University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 1

Date: 26-May-2023

Group Number: 17

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Khammy Saychaleun	40259574	v1m0l	khammyschl@gmail.com
Keshav Gopinath	61086260	z8z2b	keshavkarthikk@gmail.com
Aayush Kogar	62235312	s7j9s	18aayushk@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

University of British Columbia, Vancouver
Department of Computer Science

Project Description

Problem Statement: Finding an affordable apartment is known to be incredibly difficult in Vancouver. It's especially difficult for first-time searchers in students, who often are forced to look for non-dorm residences due to shortages of on-campus housing. The housing search is a time-consuming process that involves going on multiple different sites, sifting through scam housing, and maintaining a mini-housing CRM.

Solution: The domain of our application (**STUHOUSING) -** helps students find housing in school areas by acting as a compiler of all housing available on various different platforms and supporting your search management (from discovery and visits to signing the eventual contract). A school support contact (employee) will act as the middle-man to all the different platforms for our minimum viable product. In the future, post this course, we hope to automate this role.

Database: Our database will model a school-wise application allowing a student to query for housing to meet their specifications and then manage the search process (including visits, applications, and signing the contract). The applications will not be integrated to other platforms yet, as it is assumed this will be done by the school support contact

University of British Columbia, Vancouver
Department of Computer Science

Data Specification

User stories:

Searcher

- Searcher should be able to login
- Searcher should have specific school support contacts
- Searcher should be able to filter through rentals/sublets by standard criteria
- Searcher should be able to tag units they're interested in
- Searcher should be able to schedule visits for units
- Searcher should be able to request and store testimonials (referrals)
- Searcher should be able to view scrapped details/photos of the unit
- Searcher should be able to apply for a unit
- Searcher should be able to see the current status of applications
- Searcher should be able to receive a contract
- Searcher will represent all prospective roommates on our application. *In the future, we will be adding additional functionality to allow multiple searchers to apply to a residence together.*

School Support Contact

- School Support Contact should be able to see all applications in their school area.
- School Support Contact must be on the ground in the school area, therefore cannot be a contact for multiple schools at once (eg. a contact cannot work at UBC and SFU, only either one).

The pages the application will have for the searcher are:

- **Login Page**
- **School selection Page** (Searchers select their school)
- **Discovery Page** (Searchers filter for units details)
- **Unit Details Page** (Searchers can apply and schedule a visit)
- **Profile Page** (Searchers can update their details and request testimonials)
- **Your Search Page** (Searchers can see units they've tagged, visited, or applied for with progress)
- **Contract Page** (Searchers can see price, duration, and link to contract)
- **Visits Page** (Searchers can see all visits)

The Pages the application will have for the support contact are:

- **Login Page**
- **Selection Page** (Support contact can select a student, unit, contract, or visit to view and/or edit the details as necessary)

Application Platform

The following is the specifications of the tech stack that we plan to use for building a minimum viable product:

Backend and Frontend: PHP
OS: Linux (Undergraduate Servers)
Database: Oracle

The current scripting exploration creates a user interface that more or less looks like this:

https://www.students.cs.ubc.ca/~kitachi/oracle-test.php (note, this was the tutorial from a previous term of CPSC 304 which we used to test the connection to the Oracle Database and the proper functioning of the Apache Proxy.

Note: We are experimenting with using different frontends (React in particular) to create a more visually appealing GUI. We will first attempt to host the React front-end on the Linux Server (with X11 forwarding to open a browser window). If time does not permit, we will use Netlify to set up a publicly accessible front-end. Our main goal is to create a minimal front-end.

University of British Columbia, Vancouver
Department of Computer Science

Entity Relationship Diagram

The ERD diagram has been attached below and a link to the original document can be found here . You will have to go to the `DRAFT-ERD` tab in the draw.io page to find the ERD diagram.