University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 1

Date: September 18, 2024

Group Number: 4

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Angela Felicia	51190304	angelafc	angelafc@students.cs.ubc.ca
Qinying Li	10959491	carina00	carina00@students.cs.ubc.ca
Leo Shang	52598174	lshang04	leoshang12@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

1. Project Description

a. What is the domain of the application? Describe it

The domain of an application refers to the area of knowledge your application resides in. For example, if I am making an application for a hospital, the domain would be something like healthcare/patient management/logistics

- 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

b. What aspects of the domain are modeled by the database?

In answering this question, you will want to talk about what your project is trying to address and how it fits within the domain. It is likely that in the process of answering these questions you will bring up examples of a real-life situation that the application could be applied to.

- 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.

2. Data Specifications

a. What functionality will the database provide?

I.e., what kinds of things will people using the database be able to do. (3-5 sentences)

The database will allow users to check the status of washing machines (available, in use, or needing repair) and purchase laundry cards for machine use. It will manage user registration and track the residences they live in, while also monitoring the brand, model, and availability of laundry machines for technicians. Payment history for loading funds onto laundry cards will be tracked to provide spending insights, and technicians' repair work will be recorded to ensure accountability. Users can also check machine availability, report feedback, and set washing and drying times universally across all machines. Finally, it will track which residence managers are responsible for specific laundry machines.

3. Application Platform(2-3 sentences)

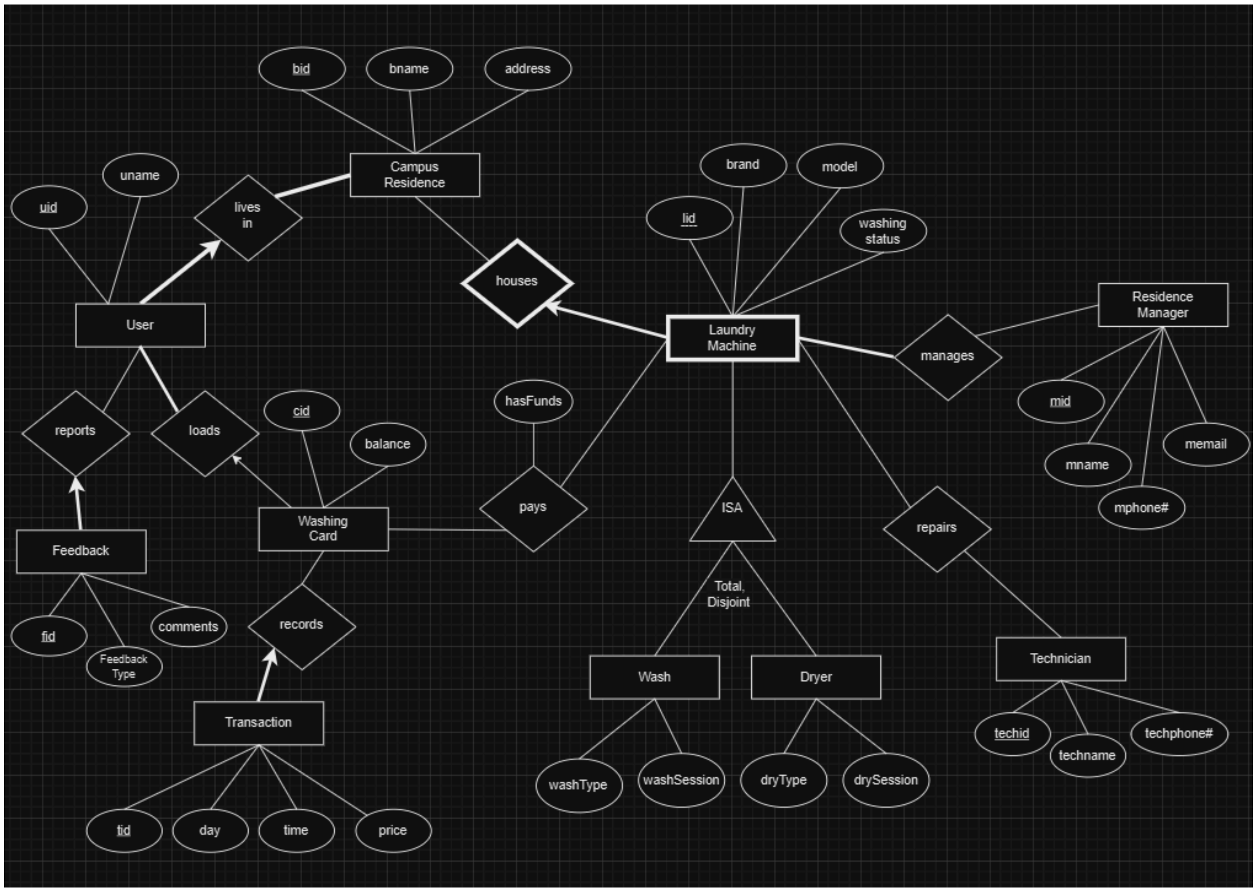
a. What database will your project use

(department provided Oracle, your own MySQL, etc.)?) See the “Project Platforms” section of this document for more information.

We will use Oracle for our database. Our application technology stack would be Oracle’s platform from the CS department and JavaScript.

b. What is your expected application technology stack (i.e., what programming languages and libraries do you want to use)? See the “Project Platforms” section of this document for more information.
i. You can change/adjust your tech stack later as you learn more about how to get started for the project via latter tutorials.
Our team will be utilizing Oracle’s platform from the CS department and incorporating JavaScript into our final project. Additionally, two of our team members have direct experience with web development so we want to implement React and Tailwind or CSS for the front end. We may change our tech stack depending on the latter tutorials.

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



lid is underlined in the weak entity. Image may not be entirely clear but it is underlined
Entities
Each entity set has at least one non-primary key attribute (primary keys are underlined).

1. User: uid, uname
2. CampusResidence: bid, bname, address

3. LaundryMachine **(weak entity)**: <u>lid </u>(No dotted underline option on google docs), brand, model, washingStatus, **ISA**
    a. Wash: washType, washSession
    b. Dryer: dryType, drySession
4. WashingCard: <u>cid</u>, balance
5. Transaction: <u>tid</u>, day, time, price
6. ResidenceManager: <u>mid</u>, mname, mphone#, email
7. Technician: <u>techid</u>, teachname, techphone#
8. Feedback: <u>fid</u>, feedbackType, comments

At least 7 or more relationships (Relationship underlined)

1. User <u>lives in</u> CampusResidence
2. User <u>loads</u> WashingCard
3. User <u>reports</u> Feedback
4. WashingCard <u>records</u> Transaction
5. WashingCard <u>pays</u> (attribute: hasFunds) LaundryMachine
6. ResidenceManager <u>manages</u> LaundryMachine
7. Technician <u>repairs</u> LaundryMachine
8. CampusResidence <u>houses</u> LaundryMachine (weak entity)

Cardinality and participation constraints

1. User (**many, participation constraint**) <u>lives in</u> CampusResidence (**one, participation constraint**)
    a. All users must live in campus resident, and all campus resident needs at least one user
2. User (**one, participation constraint**) <u>loads</u> WashingCard (**many**)
3. User(**one**) <u>reports</u> Feedback (**many, participation constraint**)
4. WashingCard (**one**) <u>records</u> Transaction (**many, participation constraint**)
5. WashingCard (**many**) <u>pays</u> (attribute: hasFunds) LaundryMachine (**many**)
6. ResidenceManager (**many**) <u>manages</u> LaundryMachine (**many, participation constraint**)
7. Technician (**many**) <u>repairs</u> LaundryMachine (**many**)
8. CampusResidence (**one**) <u>houses</u> LaundryMachine (**many**) => weak entity

At least one meaningful (non-trivial) ISA hierarchy (of entities)

1. LaundryMachine ISA <u>Wash</u> (attribute: washType, washSession) or <u>Dryer</u> (attribute: dryType, drySession)

At least one meaningful (non-trivial) weak entity (weak entity is underlined)

1. CampusResidence houses <u>LaundryMachine</u>