

Customers

<u>S/N</u>	<u>Use Case</u>	<u>PseudoCode</u>
1	Login	Hash the password before passing back the password field
	Logout	
	Register	Hash the password before passing back the password field
<i>Customer who looks for an office space</i>		
	Search Listing	1. Type in keywords 2. Check filters 3. Submit 4. Display Relevant Listings
	View a Listing	1. Click a listing from a list of listings
	Make a booking	1. From that listing, choose a slot. 2. Confirm booking 3. Make payment with paypal. 4. Booking receipt. Create booking, associate to customer and slot. Add booking to customer and assign to slot. Not allowed when the slot is available
	Update a booking	<pre>makeABooking(Long cld) { Listing l = ListingManager.CreateListing() customer = CustomerManager.addAListing(cld, Listing); ListingManager.assignACustomer(customer); }</pre>
	View all booking(completed)	Click on booking and display all bookings
	View a booking	Click a booking from a list of bookings
	Cancel a booking	1. Confirmation pop-up box
<i>Customer who offer spaces</i>		
	Make a listing	<u>UseCaseSession</u> <pre>makeAListing(Long cld) { Listing l = ListingManager.CreateListing() customer = CustomerManager.addAListing(cld, Listing); ListingManager.assignACustomer(customer); }</pre>

		Return customer }/customer to have Fetch.EAGER?
	Update a listing	Only update those that which are not associated to booking
	Remove a listing	Remove Remove listing from customer and disassociate listing from customer. Not allowed when there is a booking

Staff

<u>S/N</u>	<u>Use Case</u>	<u>Description</u>
1	Login	
	Logout	

HostDoorSystem

<u>S/N</u>	<u>Use Case</u>	<u>Description</u>
1	Login	
2	Logout	
3	Verify QR Code	
<i>Raspberry Hardware Methods</i>		
	Activate camera with button press	
	Light Up LED	
	Unlock Door with Servo	
	Lock Door with Servo	

Specialize Managers

Managers who handle use cases:

//one who handles all the use cases done by customer on dashboard such as PostListing
Customer Dashboard manager

postListing: Boolean

- createListing()
- customer.addListing()
- listing.addCustomer()

placeABooking: Boolean

- createBooking
- customer.addBooking
- booking.addCustomer()

....

Object CRUD Manager

CustomerManager (stateless)

```
public void createCustomer(email, password)
public Customer retrieveCustomer(id)
public Customer retrieveCustomerByEmail(String email)
public List<Customer> retrieveAllCustomers()
public void updateCustomer (email, password)
public void deleteCustomer(cid)
public void addListing(Long cld, buildingType, (ListingDetails param), (Location params))
public void removeListing(Long cld, Listing lld)
public void addBooking(Long cld, (Booking param))
public void removeBooking(Long cld, Booking bld)
public void addFeedback(Long cld, (Feedback params))
public void removeFeedback(Long cld, Feedback fld)
```

AdminManager (stateless)

```
public void createAdmin(Admin admin)
public Admin retrieveAdmin(Long ald)
public Admin retrieveAdminByEmail(String email)
public Admin retrieveAllAdmin()
public void updateAdmin(Admin admin)
public void deleteAdmin(Long id)
```

HostDoorSystemManager (stateless)

```
public void createDoorAccount(HostDoorSystem hostDoorSystem)
public HostDoorSystem retrieveDoorAccount(Long hld)
public HostDoorSystem retrieveDoorAccountByHost(Customer customer)
public List<HostDoorSystem> retrieveAllDoorAccount()
public void updateDoorAccount(HostDoorSystem hostDoorSystem)
public void deleteDoorAccount(Long hld)
```

ListingManager (stateless)

```
public Listing createListing(Listing listing)
public List<Listing> retrieveAllListing()
public List<Listing> retrieveListingByDoorId(HostDoorSystem hostDoorSystem)
public List<Listing> retrieveListingByCustomer(Customer customer)
public List<Listing> retrieveListingByFields(DateTime startDate, DateTime endDate, Location
location, ListingDetail listingDetail, Feedback feedback, Booking booking)
public void updateListing(Listing listing)
public void deleteListing(Long lld)
```

BookingManager (stateless)

```
public void createBooking(Booking booking)
public Booking retrieveBooking(Long bld)
public List<Booking> retrieveAllBooking()
public Booking retrieveBookingByField(Customer customer, Location location, DateTime
startDate, DateTime endDate, Enum type, Listing listing)
public void updateBooking(Booking booking)
public void deleteBooking(Long bld)
```

TransactionManager (stateless)

```
public void createTransaction(Transaction transaction)
public Transaction retrieveTransaction(Transaction transaction)
public List<Transaction> retrieveAllTransaction()
public List<Transaction> retrieveTransactionByCustomer(Customer customer)
public void updateTransaction(Transaction transaction)
public void deleteTransaction( Transaction transaction)
```

FeedbackManager (stateless)

```
public void createFeedback(Transaction transaction)
public Transaction retrieveFeedback(Transaction transaction)
```

```
public List<Feedback> retrieveAllFeedback()
public List<Feedback> retrieveFeedbackByCustomer(Customer customer)
public List<Feedback> retrieveFeedbackByListing(Listing listing)
public void updateFeedback(Feedback feedback)
public void deleteFeedback(Feedback feedback)
```

LocationManager (stateless)

```
public void createLocation(Location location)
public Location retrieveLocation(Long lId)
public void updateLocation(Location location)
public void deleteLocation(Long lId)
```

LocationDetailManager (stateless)

```
public void createListingDetail(ListingDetail listingDetail);

public ListingDetail retrieveListingDetail(Long lId);

public void updateListingDetail(ListingDetail listingDetail);

public void deleteListingDetail(Long lId);
```

ProfileManager (stateless)

```
public void createProfile(Profile profile)
public Profile retrieveProfile(Long pId)
public void updateProfile(Profile profile)
public void deleteProfile(Long pId)
```