references ResidenceLaundryMachine(bid,lid)
- techid references Technician(techid)

---

pays(**cid**: INT, **bid:** INT, **lid**: INT, hasFunds: boolean)

- bid, lid references ResidenceLaundryMachine(bid,lid)
- cid references LoadsWashingCard(cid)

NOT NULL:

- hasFunds

FDs and Normal Forms

FDs and normal form violations
double check PKs and CKs
All PKs and (non-PK) CKs are identified.

UserLivesIn

- uid -> uname, bid

CampusResidence

- *PK: bid*
- *CK: bname, address*
- bid -> bname, address
- bname -> address, bid
- address -> bname, bid

ResidenceLaundryMachine

- *PK: bid, lid*
- bid, lid->brand, washingStatus, bid

Wash

- *PK: bid, lid*
- bid, lid -> washType, washSession
- washType -> washSession
    - **VIOLATION**

Dryer

- *PK: bid, lid*
- bid, lid -> dryType, drySession
- dryType -> drySession
    - **VIOLATION**

LoadsWashingCard

- *PK: cid*
- cid -> balance, uid

RecordsTransaction

- *PK: tid*
- tid -> day, time, price, cid

ResidenceManager

- *PK: mid*
- *CK: memail, mphone*
- mid -> name, mphone, email
- email -> mid, name, mphone
- mphone -> email, mid, name

Manages

- *PK: bid, lid, mid*
- *Only PK so only trivial FD where*
    - *bid, lid, mid-> bid, lid, mid*

Technician

- *PK: techid*
- techid -> techname, techphone

ReportsFeedback

- *PK: fid*
- fid -> feedbackType, comments, uid

Repairs

- *PK: bid, lid, techid*
- *Only PK so only trivial FD where*
    - *bid, lid, techid -> bid, lid, techid*

Pays

- *PK: cid, bid, lid*
- *cid, bid, lid -> hasFunds*

University of British Columbia, Vancouver
Department of Computer Science

Normalization
Normalization done and PK and CK are specified clearly
***NOTE**: primary keys are underlined and foreign keys are bolded*

Wash(bid, lid, washType, washSession)

- washType -> washSession
    - **VIOLATION of 3NF**
        - washType is not a superkey and not part of minimal key so it violates 3NF
- We break it down in BCNF and ensure that relations are not lost
    - WashTypeSession(<u>washType</u>, washSession)
    - Washer(**<u>bid</u>**, **<u>lid</u>**, **washType**)
        - bid, lid REFERENCES ResidenceLaundryMachine(bid, lid)
        - washType REFERENCES WashTypeSession(washType)

Dryer(bid, lid, washType, washSession)

- dryType -> drySession
    - **VIOLATION of 3NF**
        - dryType is not a superkey and not part of minimal key so it violates 3NF
- We break it down in BCNF and ensure that relations are not lost
    - DryTypeSession(<u>dryType</u>, drySession)
    - Dryer(**<u>bid</u>**, **<u>lid</u>**, **dryType**)
        - bid, lid REFERENCES ResidenceLaundryMachine(bid, lid)
        - dryType REFERENCES DryTypeSession(dryType)

University of British Columbia, Vancouver
Department of Computer Science

SQL DDL
All necessary CREATE TABLE statements (SQL DDL) are defined for your database.

UserLivesIn Table
```
CREATE TABLE UserLivesIn (
   uid INT PRIMARY KEY,
   bid INT NOT NULL,
   uname VARCHAR(20) NOT NULL DEFAULT 'User',
   FOREIGN KEY (bid) REFERENCES CampusResidence(bid)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

CampusResidence Table
```
CREATE TABLE CampusResidence (
   bid INT PRIMARY KEY,
   bname VARCHAR(20) NOT NULL UNIQUE,
   address VARCHAR(50) NOT NULL UNIQUE
);
```

ReportsFeedback Table
```
CREATE TABLE ReportsFeedback (
   fid INT PRIMARY KEY,
   uid INT NOT NULL,
   feedbackType VARCHAR(20),
   comments VARCHAR(70) NOT NULL,
   FOREIGN KEY (uid) REFERENCES UserLivesIn(uid)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

LoadsWashingCard Table
```
CREATE TABLE LoadsWashingCard (
   cid INT PRIMARY KEY,
   uid INT NOT NULL,
   balanceINT NOT NULL DEFAULT 0,
   FOREIGN KEY (uid) REFERENCES UserLivesIn(uid)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

Pays Table

```
CREATE TABLE Pays (
   cid INT,
   bid INT,
   lid INT,
   hasfunds BOOLEAN NOT NULL DEFAULT TRUE,
   PRIMARY KEY (cid, bid, lid),
   FOREIGN KEY (cid) REFERENCES LoadsWashingCard(cid)
         ON UPDATE CASCADE
         ON DELETE CASCADE,
   FOREIGN KEY (bid, lid) REFERENCES ResidenceLaundryMachine(bid, lid)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

ResidenceLaundryMachine Table

```
CREATE TABLE ResidenceLaundryMachine (
   bid INT,
   lid INT,
   brand VARCHAR(15) DEFAULT 'Coinamatic',
   model VARCHAR(20),
   washing_status VARCHAR(10) NOT NULL,
   PRIMARY KEY (bid, lid),
   FOREIGN KEY (bid) REFERENCES CampusResidence(bid)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

Manages Table

```
CREATE TABLE Manages (
   bid INT,
   lid INT,
   mid INT,
   PRIMARY KEY (bid, lid, mid),
   FOREIGN KEY (bid, lid) REFERENCES ResidenceLaundryMachine(bid, lid),
         ON UPDATE CASCADE
         ON DELETE CASCADE,
   FOREIGN KEY (mid) REFERENCES ResidenceManager(mid)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

Washer Table

```
CREATE TABLE Washer (
   bid INT,
   lid INT,
   washType VARCHAR(20) NOT NULL,
   PRIMARY KEY (bid, lid),
   FOREIGN KEY (bid, lid) REFERENCES ResidenceLaundryMachine(bid, lid)
         ON UPDATE CASCADE
         ON DELETE CASCADE,
   FOREIGN KEY (washType) REFERENCES WashTypeSession(washType)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

WashTypeSession Table

```
CREATE TABLE WashTypeSession (
   washType VARCHAR(20) NOT NULL,
   washSession VARCHAR(20) NOT NULL,
   PRIMARY KEY (washType)
)
```

Dryer Table

```
CREATE TABLE Dryer (
   bid INT,
   lid INT,
   dryType VARCHAR(20) NOT NULL,
   PRIMARY KEY (bid, lid),
   FOREIGN KEY (bid, lid) REFERENCES ResidenceLaundryMachine(bid, lid)
         ON UPDATE CASCADE
         ON DELETE CASCADE,
   FOREIGN KEY (dryType) REFERENCES DryTypeSession(dryType)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

DryTypeSession Table

```
CREATE TABLE DryTypeSession (
   dryType VARCHAR(20) NOT NULL,
   drySession VARCHAR(20) NOT NULL,
   PRIMARY KEY (dryType)
```

```
)
```

ResidenceManager Table

```
CREATE TABLE ResidenceManager (
   mid INT PRIMARY KEY,
   mname VARCHAR(20) NOT NULL,
   mphone VARCHAR(15) NOT NULL UNIQUE,
   memail VARCHAR(30) NOT NULL UNIQUE
);
```

RecordsTransaction Table

```
CREATE TABLE RecordsTransaction (
   tid INT PRIMARY KEY,
   cid INT NOT NULL,
   day DATE NOT NULL,
   time VARCHAR(10) NOT NULL,
   price INT NOT NULL,
   FOREIGN KEY (cid) REFERENCES LoadsWashingCard(cid)
         ON UPDATE CASCADE
);
```

Repairs Table

```
CREATE TABLE Repairs (
   bid INT,
   lid INT,
   techid INT,
   PRIMARY KEY (bid, lid, techid),
   FOREIGN KEY (bid, lid) REFERENCES ResidenceLaundryMachine(bid, lid),
         ON UPDATE CASCADE
         ON DELETE CASCADE
   FOREIGN KEY (techid) REFERENCES Technician(techid)
         ON UPDATE CASCADE
         ON DELETE CASCADE
);
```

Technician Table

```
CREATE TABLE Technician (
   techid INT PRIMARY KEY,
   techname VARCHAR(20) NOT NULL,
   techphone VARCHAR(15) NOT NULL DEFAULT '778-807-9000'
);
```

INSERT statements to populate tables:

CampusResidence

```
INSERT INTO CampusResidence (bid, bname, address) VALUES
(1, 'Nicola', '5000 Student Union Blvd'),
(2, 'Nicole', '5100 Student Union Blvd'),
(3, Nicoli, '5200 Student Union Blvd'),
(4, 'Nicolo', '5300 Student Union Blvd'),
(5, 'Nicolu', '5400 Student Union Blvd');
```

UserLivesIn

```
INSERT INTO UserLivesIn (uID, bid, uname) VALUES
(1001, 1, 'John Doe'),
(1002, 2, 'Jane Smith'),
(1003, 3, 'Bob Johnson'),
(1004, 4, 'Alice Brown'),
(1005, 5, 'Charlie Davis');
```

ReportsFeedback

```
INSERT INTO ReportsFeedback (fid, uid, FeedbackType, Comments) VALUES
(1, 1001, 'Complaint', 'Machine not working'),
(2, 1002, 'Suggestion', 'Need more dryers'),
(3, 1003, 'Compliment', 'Great service'),
(4, 1004, 'Complaint', 'Card reader malfunctioning'),
(5, 1005, 'Suggestion', 'Extend laundry hours');
```

LoadsWashingCard

```
INSERT INTO LoadsWashingCard (cid, uid, Balance) VALUES
(101, 1001, 20),
(102, 1002, 15),
(103, 1003, 30),
(104, 1004, 10),
(105, 1005, 25);
```

ResidenceLaundryMachine

```
INSERT INTO ResidenceLaundryMachine (bid, lid, brand, model, washing_status) VALUES
(1, 1, 'Coinamatic', 'WM100', 'Available'),
(1, 2, 'Coinamatic', 'DM200', 'In use'),
(2, 1, ‘Samsung’, 'Commercial', 'Available'),
(3, 1, 'LG', 'TurboWash', 'Out of order'),
(4, 1, 'Coinamatic', 'WM100', 'Available');
```

Pays

```
INSERT INTO Pays (cid, bid, lid, hasfunds) VALUES
(101, 1, 1, TRUE),
(102, 1, 2, TRUE),
(103, 2, 1, TRUE),
(104, 3, 1, FALSE),
(105, 4, 1, TRUE);
```

Manages

```
INSERT INTO Manages (bid, lid, mid) VALUES
(1, 1, 501),
(1, 2, 501),
(2, 1, 502),
(3, 1, 503),
(4, 1, 504);
```

Technician

```
INSERT INTO Technician (techid, techname, techphone) VALUES
(201, 'Mike Mike, '778-807-9001'),
(202, 'Sarah Nguyen, '778-807-9002'),
(203, 'Tom Cruise, '778-807-9003'),
(204, 'Lisa Lisa, '778-807-9004'),
(205, 'Taylor Swift', '778-807-9005');
```

Repairs

```
INSERT INTO Repairs (bid, lid, techid) VALUES
(1, 1, 201),
(1, 2, 202),
(2, 1, 203),
(3, 1, 204),
(4, 1, 205);
```

Washer Table

```
INSERT INTO Washer (bid, lid, washType) VALUES
(1, 1, 'Normal'),
(2, 1, 'Quick Wash),
(3, 1, 'Delicate'),
(4, 1, 'Delicate''),
(5, 1, 'Normal');
```

WashTypeSession Table
INSERT INTO WashTypeSession (washType, washSession) VALUES
('Normal', '40 minutes'),
('Quick Wash', '15 minutes'),
('Delicate', '30 minutes'),
('Long Wash', '50 minutes'),
('Hot Wash', '35 minutes');

Dryer Table
INSERT INTO Dryer (bid, lid, dryType) VALUES
(1, 2, 'Normal', '45 minutes'),
(2, 2, 'Delicate', '30 minutes'),
(3, 2, 'Time Dry', '60 minutes'),
(4, 2, 'Quick Dry', '20 minutes'),
(5, 2, 'Long Dry', '60 minutes');

DryTypeSession Table
INSERT INTO DryTypeSession (dryType, drySession) VALUES
('Normal', '45 minutes'),
('Delicate', '30 minutes'),
('Time Dry', '60 minutes'),
('Quick Dry', '20 minutes'),
('Long Dry', '60 minutes');

Residence Manager Table
INSERT INTO ResidenceManager (mid, mname, mphone, memail) VALUES
(501, 'John Smith', '778-807-9001', 'john.smith@gmail.com'),
(502, 'Jane Doe', '778-807-9002', 'jane.doe@gmail.com'),
(503, 'Mike Johnson', '778-807-9003', 'mike.johnson@gmail.com'),
(504, 'Sarah Brown', '778-807-9004', 'sarah.brown@gmail.com'),
(505, 'Chris Lee', '778-807-9005', 'chris.lee@gmail.com');

Records Transaction Table
INSERT INTO RecordsTransaction (tid, cid, day, time, price) VALUES
(1, 101, 2024-09-10, '22:30:00', 150),
(2, 102, '2024-09-15', '10:13:30', 300),
(3, 103, '2024-10-01', '14:09:00', 150),
(4, 104, '2024-10-05', '11:45:45', 150),
(5, 105, '2024-10-13', '21:10:10', 150);