University of British Columbia, Department of Computer Science

CPSC 304

Cover Page for Project Proposal

Date: September 27, 2018

Group Members:

Name	Student Number	CS Userid	Tutorial Section	Email Address
Edmund Oh	17335167	d4b1b	T1H	1edmundoh@gmail.com
Shamit Rahman	19965152	x2l0b	T1G	rahmanshamit@gmail.com
Guilherme Lameira de Almeida	20318151	k3n0b	T1E	gui.l.a@hotmail.com
Rodolfo Orozco	10282168	t3x0b	T1C	raov97@outlook.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 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

Deliverable 1: Identify the different kinds of users that your application will service.

Subletter:
- Require email and password
- Create, edit, and delete sublet posts
- Receive and respond sublet requests
- Have access to sublet history

Subletter will see their main landing page once logged in, where they will have their post and every request from a potential Sublette.

Sublette:
- Require email, password, Sublette information (first name, last name, and contact description)
- Browse sublet posts
- Create sublet requests
- Update Sublette information

Sublette will see a list of available sublets on the page and will be able to create sublet requests according to what they see and/or filter.

Deliverable 2: You need to have at least one INSERT statement:

Create Post (create a post with an existing room or a new room)
Input:
- AUTH (email, password)
- Price
- Start Date
- End Date
- Additional Info
- Existing Room (Residence name, Building, Room Number)if Subletter had rented out a room before
- New Room (Residence name, Building, Room Number, Floor, Gender Restriction, Unit type) if this is the first room Subletter is making a post about.

Output:
- Post Id

INSERT will be used to insert the input information filled in by the Subletter into a Sublet-Post table to have a new Sublet post tuple.

Deliverable 3: You need to have at least one DELETE statement.

Delete Post
Input:
- Auth (email, password)
- Post Id

A Subletter will be able to delete any currently active Sublet Post that he/she has made. This remove the information of that particular post from the tuple in the Sublet-Posts table using the DELETE query.

Deliverable 4: You need to have at least one UPDATE statement.

Edit Post

Input:
- Auth (email, password)
- Post Id
- Any post field to be edited – When a field is edited, the UPDATE query will be used to update a attribute in a tuple.

A Sublette will also be able to update Sublette information.

Deliverable 5: At least one meaningful query must join 3 or more tables.

Get Filter Count

Inputs:
Outputs:
- Count for every res name, unit types, has kitchen, # of bathrooms, # residents

When a Sublette will filter our specific sublet posts that meets his/her criteria, to make the query easier, using CREATE VIEW a virtual table by joining Room, Residence and Unit Type tables a Complete Post Info table will be created.

Deliverable 6: At least one other meaningful query needs to join 2 or more tables.

Get Create Post Info

Input:
- Auth (email, password)

Output:
- Rooms (Residence name, Building, Room Number, Floor, Gender Restriction, Unit type)
- Residences (Residence name)
- Unit Types (Unit type name)

When a Subletter will look up previous rooms that he/she had posted about before, all of the basic information about that room will show as well and for this a virtual table formed by using JOIN on the Room and Residence tables.

Deliverable 7: At least one query must be an interesting GROUP BY query (aggregation). Describe it.

Inputs:
Outputs:
- Count for every res name, unit types, has kitchen, # of bathrooms, # residents

When a Sublette filters out posts according to their criteria, a count will be shown for that specific criteria. For example, if they filter out to only look at posts that has a kitchen, those posts will show on the UI and the total number of the posts with a kitchen will also be shown.
For example, the query will COUNT(has kitchen), FROM(CompleteSubletPost) table, using GROUP BY(has kitchen).

Deliverables 8-10: Describe the other queries you plan to have (these can be simpler queries), so that you have at least 10 SQL statements overall.

Deliverable 8:

Log In

Input:

- Email
- Password

When the User inputs their email and Password, the SELECT query will be used to look up the given email and Password from the User table to check if it is correct.

Deliverable 9:

Get Post

Input:
- Auth (email, password)
- Post Id

Output:
- Price, Start Date, End Date, Additional Info, Room,

For eg:

```
SELECT Price, Start Date, End Date, Additional Info, Room
FROM [Complete Post Info]
```

The FROM query will be used to specify which table we will be getting specific attributes or tuples from. For example, when a user wants to see a specific post.

Deliverable 10:

Create Subletee Info

Input:
- Auth (email, password)
- First Name
- Last Name
- Contact information

When a Sublette creates their account, the INSERT query will be used to insert their information to the Subletter table for future use and authentication.

Deliverable 11: You need to have at least one view for your database, created using the CREATE VIEW statement in SQL. It should be a proper subset of a table. When the user wants to display all the data from this view, they will get all the columns specified by the view, but not all of the columns in the underlying table.

Get Filter Count

Inputs:

Outputs:

- Count for every res name, unit types, has kitchen, # of bathrooms, # residents

CREATE VIEW will be used to make a virtual table by joining Room, Residence and Unit tables to show the user only the tuples that meet their filtering criteria. CREATE VIEW is used here as opposed to other queries as we want the user to only see a virtual table instead of showing them the actual database from the backend.

Deliverable 12:

1. **Rodolfo**
2. **Gui**
3. **Edmund**
4. **Shamit**
5. **All of us**
6. **All of us**
7. **All of us**