

QBNB

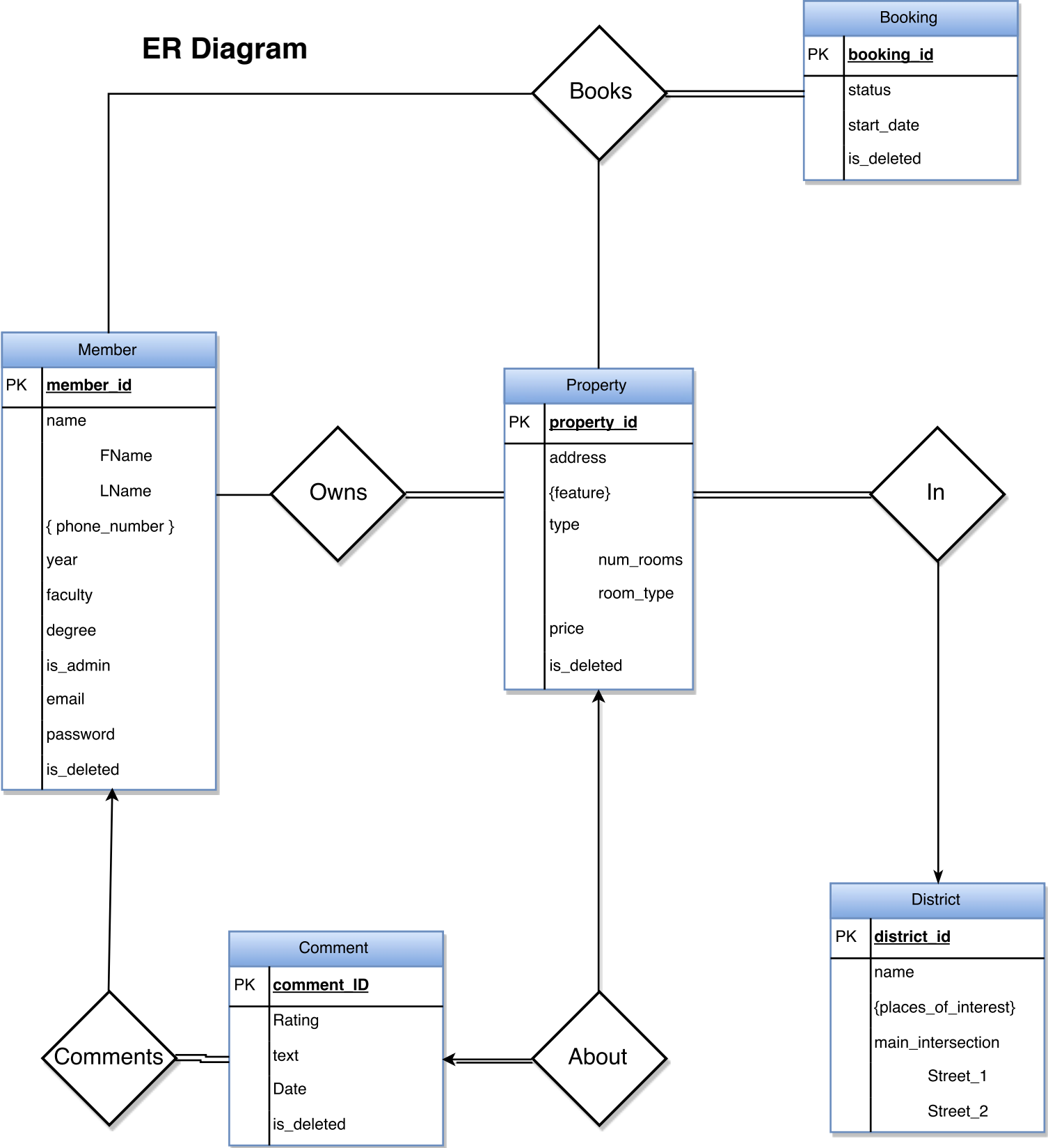
CMPE 332



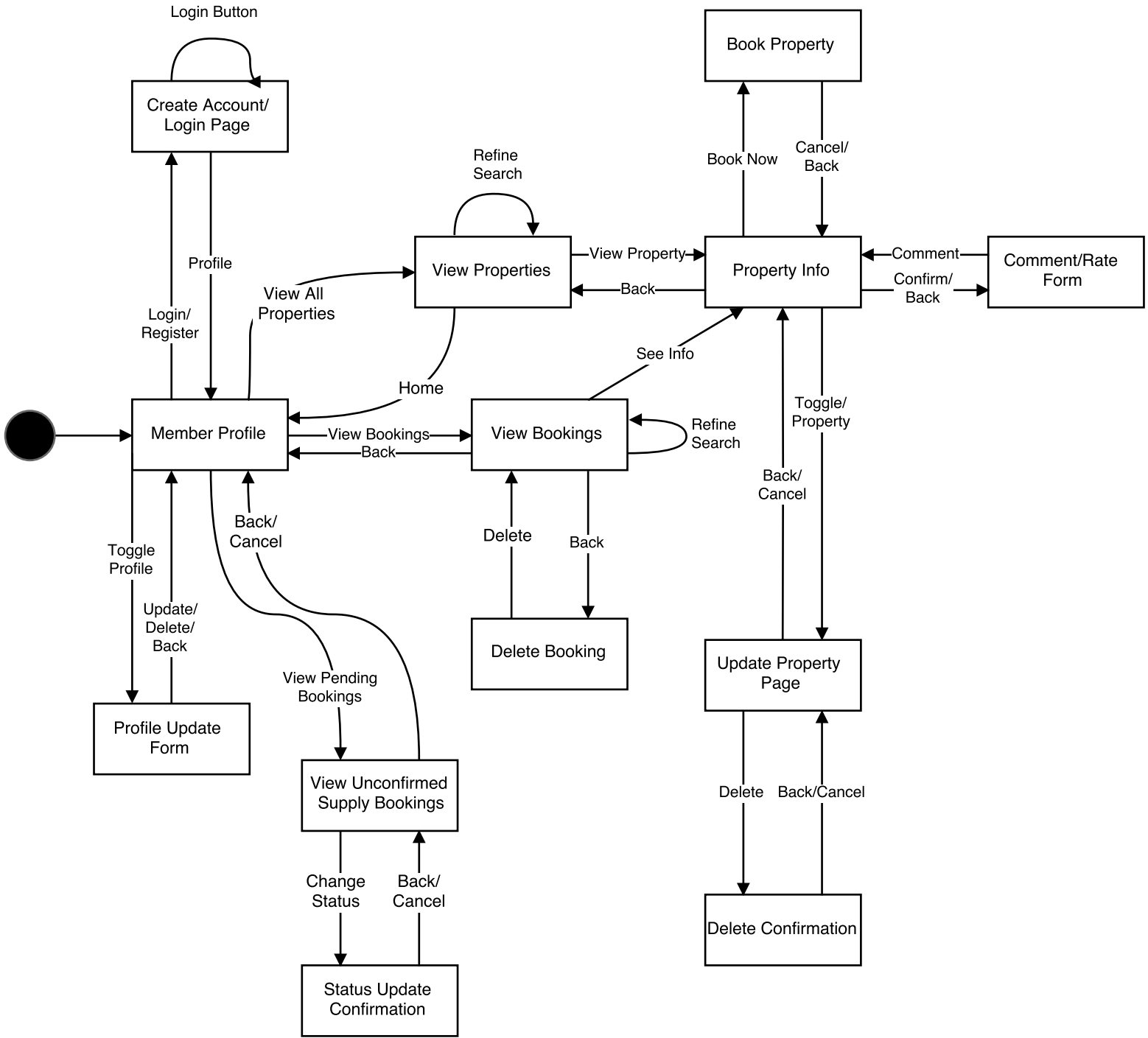
MARCH 31, 2016

Jacqueline Craig, 10043961, Justin Shimkovitz, 10048941, Julian Wilson 10053506

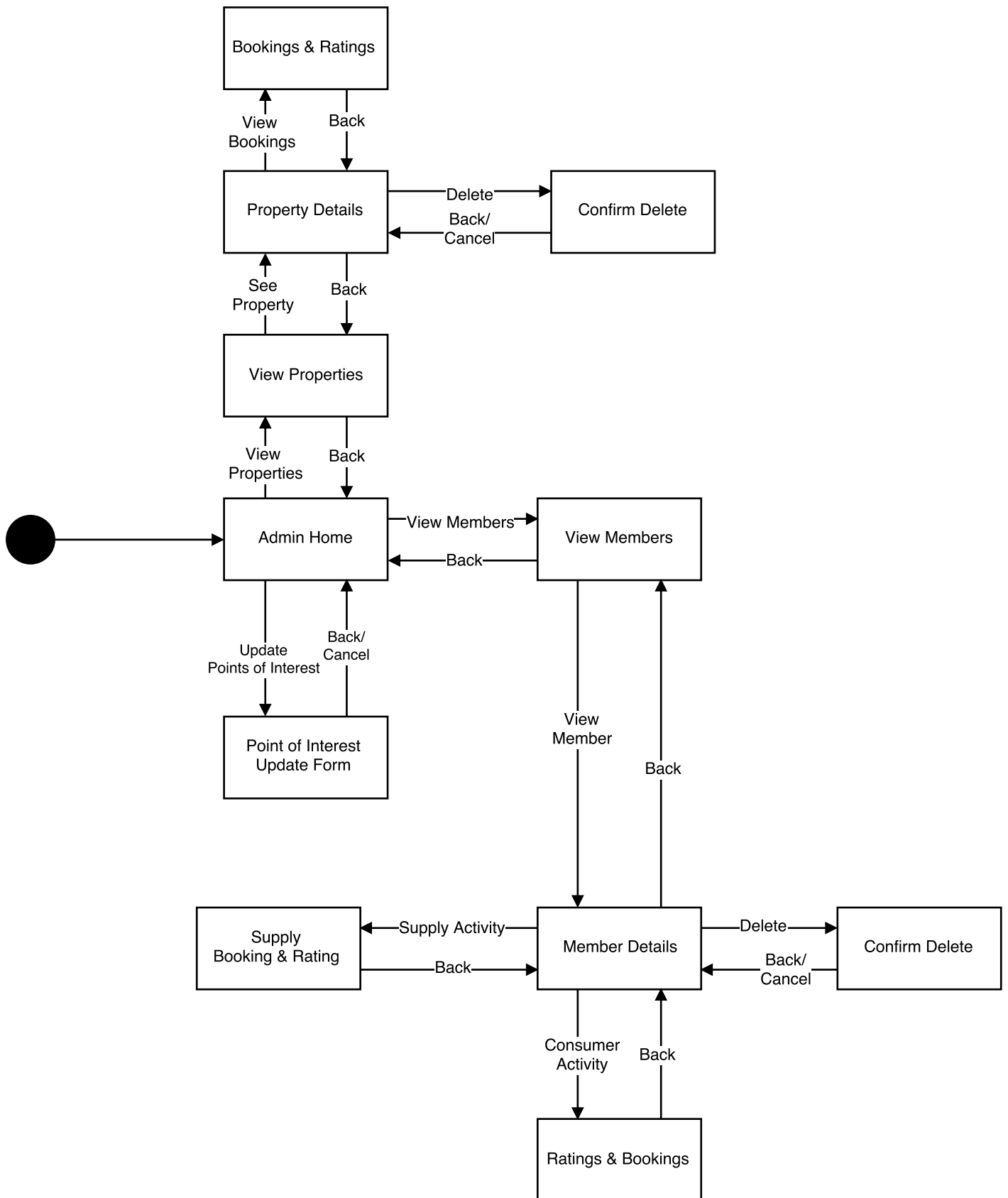
ER Diagram



State Diagram - Member



State Diagram - Admin



Assumptions

The basic functional and data requirements outlined in the project description need to be supplemented with the following assumptions to better define the scope of creating a QBbB project.

Comment

- If a member gets rid of a property and all bookings related to that will be marked as deleted with the is deleted tag.
- A member can only comment once per booking and there can be only one comment and reply
- Replies are just comments and not limited to the person who owns the property

Members (& Booking)

- When a member cancels their membership, all properties for that member will be not appear along with all bookings for that property will be marked as deleted and bookings that they have made will be deleted.
- Two members can have the same name; therefore, the unique identifier of the members are their email addresses.
- Each member can have more than one property.
- A member can sign up with only one degree, even if they have earned more than one at Queen's.
- A member can be both a supplier and a consumer.
- You have to be a member to list a property, make a booking, comment or reply.
- A member can book more than one booking (but not at the same time)

Properties (& Booking)

- Addresses are unique. For example, if there are multiple properties at one address, they are differentiated by the apartment number.
- A property is only in one district.
- A single property can't be booked during the same period for two different members.
- The booking date is the start date and lasts for one week, as mentioned in the project description.

District

- QBnB occurs in only one city.
- The city districts don't change over time.
- The admins can add or remove points of each District.
- Districts are unique.
- Each district will have the main intersection that is included in that district as a point of reference for the member booking a property.

Database qbnb

Table structure for table Booking

Column	Type	Null	Default
<i>booking_id</i>	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Booking

1	Confirmed	2015-05-06	0	2	1
2	Deleted	2015-05-06	0	2	2
3	Denied	2015-05-08	0	3	1
4	Unconfirmed	2015-05-09	0	4	2
5	Confirmed	2015-08-04	0	1	3
6	Unconfirmed	2014-11-01	0	1	2
7	Denied	2014-11-18	0	1	6
8	Confirmed	2014-12-01	0	4	7
9	Unconfirmed	2014-12-23	0	2	3
10	Confirmed	2015-08-27	0	4	8
11	Confirmed	2015-12-23	0	7	9
12	Unconfirmed	2016-02-25	0	7	10
13	Confirmed	2016-02-28	1	3	11
14	Unconfirmed	2016-05-19	0	4	1
15	Unconfirmed	2016-09-06	0	3	2
16	Denied	2016-09-23	0	3	3
17	Confirmed	2016-11-02	0	3	5
18	Confirmed	2016-11-07	0	4	7
19	Denied	2017-05-24	0	2	9
20	Unconfirmed	2017-06-29	1	8	11
21	Confirmed	2017-07-06	0	4	2
22	Unconfirmed	2017-08-21	0	4	5
23	Denied	2017-11-07	0	5	4
24	Confirmed	2018-01-10	0	2	6

25 Denied	2018-04-20 0 7 8
26 Confirmed	2018-05-12 0 4 10
27 Denied	2018-06-07 0 2 7
28 Unconfirmed	2018-08-12 0 1 2
29 Denied	2018-09-14 0 1 7
30 Unconfirmed	2018-11-09 0 1 2

Table structure for table Comment

Column	Type	Null	Default
<i>comment_id</i>	int(11)	No	
text	varchar(140)	No	
Rating	int(11)	Yes	NULL
Date	date	No	
is_deleted	tinyint(1)	No	
commenting_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Comment

30 Great place! Slightly dirty, but nothing unmanageable	4	2015-05-05	0 5 1
31 Thanks for your feedback! Ill make sure to get that fixed	NULL	2015-05-06	0 1 1
32 Terrible. Really terrifying area of town and nothing near it worth seeing. Would not recommend.	1	2016-01-03	0 4 3
33 Awesome Job, I would definitly stay here again!	4	2016-06-07	0 1 1
34 This was alright, not going to be here again	2	2013-10-23	0 2 2
35 Really liked the light in this place!	4	2013-10-26	0 2 3
36 Awesome Job, I would definitly stay here again!	4	2014-01-28	0 3 4
37 So many points of interest around here!	4	2014-02-23	0 4 4
38 I love the Qbnb app	5	2014-04-10	0 6 5
39 Awesome Job, I would definitly stay here again!	4	2014-05-13	0 3 6

40 Had a great time!	5	2014-08-17	0 5 6
41 Shouldnt have gone here	3	2014-09-12	0 7 7
42 I love the Qbnb app	5	2014-10-17	0 6 7
43 This was alright, not going to be here again	2	2014-11-23	0 5 7
44 Awesome Job, I would definietly stay here again!	4	2015-02-24	0 3 9
45 So many points of interest around here!	4	2015-06-17	0 5 8
46 Really liked the light in this place!	4	2015-08-03	0 4 8
47 Had a great time!	5	2015-08-05	0 7 1
48 Shouldnt have gone here	3	2015-08-09	0 1 3
49 So many points of interest around here!	4	2015-08-31	0 6 5
50 Awesome Job, I would definietly stay here again!	4	2015-10-03	0 6 11
51 Really liked the light in this place!	4	2015-10-26	0 5 2
52 So many points of interest around here!	4	2015-12-20	0 4 4
53 This was alright, not going to be here again	2	2016-03-20	0 6 8
54 Awesome Job, I would definietly stay here again!	4	2016-05-18	0 7 10
55 Had a great time!	5	2016-05-21	0 8 4
56 Shouldnt have gone here	3	2016-07-30	0 2 7
57 Awesome Job, I would definietly stay here again!	4	2016-08-02	0 8 9
58 This was alright, not going to be here again	2	2016-10-20	0 3 5

Table structure for table District

Column	Type	Null	Default
--------	------	------	---------

<i>district_id</i>	int(11)	No
district_name	varchar(45)	No
Street_1	varchar(45)	No
Street_2	varchar(45)	No

Dumping data for table District

1	Uptown	Bathurst Steeles
2	Waterfront Front	Spadina
3	Rosedale	St. Clair Oak
4	Chinatown Spadina	Queen
5	Midtown	Bathurst Eglinton

Table structure for table Feature

Column	Type	Null	Default
<i>feature_id</i>	int(11)	No	
feature	varchar(45)	No	
property_id	int(11)	No	

Dumping data for table Feature

1	2 Bathrooms	1
2	1 Bathroom	2
3	Pool	2
4	Backyard	3
5	Pool	4
6	Gym	5
7	2 Bathrooms	6
8	1 Bathroom	7
9	Pool	7
10	Backyard	9
11	Pool	9
12	Gym	10
13	Queen Beds	10
14	Queen Beds	11
15	Queen Beds	1

Table structure for table Member

Column	Type	Null	Default
<i>member_id</i>	int(11)	No	
email	varchar(45)	No	
password	varchar(45)	No	
FName	char(20)	No	
LName	char(20)	No	
year	int(11)	No	
faculty	varchar(45)	No	
degree	varchar(15)	No	
is_admin	tinyint(1)	No	
is_deleted	tinyint(1)	No	

Dumping data for table Member

1	jshimz@hotmail.com	itsjustshim	Justin	Shimkovitz	2016 Law	Criminal Law	1	0
2	coolboyjulez@queensu.ca	jwill	Julian	Wilson	2016 Commerce	Finance	1	0
3	jcisgod@gmail.com	jacjacjac	Jacqueline	Craig	2016 Engineering	Applied Math	1	0
4	fakeemail@hotmail.com	animals	John	Stewart	2018 Science	Biology	0	0
5	potus@gmail.com	password	Barack	Obama	2019 Arts	English	0	0
6	potus2@gmail.com	password2	Barack2	Obama2	2017 Arts	English	0	0
7	potus3@gmail.com	password3	Barack3	Obama3	2016 Arts	English	0	0
8	potus4@gmail.com	password4	Barack4	Obama4	2015 Arts	English	0	0

Table structure for table Phone_Number

Column	Type	Null	Default
<i>phone_number</i>	char(13)	No	
member_id	int(11)	No	

Dumping data for table Phone_Number

416-543-1234 1
905-881-4432 1
789-087-5432 2
123-456-7867 3
313-655-6667 3
818-987-4324 3

453-235-4433 4
323-655-6666 5
323-633-6667 6
416-543-1233 6
323-655-1236 7
323-655-6656 7
323-633-6167 8
323-655-6667 8

Table structure for table Points_Of_Interest

Column	Type	Null	Default
points_of_interest	varchar(45)	No	
district_id	int(11)	No	

Dumping data for table Points_Of_Interest

C.N. Tower	2
Beach	2
R.O.M.	3
Concert Hall	1
Bell Movie Theatre	4
Honest Eds	5

Table structure for table Property

Column	Type	Null	Default
<i>property_id</i>	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table Property

1	123 Maple Street	4	House	600	0	1	1
---	------------------	---	-------	-----	---	---	---

2	136 Oak Drive	3 House	200	0 1 4
3	1 Front Street	1 Apartment	1000	0 2 2
4	12 Wall Avenue	2 Condo	750	0 5 2
5	10 Martin Lane	2 Apartment	500	0 3 4
6	102 Martin Lane	3 Apartment	300	0 6 4
7	103 Martin Lane	4 Condo	200	0 7 4
8	104 Martin Lane	5 Apartment	300	0 2 4
9	105 Martin Lane	2 House	100	0 5 4
10	14 Martin Lane	3 Apartment	200	0 8 4
11	12 Martin Lane	2 Apartment	300	1 6 4

QBnB User and Admin Guide

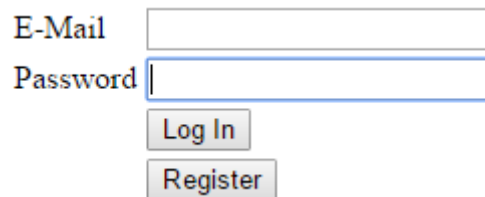
Thank you for choosing QBnB. QBnB is a revolutionary service designed to help you find places to stay in the Toronto area and also list your accommodations for others to rent in the Queen's community. It removes the expensive hotel from the equation and allows you to find a reasonably priced place to stay anywhere in Toronto *for a week*. If you're not in town, you can list your place for other students and make some money on the side.

Set-up

To set up QBnB on a local machine, ensure that you have both Apache and a MySQL server set-up properly. Create a new database called 'qbnb' with coalition 'utf_unicode_520_ci' in your phpmyadmin page. You'll now need to create an admin user to run the application. Create a new user with username: 'Assingment1', password: 'cmpe332!', hostname: 'localhost', with full access to database 'qbnb'. Provided all the .php files are accessible by the server, the application should now be accessible at localhost/qbnb/index.php, the homepage.

For users

First, log on to www.qbnb.com.¹ You'll be treated with the following prompt:

The image shows a login form with two input fields. The first field is labeled 'E-Mail' and the second is labeled 'Password'. Below the 'Password' field are two buttons: 'Log In' and 'Register'. The 'Log In' button is highlighted with a blue border.

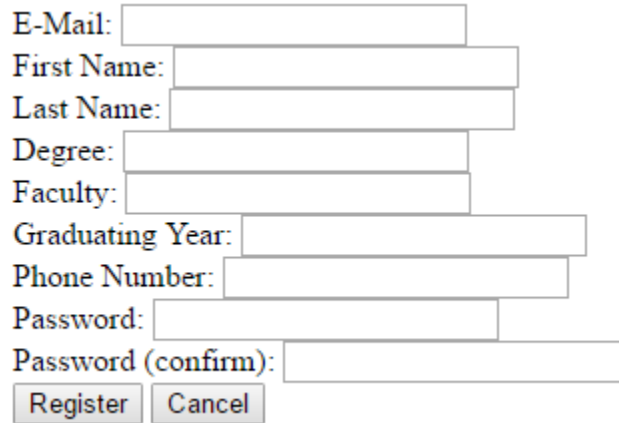
E-Mail

Password

Figure 1: The QBnB login prompt

If you are a new user, click register, and fill out the following form:

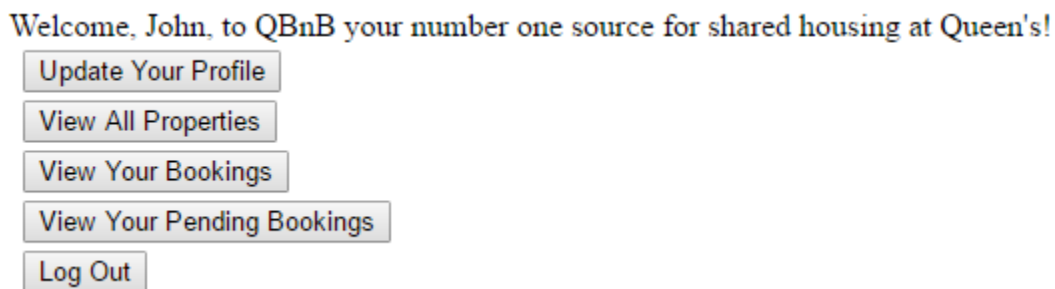
¹ For this guide, we will assume that the application and database has been hosted on a remote server and that we can access it via www.qbnb.com



A registration form with the following fields: E-Mail, First Name, Last Name, Degree, Faculty, Graduating Year, Phone Number, Password, and Password (confirm). Each field is represented by a rectangular input box. Below the fields are two buttons: 'Register' and 'Cancel'.

Figure 2: The QBnB registration page

Ensure the phone number is 10 digits, without hyphens or brackets. Once you've confirmed the information is correct, click 'Register', and you will be logged in to the application. You'll be greeted with the home page, below:



Welcome, John, to QBnB your number one source for shared housing at Queen's!

- Update Your Profile
- View All Properties
- View Your Bookings
- View Your Pending Bookings
- Log Out

Figure 3: The QBnB homepage

Here I am logged in as John. From the home page, you can update your profile where you can change your password and your personal information. Simply click 'Update Your Profile', fill out all the fields (all fields will be pre-populated based on your current information), and click 'Update Profile' on the page, and you will be returned to the homepage.

The next sections of the guide will be separated into parts, depending on what the user wishes to do.

Booking a property

If you wish to make a booking at a property, from the home page click 'View All Properties'. You'll be greeted with a page that looks like the following:

View all Properties

Address: 123 Maple Street - Number of Rooms: 4 - Room Type: House - Price: 600

View Property:

Address: 136 Oak Drive - Number of Rooms: 3 - Room Type: House - Price: 200

View Property:

Address: 1 Front Street - Number of Rooms: 1 - Room Type: Apartment - Price: 1000

View Property:

Address: 12 Wall Avenue - Number of Rooms: 2 - Room Type: Condo - Price: 750

View Property:

Address: 10 Martin Lane - Number of Rooms: 2 - Room Type: Apartment - Price: 500

View Property:

Address: 107 Apple Street - Number of Rooms: 4 - Room Type: Apartment - Price: 600

View Property:

Figure 4: Example of the view all properties page.

From here, select a property you wish to view/book and click the corresponding number for the property. Suppose you wished to look at the property at 123 Maple Street, property 1. Click '1' and you will be taken to a page that looks like this:

Property Information

Address: 123 Maple Street - Number of Rooms: 4 - Room Type: House - Price: 600

Owned By: Justin Shimkovitz

Features

2 Bathrooms

Queen Beds

Unavailable during the weeks starting on these dates:

2015-05-06

Comments:

On 2015-05-05, Barack Obama said: Great place! Slightly dirty, but nothing unmanageable

On 2015-05-06, Justin Shimkovitz said: Thanks for your feedback! I'll make sure to get that fixed

[Book Property](#)

[Comment](#)

[Back](#)

Figure 5: An example 'view property' page of 123 Maple Street

From here, you can see all information about the relevant property, including the owner and his/her e-mail address and the price per week of the place. You can also see *features*, which are owner-defined details of the accommodation. The booking dates that are unavailable are listed as well as comments from past tenants. If you're satisfied with the property, click 'Book Property'. If not, click 'Back' and you'll go back to the 'View all Properties page'. Suppose you're happy with 123 Maple Street and you click 'Book Property'. You'll get the following page:

Book a Property

Start Date (YYYY-MM-DD):

Figure 6: Selecting the date to book 123 Maple Street

Once you click book, you'll go back to 'view property' page. The booking will not be reflected in the 'Unavailable during the weeks...' section of the page until the owner (in this case, Justin Shimkovitz) approves the booking. Payment will occur once you have stayed in the place for a week.

Making a comment on a property

Suppose you stayed at 123 Maple Street for a week and wished to comment on the property for future users. To do this, simply click 'Comment' in the 'View Property' page (Figure 5) and you will be taken to the commenting page:

Make a comment

Comment:

Rating: ▼

Figure 7: Making a comment dialog

Enter your comment, the rating of your stay, and click 'Comment'. It will be reflected on the 'View property' page:

Comments:

On 2015-05-05, Barack Obama said: Great place! Slightly dirty, but nothing unmanageable

On 2015-05-06, Justin Shimkovitz said: Thanks for your feedback! I'll make sure to get that fixed

On 2016-03-29, John Stewart said: Excellent place. Great price and accommodations.

Figure 8: Comment reflected for 123 Maple Street

Setting up your property

Suppose you wished to list your property for a couple weeks because you're out of town. If you're on the homepage, click 'View All Properties', and then 'Add Property'. You'll get the property set-up page:

Address: 200 King Street East
Number of Rooms: 1
Room Type: Apartment
Price: 500
District: Waterfront ▼
[Add Property](#) [Back](#)

Figure 9: Setting up your property to list

Enter the address, the number of rooms, and the room type which could be, for instance, 'Apartment', 'Bedroom in house', 'Whole house', or 'Sofa bed'. Try to be as descriptive as possible. Enter the price you will be charging for the week and select a district from the drop-down menu. If you are happy with it all (you can change it later), click 'Add Property'. It will be reflected in the 'View all Properties' page:

Address: 10 Martin Lane - Number of Rooms: 2 - Room Type: Apartment - Price: 500
View Property: [5](#)

Address: 107 Apple Street - Number of Rooms: 4 - Room Type: Apartment - Price: 600
View Property: [7](#)

Address: 200 King Street East - Number of Rooms: 1 - Room Type: Apartment - Price: 500
View Property: [8](#)

Figure 10: The new property is reflected in the 'all properties' page.

Once you've added it to the page, you should add some features to the property, and make changes, if any. To do this, click on your new listed property ('8' in the above figure), and you'll be taken to blank property page:

Property Information

Address: 200 King Street East - Number of Rooms: 1 - Room Type: Apartment - Price: 500

Owned By: John Stewart

No available features

No current bookings

No current comments

Update Property

Delete Property

Comment

Back

Figure 11: New property page - 200 King Street East

Then click 'Update Property':

Update Property

Update Property Info

Number of Rooms:

Room Type:

Price:

Update Property

Update Property Features

Feature:

Add Additional Feature

Back

Figure 12: Updating the property info of 200 King Street East

If necessary, make changes to the number of rooms, room type, and price and then click 'Update Property'. Independently, you can add features by typing the feature in the Feature: box (ex: 'Backyard pool' or 'kitchen', 'two bathrooms') and then by clicking 'Add Additional Feature'.

Deleting/un-listing a property

Suppose you have rented out 200 King Street East for a while and you wish to un-list it. Click on the property in the 'View all Properties' page, in this case '8':

View all Properties

Address: 123 Maple Street - Number of Rooms: 4 - Room Type: House - Price: 600

View Property:

Address: 136 Oak Drive - Number of Rooms: 3 - Room Type: House - Price: 200

View Property:

Address: 1 Front Street - Number of Rooms: 1 - Room Type: Apartment - Price: 1000

View Property:

Address: 12 Wall Avenue - Number of Rooms: 2 - Room Type: Condo - Price: 750

View Property:

Address: 10 Martin Lane - Number of Rooms: 2 - Room Type: Apartment - Price: 500

View Property:

Address: 107 Apple Street - Number of Rooms: 4 - Room Type: Apartment - Price: 600

View Property:

Address: 200 King Street East - Number of Rooms: 1 - Room Type: Apartment - Price: 500

View Property:

Figure 13: Deleting 200 King Street East

And then just click delete in the 'Property Information' Window:

Property Information

Address: 200 King Street East - Number of Rooms: 1 - Room Type: Apartment - Price: 500

Owned By: John Stewart

No available features

No current bookings

No current comments

Update Property

Delete Property

Comment

Back

Figure 14: Deleting 200 King Street East

Booking History

To see bookings that you've made, simply click 'View Your Bookings' on the homepage. Here you'll see all the bookings you've made:

View Your Bookings

Address: 123 Maple Street 0

Number of Rooms: 4 - Room Type: House - Price: 600 - Owner: Justin Shimkovitz - Move In Date: 2018-01-03

Points of Interest: Concert Hall

Delete Booking: 6

Address: 123 Maple Street 0

Number of Rooms: 4 - Room Type: House - Price: 600 - Owner: Justin Shimkovitz - Move In Date: 2016-12-23

Points of Interest: Concert Hall

Delete Booking: 7

Cancel

Figure 15: All active bookings for Jon Stewart

To delete one of your upcoming bookings (that you've made), just click the number next to the booking.

Reject/accept bookings

If you're the owner of a property and another user books your property, you can choose to reject or confirm it. Rejecting it will remove the pending booking and refresh the availability dates on the property. Confirming the booking will book the property for that week and refresh the availability dates on the property. To get to this page, click 'View Your Pending Bookings' and you'll get the following page:

Booking ID: 8 - Address: 12 Wall Avenue - Date: 2015-06-05 - Status: Unconfirmed - Booked By: Jacqueline Craig

Booking ID: 17 - Address: 12 Wall Avenue - Date: 2016-03-04 - Status: Unconfirmed - Booked By: Jacqueline Craig

Booking ID:

Figure 16: Rejecting and accepting bookings

Simply type one of the Booking ID's in the field and click 'Confirm' or 'Deny' to accept or reject the booking.

For Administrators

If you're an administrator and wish to make changes to the system, log in using your admin email and password. Once logged in, you'll be greeted with the admin homepage:

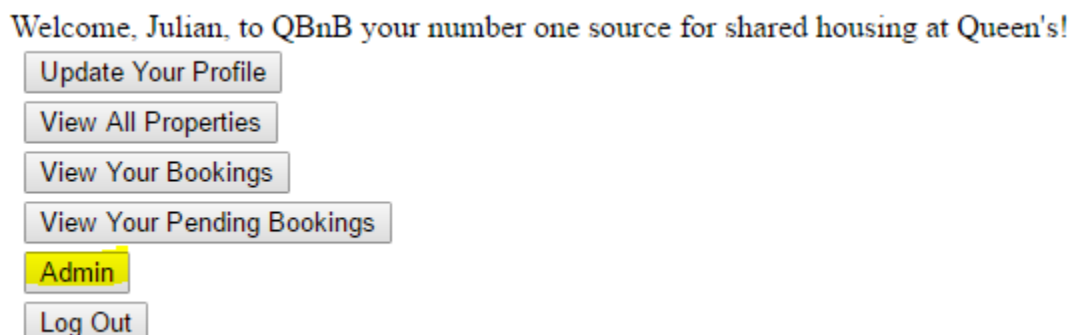


Figure 17: The altered homepage with an admin button

As you can see, there's an extra 'Admin' button that will bring you to the Admin dashboard:

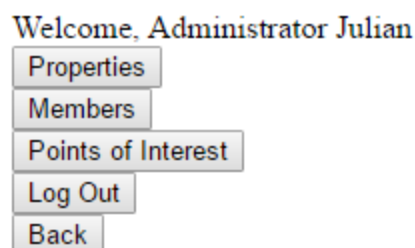


Figure 18: Administrator dashboard

Properties for Admins

If you wish to view information on properties as an admin and delete properties if needed, click the 'Properties' button on the dashboard. You'll get the 'view all properties' page for admins:

View All Properties

Active Properties

Address: 123 Maple Street
Number of Rooms: 4 Room Type: House
Price: \$600
District: Uptown
Owner: Justin Shimkovitz
View Property:

Address: 1 Front Street
Number of Rooms: 1 Room Type: Apartment
Price: \$1000
District: Waterfront
Owner: Julian Wilson
View Property:

Address: 12 Wall Avenue
Number of Rooms: 2 Room Type: Condo
Price: \$750
District: Waterfront
Owner: Barack Obama
View Property:

Figure 19: View all properties for admins

Clicking on a property, say '12 Wall Avenue' and '4', will bring you to the accommodation activity page for that property. From this page you can delete the property, see past bookings on the property and whether they were confirmed or denied, and also see ratings members have given the property:

Welcome, Administrator Julian

Delete Property

Accodmodation Activity:

Accomodation Booking:

Name: Barack Obama

Status: Denied

Start Date: 2017-11-07

Accomodation Rating:

Name: Jacqueline Craig

Rating: 4

Name: John Stewart

Rating: 4

Name: John Stewart

Rating: 4

Name: Barack4 Obama4

Rating: 5

Cancel

Figure 20: Admin accommodation page

Members for Admins

As an admin, you can also view member activity and delete the member from the system if needed.

Click the 'Members' button in the admin dashboard as seen in Figure 18: Administrator dashboardFigure 18 and you'll be brought to this page:

View all members

Welcome, Administrator Julian

Active Members

Name: Justin Shimkovitz
E-Mail: jshimz@hotmail.com
Faculty: Law
View Member: [1](#)

Name: Julian Wilson
E-Mail: coolboyjulez@queensu.ca
Faculty: Commerce
View Member: [2](#)

Name: Jacqueline Craig
E-Mail: jcisgod@gmail.com
Faculty: Engineering
View Member: [3](#)

Name: John Stewart
E-Mail: fakeemail@hotmail.com
Faculty: Science
View Member: [4](#)

Name: Barack Obama
E-Mail: potus@gmail.com
Faculty: Arts
View Member: [5](#)

Figure 21: View members page for admins

Click on a member whose activity you want to examine or who you want to delete from the system by clicking on the relevant number. Suppose I click on '3' to check out Jacqueline's activity on the site:

Welcome, Administrator Julian
You are looking at the Jacqueline's profile

Delete Member

Consumer Activity:

Consumer Bookings:

Status: Denied Start Date: 2015-05-08 Address: 123 Maple Street

Status: Confirmed Start Date: 2016-02-28 Address: 12 Martin Lane

Status: Unconfirmed Start Date: 2016-09-06 Address: 136 Oak Drive

Status: Denied Start Date: 2016-09-23 Address: 1 Front Street

Status: Confirmed Start Date: 2016-11-02 Address: 10 Martin Lane

Consumer Rating:

Name: Jacqueline Craig
Address: 12 Wall Avenue
Rating: 4

Name: Jacqueline Craig
Address: 102 Martin Lane
Rating: 4

Name: Jacqueline Craig
Address: 105 Martin Lane
Rating: 4

Name: Jacqueline Craig
Address: 10 Martin Lane
Rating: 2

Cancel

Supply Activity:

Supply Booking:

Address: 10 Martin Lane
Booked By: Jacqueline Craig
Start Date: 2016-11-02
Status: Confirmed

Address: 10 Martin Lane
Booked By: John Stewart
Start Date: 2017-08-21
Status: Unconfirmed

Supply Rating:

Name: Barack2 Obama2
Address: 10 Martin Lane
Rating: 5

Name: Barack2 Obama2
Address: 10 Martin Lane
Rating: 4

Name: Jacqueline Craig
Address: 10 Martin Lane
Rating: 2

From here, you can see several things about the member including comments/ratings tenants have made on his/her properties as well as bookings he/she currently has confirmed/has to confirm/has denied. Furthermore, you can see bookings the member has made in properties and comments/ratings he/she has made for properties. You can delete the member by clicking the 'Delete member' button.

Points of Interest for Admins

Admins can also manage *points of interest* in the system. These are main attractions in Toronto. For instance, the C.N. tower, the waterfront, etc. When a user adds a new property in a district, that property is associated with these points of interest to help the consumer understand what he/she is getting in the property. To view this page, click 'Points of Interest' in the admin dashboard to be brought to this page:

Toggle Points of Interest

Current Points of Interest

District: Waterfront
POI: C.N. Tower

District: Waterfront
POI: Beach

District: Rosedale
POI: R.O.M.

District: Uptown
POI: Concert Hall

District: Chinatown
POI: Bell Movie Theatre

District: Midtown
POI: Honest Eds

Add Points of Interest

Point of Interest:

District: ▼

Figure 22: Admin points of interest page

From here, you can add points of interest for each district if new ones pop up as well as see all the active points of interest. To add new points of interest, simply type it in the 'Points of Interest:' box, select the corresponding district, and then click 'Add Point of Interest'.

Discussion

Here we present several issues that we encountered during the development of this application along with design and implementation decisions, technologies used, and possible improvements and extensions to our take on QBnB.

Problems encountered during development and solutions

Version control

One of the more prominent issues we had during development was version control. We had a team of three people and it was often difficult to manage versions of the project. Often we would have several different versions of the project ongoing at the same time with different people going in different directions. Our solution to this was to use GitHub to track for versions. There was a learning curve involved but once there, it was certainly easier to control version and manage changes.

Program architecture and workflow

Initially, we tried to use the Model View Controller architecture for our program. We know from class and online that this is one of the more prominent frameworks for web development. Unfortunately, we knew very little about it and were short on time so our solution was to use a basic object-oriented framework. For instance, we had a class that managed all our database transactions, a class to model each user and the things they can do, a class to manage different properties, etc. This worked well as it allowed us to separate our work: one of us could design the object abstraction, someone could work on the UX design in HTML, and someone could work on the documentation/organization of the code.

Old and deprecated PHP functions

When trying to learn all of the PHP and HTML required to create this application, we went to several tutorials on Moodle and online to learn the basics. Unfortunately, we ran into a lot of issues using old (and unsafe) functions and coding practices that we didn't know were actually insecure. This resulted us going over a lot of our code near the end to ensure that SQL queries were safely executed and connection objects were always closed, etc.

Learning PHP's session-oriented features

In order to keep track of who was logged in and which property was being viewed at any point in time we took advantage of PHP's 'SESSION' objects. This allowed the user to open a session with the server, perform any action he/she wanted to do, concurrently with other users; however, initially we were unfamiliar with the capabilities of the session and post variables. The main issue we encountered was a conflict when the admin signed in and viewed information about other members. When the admin signed in, we set the session's member id to the admin's. This resulted in some confusion of who was actually signed in and if he/she was an admin or not. We fixed this by checking for the admin rights early in the application flow.

Design and implementation decisions

Comments

We allowed each member to make a comment on any property because they could ask questions about properties that they were interested in booking. As well, they could reply to any comments someone else made to add into the conversation.

Dates Booking Conflicts

Using the assumption stated in the project outline that no bookings would conflict. The owner was the person who rejected the booking if they conflicted.

Technologies used

GitHub

We used the desktop version of GitHub to develop the QBnB website. This allowed our team to make sure that each update was compatible with what other people were doing. It allowed us to work productively while not physically beside each other. As well, we were able to monitor what we have done on the database thus far and keep track of everything. The repository we used was public and so it would have been better if it was private. Big learning curve.

XAMPP

We used XAMPP as our server to connect our PHP script to the database.

PHP

We used PHP to connect our web application to the database. This allowed us to dynamically show pages in our application. We created classes to keep track of members, properties and the database. PHP was used at the recommendation done during class. Since this was the first time any of our team has used PHP we didn't exploit many of its capabilities, next time we will.

MySQL

We used MySQL to connect to the database in Apache. This was done because we were familiar of how it worked through class. We were implementing fairly rigorous SQL but due to the incremental parts of the project, in the last phase we didn't create many new SQL statements.

Possible improvements and extensions

Application architecture and framework

Looking back, our framework for the program was minimal or non-existent. Despite having a decent object-oriented model, there was still clear interdependency between the workflow of all the members. There were often pauses in development and the general file organization was poor. Adapting the Model-View-Control architecture would have solved a lot of these issues.

Implementation on a web server

When we were accessing our server it was strictly on our local computers. To actually implement QBnB, we would certainly need to purchase a host server to store all our files and get a website domain like www.qbnb.com.

Bootstrap

Our application doesn't have any CSS implemented. If we were to implement a CSS wrapper in our web application, such as Bootstrap, it would make the user experience much better. Unfortunately, due to time constraints, we struggled with extending this part of the assignment.

Geo-locating in the application

Currently members have to enter the district that they are in. An extension to the project is that they type the address and then the district can be filled in automatically. This would need intergration with some sort of apps.

Payment

Currently QBnB doesn't support payments; integrating something like stripe into our website would allow people to make payments with a turnkey system. We assume in our system that the payment will occur in person when the tenant goes to the accommodation.

Password Encryption/Hashing

For this iteration we didn't hash the password. In order to create a better security for our application, we would need to encrypt and hash the password and then store the password instead.

Query Statements

These statements are all of the SQL required to navigate our system.

-- Create new user (1)

-- 1. Member table

```
INSERT INTO MEMBER (email, password, FName, LName, year, faculty, degree,
is_admin)
VALUES ('dummy@gmail.com', 'password', 'dummy1', 'dummy2', '2012',
'Computer Science', 'B.Sc', 0);
```

-- 2. Phone_number table

```
INSERT INTO phone_number (phone_number, member_id)
VALUES ('647-123-4567', (select member_id from member where email =
'dummy@gmail.com'));
```

-- Create new property (2)

-- 1. Property table

```
INSERT INTO PROPERTY (address, number_of_rooms, room_type, price,
owner_id, district_id)
VALUES ('107 Apple Street', 4, 'Apartment', 600, 3, (select district_id from
district where district_name = 'Waterfront'));
```

-- 2. Feature table

```
INSERT INTO FEATURE (feature, property_id)
VALUES ('Pool', (select property_id from property where address = '107
Apple Street'));
```

-- View all properties (address, room_type, price, number_of_rooms) (3)

```
select property.address, property.room_type, property.price,
property.number_of_rooms
from property
where is_deleted = 0;
```

-- Search for property with certain criteria (4)

```
select property.address, property.room_type, property.price,
property.number_of_rooms
from property
where room_type = 'Condo' and price < 1000 and
district_id = (select district_id from district where district_name = 'Waterfront')
and is_deleted = 0;
```

-- Reset search constraints on properties and redisplay (5)

```
select property.address, property.room_type, property.price,
property.number_of_rooms
from property
where is_deleted = 0;
```



```

-- Get all property information related to property_id '1' (6)
    -- 1. Show specific property information, owner information, district information (6_1)
        select property.address, property.room_type, property.number_of_rooms,
property.price,
                member.FName, member.LName,
                district.district_name, district.Street_1, district.Street_2
        from property
        inner join district on district.district_id = property.district_id
        inner join member on property.owner_id = member.member_id
        where property.property_id = '1'
        and property.is_deleted = 0;

    -- 2. Show all comments on that property (6_2)
        select comment.text, comment.Rating, comment.Date, member.FName,
member.LName
        from comment
        inner join property on property.property_id = comment.property_id
        inner join member on member.member_id =
comment.commenting_member_id
        where property.property_id = '1'
        and property.is_deleted = 0 and comment.is_deleted = 0;

    -- 3. Show all features on that property (6_3)
        select feature
        from feature
        inner join property on property.property_id = feature.feature_id
        where property.property_id='1'
        and property.is_deleted = 0;

    -- 4. Show all points of interest in the district (6_4)
        select points_of_interest.points_of_interest
        from points_of_interest
        inner join property on property.district_id = points_of_interest.district_id
        where property.property_id='1'
        and property.is_deleted = 0;

-- Comment on specific property (7)
    insert into comment (text, Rating, Date, commenting_member_id, property_id)
        values ('This place is excellent', 5, '2012-12-13', '2', '2');

-- View bookings with given member_id (the member's bookings) (8)
    select booking.status, booking.start_date, property.address
    from booking
    inner join property on property.property_id = booking.property_id

```

```
where booking.booking_member_id = '1'
```

```
-- View bookings for a given owner_id. Here the owner owns the property  
-- and is looking for pending bookings; (9)
```

```
select booking.start_date, property.address  
from booking  
inner join property on property.property_id = booking.property_id  
inner join member on property.owner_id = member_id  
where member_id = '1' and booking.status = 'Unconfirmed'  
and property.is_deleted = 0;
```

```
-- View information about property owned by a given owner_id. (10)
```

```
select property.address, property.number_of_rooms, property.room_type,  
property.price  
from property  
inner join member on property.owner_id = member.member_id  
where member_id = '1'  
and property.is_deleted = 0;
```

```
-- View details of property and bookings associated with it. (11_1)
```

```
select property.address, property.number_of_rooms, property.room_type,  
property.price  
from property  
where property_id = '1'
```

```
-- View the bookings associated with the owned property (11_2)
```

```
select booking.status, booking.start_date, member.FName, member.LName  
from booking inner join member on booking.booking_member_id = member.member_id  
where property_id = '1'
```

```
--View members as an admin (12)
```

```
select * from member
```

```
-- Delete a member (13_1)
```

```
update member  
set is_deleted = '1'  
where member_id = '5'
```

```
-- Delete all of that member's accomodations (13_2)
```

```
update property  
set is_deleted = '1'  
where owner_id = '5'
```

```
-- Delete all of that member's bookings (13_3)
```

```

update booking
  set is_deleted = '1'
  where booking_member_id = '5'

-- View properties as an admin (14)
select * from property

-- Delete an accomodation (15_1)
update property
  set is_deleted = '1'
  where property_id = '1'

-- Delete all of that accomodation's bookings (15_2)
update booking
  set is_deleted = '1'
  where property_id = '1'

-- Summarize bookings by accomodation (16_1)
select booking.status, booking.start_date, member.FName, member.LName
  from booking inner join member on booking.booking_member_id = member.member_id
  where property_id = '1'

-- Summarize ratings by accomodation (16_2)
select comment.Rating, member.FName, member.LName
  from comment inner join member on comment.commenting_member_id =
member.member_id
  where property_id = '1' and Rating is not null

-- Summarize bookings by supplier (17_1)
select booking.status, booking.start_date, member.FName, member.LName,
property.address
  from booking
  inner join member on booking.booking_member_id = member.member_id
  inner join property on booking.property_id = property.property_id
  where property.owner_id = '1'

-- Summarize ratings by supplier (17_2)
select comment.Rating, member.FName, member.LName, property.address
  from comment
  inner join member on comment.commenting_member_id = member.member_id
  inner join property on comment.property_id = property.property_id
  where property.owner_id = '1' and Rating is not null

-- Summarize bookings by consumer (18)

```

```

select booking.status, booking.start_date, property.address
  from booking
 inner join property on booking.property_id = property.property_id
 where booking_member_id = '2'

-- Update member profile (19)
update member
  set email = 'dummy2@gmail.com', password = 'pppassword', FName = 'Bob', LName =
'dummy2', year = '2012', faculty = 'Computer Science', degree = 'B.Sc'
  where member_id = '6'

-- Admins add a point of interest (20)
insert into Points_Of_Interest (points_of_interest, district_id)
  VALUES ('Armstrong Pond', 1);

-- See all dates that a property is booked for (21)
select booking.start_date
  from booking
  where property_id = '3' and status = 'Confirmed' and is_deleted = '0'

-- Cancel a booking (22)
update booking
  set booking.status = 'Cancelled'
  where booking_id = 5

-- Member 1 deleting his own comment with 'id' 1 (23)
update comment
  set is_deleted = 1
  where comment_id = '1';

-- Member 1 deleting his own profile (24)
update member
  set is_deleted = 1
  where member_id = '1';

-- Member 1 updating his property profile (address, number of rooms, room type, price,
district) (25)
update property
  set address = '107 Maple Street', number_of_rooms = '3', room_type = 'House',
price = 1000, district_id = '1'
  where property_id = '1';

-- Member 1 making a booking (given property_id and member_id) (26)
insert into booking (status, start_date, booking_member_id, property_id)

```

values ('UNCONFIRMED', 2012-12-23, booking_member_id = '1', property_id = '1')

--Member 1 logging in (given email and password) (27)

select * from member where member_id = '1'

--Booking denied (28)

UPDATE `Booking` SET `status`='Denied' WHERE booking_id = '2'

--Booking Confirmed

UPDATE `Booking` SET `status` = 'Confirmed' where booking_id = 3

Query Outputs

These outputs are examples of queries that members or admins can do.

Query 1 Part 1

Table structure for table member

Column	Type	Null	Default
member_id	int(11)	No	
email	varchar(45)	No	
password	varchar(45)	No	
FName	char(20)	No	
LName	char(20)	No	
year	int(11)	No	
faculty	varchar(45)	No	
degree	varchar(15)	No	
is_admin	tinyint(1)	No	
is_deleted	tinyint(1)	No	

Dumping data for table member

1	jshimz@hotmail.com	itsjustshim	Justin	Shimkovitz	2016 Law	Criminal Law	1	0
2	coolboyjulez@queensu.ca	jwill	Julian	Wilson	2016 Commerce	Finance	1	0
3	jcisgod@gmail.com	jacjacjac	Jacqueline	Craig	2016 Engineering	Applied Math	1	0
4	fakeemail@hotmail.com	animals	John	Stewart	2018 Science	Biology	0	0
5	potus@gmail.com	password	Barack	Obama	2019 Arts	English	0	0
6	dummy@gmail.com	password	dummy1	dummy2	2012 Computer Science	B.Sc	0	0

Query 1 Part 2

Table structure for table phone_number

Column	Type	Null	Default
phone_number	char(13)	No	
member_id	int(11)	No	

Dumping data for table phone_number

416-543-1233	1
416-543-1234	1
905-881-4432	1
789-087-5432	2
123-456-7867	3
818-987-4324	3
453-235-4433	4
323-655-6666	5
647-123-4567	6

Query 2 Part 1

Table structure for table property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	

number_of_rooms	int(11)	No
room_type	varchar(45)	No
price	int(11)	No
is_deleted	tinyint(1)	No
owner_id	int(11)	No
district_id	int(11)	No

Dumping data for table property

1	123 Maple Street	4 House	600	0	1	1
2	136 Oak Drive	3 House	200	0	1	4
3	1 Front Street	1 Apartment	1000	0	2	2
4	12 Wall Avenue	2 Condo	750	0	5	2
5	10 Martin Lane	2 Apartment	500	0	3	4
7	107 Apple Street	4 Apartment	600	0	3	2

Query 2 Part 2

Table structure for table feature

Column	Type	Null	Default
feature_id	int(11)	No	
feature	varchar(45)	No	
property_id	int(11)	No	

Dumping data for table feature

1	2 Bathrooms	1
2	Big Backyard	3
3	Pool	4
4	Pool	2
5	Queen Beds	1
6	Pool	7

Query 3

Table structure for table property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table property

123 Maple Street	House	600	4
136 Oak Drive	House	200	3
1 Front Street	Apartment	1000	1

12 Wall Avenue	Condo	750	2
10 Martin Lane	Apartment	500	2
107 Apple Street	Apartment	600	4

Query 4

Table structure for table property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table property

12 Wall Avenue	Condo	750	2
----------------	-------	-----	---

Query 5

Table structure for table property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table property

123 Maple Street	House	600	4
136 Oak Drive	House	200	3
1 Front Street	Apartment	1000	1
12 Wall Avenue	Condo	750	2
10 Martin Lane	Apartment	500	2
107 Apple Street	Apartment	600	4

Query 6 Part 1

Dumping data for table property

123 Maple Street	House	4	600	Justin Shimkovitz	Uptown	Bathurst Steeles
------------------	-------	---	-----	-------------------	--------	------------------

Query 6 Part 2

Dumping data for table comment

Great place! Slightly dirty, but nothing unmanageable 4 2015-05-05 Barack Obama
Thanks for your feedback! I'll make sure to get that fixed NULL 2015-05-06 Justin Shimkovitz

Query 6 Part 3

Table structure for table feature

Column	Type	Null	Default
feature_id	int(11)	No	
feature	varchar(45)	No	
property_id	int(11)	No	

Dumping data for table feature

2 Bathrooms

Query 6 Part 4

Table structure for table points_of_interest

Column	Type	Null	Default
points_of_interest	varchar(45)	No	
district_id	int(11)	No	

Dumping data for table points_of_interest

Concert Hall

Query 7

Table structure for table Comment

Column	Type	Null	Default
comment_id	int(11)	No	
text	varchar(140)	No	
Rating	int(11)	Yes	NULL
Date	date	No	
is_deleted	tinyint(1)	No	
commenting_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Comment

1	Great place! Slightly dirty, but nothing unmanageable	4	2015-05-05	0	5	1
2	Thanks for your feedback! I'll make sure to get that fixed	NULL	2015-05-06	0	1	1
3	Terrible. Really terrifying area of town and nothing near it worth seeing. Would not recommend.	1	2016-01-03	0	4	3
6	This place is excellent	5	2012-12-23	0	2	2

Query 8

Dumping data for table booking

Confirmed 2015-08-04 1 Front Street

Query 9

Dumping data for table booking

2015-05-09 136 Oak Drive

Query 10

Table structure for table property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table property

123 Maple Street 4 House 600

136 Oak Drive 3 House 200

Query 11 Part 1

Table structure for table property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table property

123 Maple Street 4 House 600

Query 11 Part 2

Dumping data for table booking

Confirmed 2015-05-06 Julian Wilson

Denied 2015-05-08 Jacqueline Craig

Query 12

Table structure for table member

Column	Type	Null	Default
member_id	int(11)	No	
email	varchar(45)	No	
password	varchar(45)	No	

FName	char(20)	No
LName	char(20)	No
year	int(11)	No
faculty	varchar(45)	No
degree	varchar(15)	No
is_admin	tinyint(1)	No
is_deleted	tinyint(1)	No

Dumping data for table member

1	jshimz@hotmail.com	itsjustshim	Justin	Shimkovitz	2016 Law	Criminal Law	1	0
2	coolboyjulez@queensu.ca	jwill	Julian	Wilson	2016 Commerce	Finance	1	0
3	jcisgod@gmail.com	jacjacjac	Jacqueline	Craig	2016 Engineering	Applied Math	1	0
4	fakeemail@hotmail.com	animals	John	Stewart	2018 Science	Biology	0	0
5	potus@gmail.com	password	Barack	Obama	2019 Arts	English	0	0
6	dummy@gmail.com	password	dummy1	dummy2	2012 Computer Science	B.Sc	0	0

Query 13 Part 1

Table structure for table Member

Column	Type	Null	Default
member_id	int(11)	No	
email	varchar(45)	No	
password	varchar(45)	No	
FName	char(20)	No	
LName	char(20)	No	
year	int(11)	No	
faculty	varchar(45)	No	
degree	varchar(15)	No	
is_admin	tinyint(1)	No	
is_deleted	tinyint(1)	No	

Dumping data for table Member

1	jshimz@hotmail.com	itsjustshim	Justin	Shimkovitz	2016 Law	Criminal Law	1	0
2	coolboyjulez@queensu.ca	jwill	Julian	Wilson	2016 Commerce	Finance	1	0
3	jcisgod@gmail.com	jacjacjac	Jacqueline	Craig	2016 Engineering	Applied Math	1	0
4	fakeemail@hotmail.com	animals	John	Stewart	2018 Science	Biology	0	0
5	potus@gmail.com	password	Barack	Obama	2019 Arts	English	0	1
6	dummy@gmail.com	password	dummy1	dummy2	2012 Computer Science	B.Sc	0	0

Query 13 Part 2

Table structure for table Property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	

room_type	varchar(45)	No
price	int(11)	No
is_deleted	tinyint(1)	No
owner_id	int(11)	No
district_id	int(11)	No

Dumping data for table Property

1	123 Maple Street 4 House	600	0	1	1
2	136 Oak Drive 3 House	200	0	1	4
3	1 Front Street 1 Apartment	1000	0	2	2
4	12 Wall Avenue 2 Condo	750	1	5	2
5	10 Martin Lane 2 Apartment	500	0	3	4
6	107 Apple Street 4 Apartment	600	0	3	2

Query 13 Part 3

Table structure for table Booking

Column	Type	Null	Default
booking_id	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Booking

1	Confirmed	2015-05-06	0	2	1
2	Deleted	2015-05-06	0	2	2
3	Denied	2015-05-08	0	3	1
4	Unconfirmed	2015-05-09	0	4	2
5	Confirmed	2015-08-04	0	1	3

Query 14

Table structure for table property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table property

1	123 Maple Street 4 House	600	0	1	1
2	136 Oak Drive 3 House	200	0	1	4

3	1 Front Street	1 Apartment	1000	0	2	2
4	12 Wall Avenue	2 Condo	750	1	5	2
5	10 Martin Lane	2 Apartment	500	0	3	4
6	107 Apple Street	4 Apartment	600	0	3	2

Query 15 Part 1

Table structure for table Property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	
district_id	int(11)	No	

Dumping data for table Property

1	123 Maple Street	4 House	600	1	1	1
2	136 Oak Drive	3 House	200	0	1	4
3	1 Front Street	1 Apartment	1000	0	2	2
4	12 Wall Avenue	2 Condo	750	1	5	2
5	10 Martin Lane	2 Apartment	500	0	3	4
6	107 Apple Street	4 Apartment	600	0	3	2

Query 15 Part 2

Table structure for table Booking

Column	Type	Null	Default
booking_id	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Booking

1	Confirmed	2015-05-06	1	2	1
2	Deleted	2015-05-06	0	2	2
3	Denied	2015-05-08	1	3	1
4	Unconfirmed	2015-05-09	0	4	2
5	Confirmed	2015-08-04	0	1	3

Query 16 Part 1

Dumping data for table booking

Confirmed	2015-05-06	Julian	Wilson
-----------	------------	--------	--------

Denied 2015-05-08 Jacqueline Craig

Query 16 Part 2

Dumping data for table comment

4 Barack Obama

Query 17 Part 1

Dumping data for table comment

4 Barack Obama

Query 17 Part 2

Dumping data for table comment

4 Barack Obama 123 Maple Street

5 Julian Wilson 136 Oak Drive

Query 18

Dumping data for table booking

Confirmed 2015-05-06 123 Maple Street

Deleted 2015-05-06 136 Oak Drive

Query 19

Table structure for table Member

Column	Type	Null	Default
member_id	int(11)	No	
email	varchar(45)	No	
password	varchar(45)	No	
FName	char(20)	No	
LName	char(20)	No	
year	int(11)	No	
faculty	varchar(45)	No	
degree	varchar(15)	No	
is_admin	tinyint(1)	No	
is_deleted	tinyint(1)	No	

Dumping data for table Member

1	jshimz@hotmail.com	itsjustshim	Justin	Shimkovitz	2016 Law	Criminal Law	1	0
2	coolboyjulez@queensu.ca	jwill	Julian	Wilson	2016 Commerce	Finance	1	0
3	jcisgod@gmail.com	jacjacjac	Jacqueline	Craig	2016 Engineering	Applied Math	1	0
4	fakeemail@hotmail.com	animals	John	Stewart	2018 Science	Biology	0	0
5	potus@gmail.com	password	Barack	Obama	2019 Arts	English	0	1
6	dummy2@gmail.com	pppassword	Bob	dummy2	2012 Computer Science	B.Sc	0	0

Query 20

Table structure for table Points_Of_Interest

Column	Type	Null	Default
--------	------	------	---------

points_of_interest varchar(45) No

district_id int(11) No

Dumping data for table Points_Of_Interest

C.N. Tower 2

Beach 2

R.O.M. 3

Concert Hall 1

Bell Movie Theatre 4

Armstrong Pond 1

Query 21

Table structure for table booking

Column	Type	Null	Default
booking_id	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table booking

2015-08-04

Query 22

Table structure for table Booking

Column	Type	Null	Default
booking_id	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Booking

1 Confirmed 2015-05-06 1 2 1

2 Deleted 2015-05-06 0 2 2

3 Denied 2015-05-08 1 3 1

4 Unconfirmed 2015-05-09 0 4 2

5 Cancelled 2015-08-04 0 1 3

Query 23

Table structure for table Comment

Column	Type	Null	Default
comment_id	int(11)	No	
text	varchar(140)	No	

Rating	int(11)	Yes NULL
Date	date	No
is_deleted	tinyint(1)	No
commenting_member_id	int(11)	No
property_id	int(11)	No

Dumping data for table Comment

1 Great place! Slightly dirty, but nothing unmanageable	4	2015-05-05	1	5	1
2 Thanks for your feedback! I'll make sure to get that fixed	NULL	2015-05-06	0	1	1
3 Terrible. Really terrifying area of town and nothing near it worth seeing. Would not recommend.	1	2016-01-03	0	4	3
6 This place is excellent	5	2012-12-23	0	2	2

Query 24

Table structure for table Member

Column	Type	Null	Default
member_id	int(11)	No	
email	varchar(45)	No	
password	varchar(45)	No	
FName	char(20)	No	
LName	char(20)	No	
year	int(11)	No	
faculty	varchar(45)	No	
degree	varchar(15)	No	
is_admin	tinyint(1)	No	
is_deleted	tinyint(1)	No	

Dumping data for table Member

1 jshimz@hotmail.com	itsjustshim	Justin	Shimkovitz	2016 Law	Criminal Law	1	1
2 coolboyjulez@queensu.ca	jwill	Julian	Wilson	2016 Commerce	Finance	1	0
3 jcisgod@gmail.com	jacjacjac	Jacqueline	Craig	2016 Engineering	Applied Math	1	0
4 fakeemail@hotmail.com	animals	John	Stewart	2018 Science	Biology	0	0
5 potus@gmail.com	password	Barack	Obama	2019 Arts	English	0	1
6 dummy2@gmail.com	ppppassword	Bob	dummy2	2012 Computer Science	B.Sc	0	0

Query 25

Table structure for table Property

Column	Type	Null	Default
property_id	int(11)	No	
address	varchar(45)	No	
number_of_rooms	int(11)	No	
room_type	varchar(45)	No	
price	int(11)	No	
is_deleted	tinyint(1)	No	
owner_id	int(11)	No	

district_id int(11) No

Dumping data for table Property

1	107 Maple Street	3 House	1000	1	1	1
2	136 Oak Drive	3 House	200	0	1	4
3	1 Front Street	1 Apartment	1000	0	2	2
4	12 Wall Avenue	2 Condo	750	1	5	2
5	10 Martin Lane	2 Apartment	500	0	3	4
6	107 Apple Street	4 Apartment	600	0	3	2

Query 26

Table structure for table Booking

Column	Type	Null	Default
booking_id	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Booking

1	Confirmed	2015-05-06	1	2	1
2	Deleted	2015-05-06	0	2	2
3	Denied	2015-05-08	1	3	1
4	Unconfirmed	2015-05-09	0	4	2
5	Cancelled	2015-08-04	0	1	3
6	UNCONFIRMED	2012-12-23	0	1	1

Query 27

Table structure for table member

Column	Type	Null	Default
member_id	int(11)	No	
email	varchar(45)	No	
password	varchar(45)	No	
FName	char(20)	No	
LName	char(20)	No	
year	int(11)	No	
faculty	varchar(45)	No	
degree	varchar(15)	No	
is_admin	tinyint(1)	No	
is_deleted	tinyint(1)	No	

Dumping data for table member

1	jshimz@hotmail.com	itsjustshim	Justin Shimkovitz	2016 Law Criminal Law	1	0
---	--------------------	-------------	-------------------	-----------------------	---	---

Query 28

Table structure for table Booking

Column	Type	Null	Default
booking_id	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Booking

```
1 Confirmed 2015-05-06 0 2 1
2 Denied    2015-05-06 0 2 2
3 Denied    2015-05-08 0 3 1
4 Denied    2015-05-09 0 4 2
5 Confirmed 2015-08-04 0 1 3
```

Query 29

Table structure for table Booking

Column	Type	Null	Default
booking_id	int(11)	No	
status	varchar(45)	No	
start_date	date	No	
is_deleted	tinyint(1)	No	
booking_member_id	int(11)	No	
property_id	int(11)	No	

Dumping data for table Booking

```
1 Confirmed 2015-05-06 0 2 1
2 Denied    2015-05-06 0 2 2
3 Confirmed 2015-05-08 0 3 1
4 Denied    2015-05-09 0 4 2
5 Confirmed 2015-08-04 0 1 3
```