



AlunnoCasa

(Student homes)

Group 1-

| | |
|-----------------------|-------------|
| Aishwarya Nandapurkar | (avn160230) |
| Deepanshi Puri | (dxp174930) |
| Dhairya Pawar | (dxp170000) |
| Pawan Dandekar | (ppd170230) |

Table of Contents:

| | |
|--|-------|
| 1. AllunoCasa Background: | 2 |
| 2. Problems: | 3 |
| 3. Solution: | 3 |
| 4. Process: | 4 |
| 5. Tables: | 5 |
| 6. Entity Relationship Diagram: | 6 |
| 7. Relational Database Schema: | 7 |
| 8. Create Table Statements: | 8-11 |
| 9. Main Menu and Data Input Screen Form: | 12-20 |
| 10. Sample Reports: | 21-24 |
| 11. Contributions: | 25 |



Background

Hassles of reservations or bookings is something that everyone of us students faced when we came to the United States. Right from booking a temporary accommodation for ourselves before moving into our leased-out apartments, to arranging holiday stays in different locations, not to forget times when people have their loved ones come over for their convocation, figuring out the places for short stays is and was always a problem. This made us come up with the idea of “Alunnocasa”, which will be a student-community based Airbnb. This will literally be by the students, for the students, of the students! It will help international as well as local students. Just for instance, The University of Texas, Dallas has a huge number of undergraduate and graduate students coming in every year. Almost 50% of which are international students. We can now have an idea of how many students study in the country and how this student base will keep on increasing. So, if we consider the length and breadth of the student Airbnb concept, it will be almost impossible to keep a track of our data without databases. Excel sheets might be an alternative but not something that will suffice the ever-growing amount of data that will be coming in.

The idea is great to implement but cannot be done just by calling guests to book their apartments. There will have to be a process in place and data would have to be stored just so it can be accessed and processed when required. Database seems like a great option to do this and the herculean task of keeping track of all the small details related to every user/ accommodation will then seem reasonable.

Problems:

- Keeping track of multiple students who need a temporary accommodation in peak seasons such as start of the semester or vacations, proves to be a tough job for the company as the data is not managed centrally.
- As the data is stored in multiple lists and files modification to these entries is a manual, time consuming and taxing task. For example, one person leaves the accommodation and another person starts his or her stay then to update these multiple entries of multiple students is a very complicated job.
- Data entered in each field can also vary because data format inconsistency. E.g. Dates could be entered as 12-01-2014, 12/01/2014 or Dec-01-2014.
- Storing data in excel lists were causing problems like
 - Multiple themes - For customer details, there will be multiple entries related to host, location of the property, amenities of the property etc.
 - Redundancy of data - Same information about the customer is already included into property details and host details etc.
- Targeted analysis of the data or querying data from excel is also a complex job. E.g. to find a revenue for last 7 days for a particular location needed an extensive work to be done on the excel file.
- As the data is not stored in an organized way and at one central location, it would further create problems while migrating the data to third party analysis tools.

Solution:

- This project will help the company to easily track and handle multiple entries from multiple students for temporary accommodations.
- It will also efficiently handle modification to these entries and will help the company to provide the service in more effective way.
- Issues like list modification, data uniformity, redundancy and multi- themes list will be solved
- Better analysis of the data would help to understand the student behaviour and will help to target the student demographic more precisely.

Process:

1. The user has to create a profile on the company's website using the official school email id either as a host or a guest.
2. The user could be either host(*Host-User*), guest(*Guest-User*) or both. But for 1 complete transaction, he/she could be only Host-User or Guest-User.
3. Once the account is created, the user needs to log in as Host-User or Guest-User for further processing.
4. The Host-User has to register the property on company website with property details. The required property details would be - 'Type of Property', 'Maximum Guests Allowed', 'Number of Bedrooms and Bathrooms', 'Rent of the Property', 'Its Availability' etc.
5. Host-User can also mention the details of amenities such as swimming pool, gym, garage etc.
6. The Guest-User could search the property on company website. He/she could search a property using fields such as 'The required date', 'Type of Property', 'Number of Bedrooms and Bathrooms', 'Rent of the Property' etc. He/she can also filter for the amenities such as swimming pool, gym, garage etc.
7. If the Guest-User finds the desired property then he/she will continue with the booking procedure and then will follow up by making the payment to the company.
 - ❖ Cancellation policy - 48 hours prior to your booking, the Guest-User will get the entire amount refunded else nothing will be refunded.
8. After completing the stay, once the Guest-User checks out, company issues the payment to the host after deducting 20% of the amount.
9. In the end, both the Guest-User and the Host-User will rate each other on the company's website.

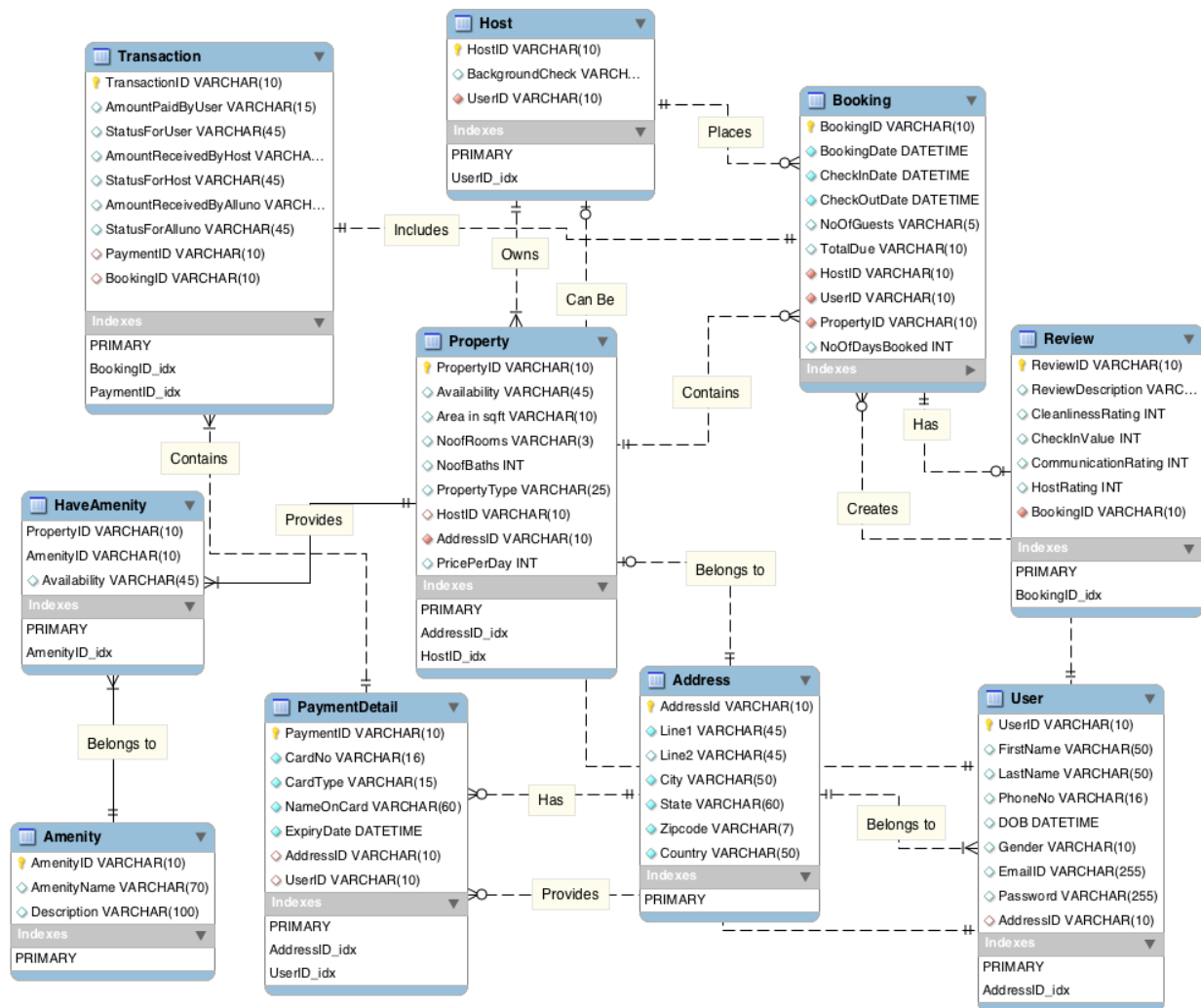
Tables:

According to initial analysis below tables will require to store and present data.

1. **User** - This table includes the details of a person who registered on the website as a user. Information such as name, email address, password, gender etc. Table Host and Guest will link user (created above) to either as a host, guest or both.
2. **Host** - This table link user details as a host with host ID.
3. **Address** - This table will contain the information of property or a user such as street address, city, state and Zipcode. This table will link to user table, property table and payment detail table via foreign key as a AddressID.
4. **Booking** - This table will contain details of a booking done such as requested check-in and check-out date, booking date, amount due etc.
5. **Property** - This table will include details of a property of host such as type of property type, price per day, number of bathrooms and bedrooms, availability and area.
6. **Amenity** - This includes the amenities in each property.
7. **HaveAmenity** - This is an association table having the property id and amenity id along with availability.
8. **Payment Detail** - Payment information will be included in this table such as card no and other card details.
9. **Transaction** - This table would include all transactions like amount paid by user, status for user, amount received by host etc.
10. **Review** - Review of a user will be captured in this table and it will be assigned to each property.

** Property table has a many to many relationship with the Amenity table and thus we are making an association table for it. Also, we are not including amenity as an attribute in the property table because it can act like a multi-valued attribute.

Entity-Relationship Diagram:



Relational Database Schema:

Address: (AddressID, Line1, Line2, City, State, Zipcode, Country)

Amenity: (AmenityId, AmenityName, Description)

User: (UserID, FirstName, LastName, PhoneNo, DOB, Gender, EmailID, Password, *AddressID*)

Host: (HostID, BackgroundCheck, *UserID*)

Property: (PropertyID, Availability, Area in sqft, NoofRooms, NoofBaths, PropertyType, *HostID*, *AddressID*, PricePerDay)

HaveAmenity: (*PropertyID*, *AmenityID*, Availability)

Payment Details: (PaymentID, CardNo, CardType, NameOnCard, ExpiryDate, *AddressID*, *UserID*)

Transaction: (TransactionID, AmountPaidByUser, StatusForUser, AmountReceivedByHost, StatusForHost, AmountReceivedByAlluno, StatusForAlluno, *PaymentID*, *BookingID*)

Booking: (BookingID, BookingDate, CheckInDate, CheckOutDate, NoOfGuests, TotalDue, *HostID*, *UserID*, *PropertyID*, NoOfDaysBooked)

Review: (ReviewID, ReviewDescription, CleanlinessRating, CheckInValue, CommunicationRating, HostRating, *BookingID*)

Create Table Statements:

CREATE TABLE Address (

```
AddressID VARCHAR (10) NOT NULL PRIMARY KEY,  
Line1     VARCHAR (45) NOT NULL ,  
Line2     VARCHAR (45),  
City      VARCHAR (50) NOT NULL,  
State     VARCHAR (60) NOT NULL,  
Zipcode   VARCHAR (7) NOT NULL,  
Country   VARCHAR (50) NOT NULL)
```

CREATE TABLE USER(

```
UserID   VARCHAR(10) NOT NULL PRIMARY KEY,  
FirstName VARCHAR(50) ,  
LastName  VARCHAR(50),  
PhoneNo   VARCHAR(16),  
DOB       VARCHAR(45),  
Gender    VARCHAR(10),  
EmailID   VARCHAR(255),  
Password  VARCHAR(255),  
AddressID VARCHAR(10),  
CONSTRAINT FKAddressID FOREIGN KEY (AddressID) REFERENCES Address  
ON UPDATE CASCADE)
```

Create Table Host(

```
HostID VARCHAR(10) NOT NULL PRIMARY KEY,  
BackgroundCheck VARCHAR(45) ,  
UserID VARCHAR(10) NOT NULL,  
CONSTRAINT FKUserID FOREIGN KEY (UserID)  
REFERENCES User ON UPDATE CASCADE)
```

CREATE TABLE Property(

```
PropertyID  VARCHAR(10) NOT NULL PRIMARY KEY,  
Availability VARCHAR(45) ,  
AreaInSqft  VARCHAR(10),  
NoOfRooms   VARCHAR(3),  
NoOfBaths   INTEGER,  
PropertyType VARCHAR(25),  
HostID       VARCHAR(10),  
AddressID    VARCHAR(10) NOT NULL,  
PricePerDay  INTEGER,  
CONSTRAINT FK_AddressID FOREIGN KEY (AddressID) REFERENCES Address ON UPDATE CASCADE,  
CONSTRAINT FKHostID FOREIGN KEY (HostID) REFERENCES Host ON UPDATE CASCADE)
```

CREATE TABLE BOOKING(

```
BookingID  VARCHAR(10) NOT NULL PRIMARY KEY,  
BookingDate DATE NOT NULL,  
CheckInDate DATE NOT NULL,  
CheckOutDate DATE NOT NULL,  
NoOfGuests VARCHAR(5),  
TotalDue   VARCHAR(10),  
HostID     VARCHAR(10) NOT NULL,  
UserID     VARCHAR(10) NOT NULL,  
PropertyID VARCHAR(10) NOT NULL,  
NoOfDaysBooked INTEGER ,  
CONSTRAINT FK_UserID FOREIGN KEY (UserID) REFERENCES User ON UPDATE CASCADE,  
CONSTRAINT FK_HostID FOREIGN KEY (HostID) REFERENCES Host ON UPDATE CASCADE,  
CONSTRAINT FK_PropertyID FOREIGN KEY (PropertyID) REFERENCES Property ON UPDATE CASCADE)
```

CREATE TABLE REVIEW(


ReviewID VARCHAR(10) NOT NULL PRIMARY KEY,
ReviewDescription VARCHAR(100) ,
CleanlinessRating INTEGER,
CheckInValue INTEGER,
CommunicationRating INTEGER,
HostRating INTEGER,
BookingID VARCHAR(10) NOT NULL,
CONSTRAINT FK_BookingID FOREIGN KEY (BookingID) REFERENCES Booking ON UPDATE CASCADE)

CREATE TABLE PAYMENTDETAIL(

PaymentID VARCHAR(10) NOT NULL PRIMARY KEY,
CardNo VARCHAR(16) NOT NULL,
CardType VARCHAR(15) NOT NULL,
NameOnCard VARCHAR(60) NOT NULL,
ExpiryDate DATE NOT NULL,
AddressID VARCHAR(10) ,
UserID VARCHAR(10),
CONSTRAINT FK__AddressID FOREIGN KEY (AddressID) REFERENCES Address ON UPDATE CASCADE,
CONSTRAINT FK__UserID FOREIGN KEY (UserID) REFERENCES User ON UPDATE CASCADE)

CREATE TABLE Transaction(

TransactionID VARCHAR(10) NOT NULL PRIMARY KEY,
AmountPaidByUser VARCHAR(15) ,
StatusForUser VARCHAR(45) ,
AmountReceivedByHost VARCHAR(15) ,
StatusForHost VARCHAR(45),
AmountReceivedByAlluno VARCHAR(15),



```
StatusForAlluno VARCHAR(45),  
PaymentID VARCHAR(10) ,  
BookingID VARCHAR(10),  
CONSTRAINT FKPaymentID FOREIGN KEY (PaymentID) REFERENCES PaymentDetail ON UPDATE  
CASCADE,  
CONSTRAINT FK__BookingId FOREIGN KEY (BookingID) REFERENCES Booking ON UPDATE CASCADE)
```

CREATE TABLE Amenity (

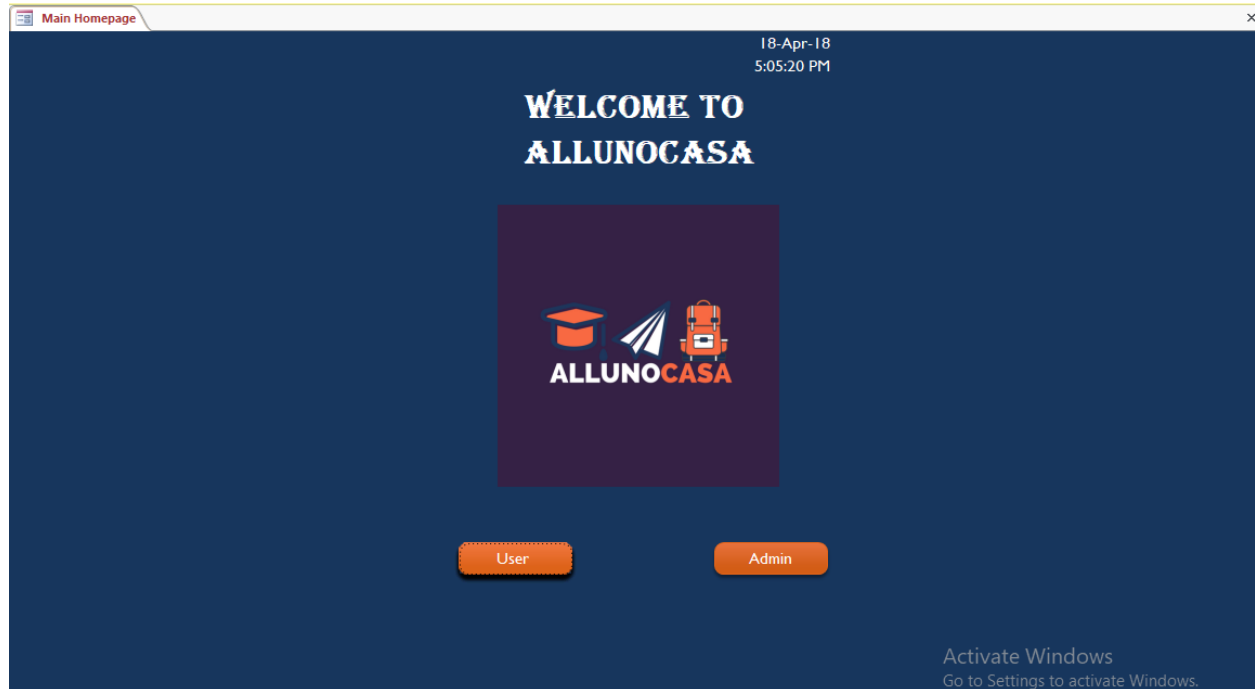
```
AmenityID VARCHAR(10) NOT NULL PRIMARY KEY,  
AmenityName VARCHAR(70) ,  
Description VARCHAR(100) )
```

CREATE TABLE HaveAmenity (

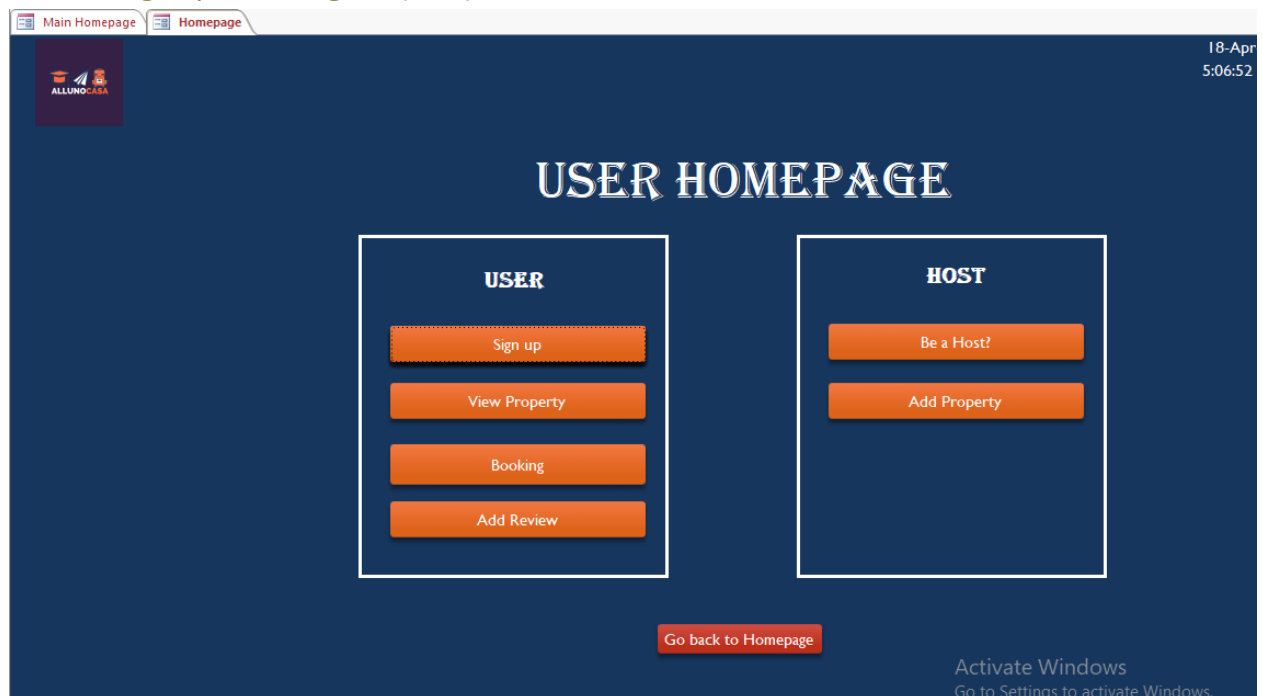
```
FK_AmenityID VARCHAR(10) NOT NULL,  
FK_PropertyID VARCHAR(10) NOT NULL ,  
Availability VARCHAR(45),  
PRIMARY KEY (FK_AmenityID, FK_PropertyID))
```

Main Menu and Data Input Screen Forms:

1. ALLUNOCASA DBMS has a **Main Home Page** from which the user and admin can navigate to other objects like **Forms** and **Reports**.



2. If a user accesses the DBMS he will be directed to the **User Homepage**. This form will allow users to sign up either a guest(user) or a host.



3. After clicking on **sign up** button on user homepage, the system will then require the user to fill in address details in the given form.

ADDRESS DETAILS

18-APR-18 5:07:22 PM

AddressID

Line1

Line2

City

State

Zipcode

Country

Activate Windows
Go to Settings to activate Windows.

The following forms would appear if the user chooses to be a guest(user):

4. The user needs to fill his details in the given form and sign up.

USER DETAILS

18-Apr-18 5:11:13 PM

User ID

First Name Last Name

Phone Number DOB

Gender

Email ID Password

Address ID

Activate Windows
Go to Settings to activate Windows.

5. Once the user signs up the data is saved to database and a pop up appears then the user navigates automatically to the homepage.

The screenshot shows a web application interface with a dark blue background. At the top, there's a navigation bar with tabs: 'Main Homepage', 'Homepage', 'User Address_Form', and 'User ID'. The 'User ID' tab is active. A 'Microsoft Access' pop-up window is centered, displaying 'User Data Saved' with an 'OK' button. Below the pop-up, the 'User Details' form is visible. It includes fields for 'First Name' (Raj), 'Last Name' (Malhotra), 'Phone Number' ((863) 902-2511), 'DOB' (4/13/1990), 'Gender' (male), 'EmailID' (raj@utdallas.edu), 'Password' (masked with asterisks), and 'Address ID' (A101). There are 'Sign me up' and 'Back' buttons at the bottom of the form. The top right corner shows the date '18-Apr-18' and time '5:11:13 PM'. A 'BACK' button is also present in the top right. At the bottom right, there's a message: 'Activate Windows Go to Settings to activate Windows.'

6. User can then view all properties by clicking the 'View Property' button on homepage.

The screenshot shows the 'View Property' page. At the top, the navigation bar includes 'Main Homepage', 'Homepage', 'User Address_Form', 'User ID', and 'View Property'. The 'View Property' tab is active. A 'Back' button is in the top right. The main content area displays a table of properties. The table has columns: 'ity', 'Area in sqft', 'No. Of Room', 'No. Of Baths', 'Property Type', 'HostID', 'AddressID', and 'Price/Day'. The table contains 7 rows of data. The top right corner shows the date '18-Apr-18' and time '5:19:25 PM'. At the bottom center, there is a 'Back' button. At the bottom right, there's a message: 'Activate Windows Go to Settings to activate Windows.'

| ity | Area in sqft | No. Of Room | No. Of Baths | Property Type | HostID | AddressID | Price/Day |
|------|--------------|-------------|--------------|---------------|--------|-----------|-----------|
| 1500 | 2 | 2 | 2 | Apartment | H101 | A101 | \$50.00 |
| 2000 | 3 | 2 | 2 | Apartment | H102 | A102 | \$60.00 |
| 1000 | 2 | 1 | 1 | Apartment | H103 | A103 | \$40.00 |
| 1500 | 3 | 1 | 1 | Apartment | H104 | A104 | \$35.00 |
| 3500 | 4 | 4 | 4 | Villa | H105 | A105 | \$75.00 |
| 3000 | 5 | 3 | 3 | Apt | H104 | A107 | \$60.00 |
| 2500 | 4 | 3 | 3 | Apartment | H105 | A108 | \$45.00 |

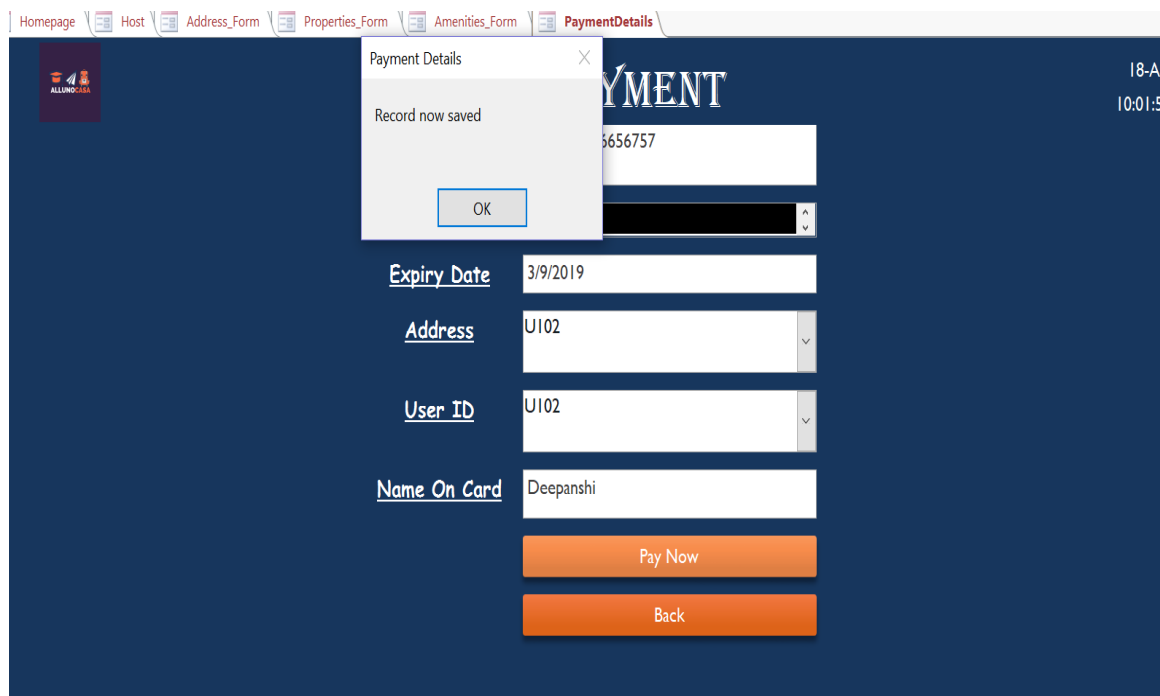
7.To book a particular property the user will navigate to the **booking form** and can fill the booking information.

The screenshot shows the 'BOOKING FORM' interface. At the top, a navigation bar includes links: Main Homepage, Homepage, User Address_Form, User ID, View Property, PaymentDetails, and Booking. The main header area displays the title 'BOOKING FORM' and the date 'Wednesday, April 18, 2018 5:20 PM'. The form fields are arranged as follows: 'BookingID' (text input), 'BookingDate' (calendar icon, showing '18-Apr-18'), 'CheckInDate' (calendar icon), 'CheckoutDate' (calendar icon), 'NoOfGuests' (text input), 'Total Days' (text input), 'Host ID' (dropdown menu), 'Property_Label' (dropdown menu), and 'Guest User ID' (dropdown menu). At the bottom, there are two orange buttons: 'SAVE AND MAKE PAYMENT' and 'BACK'. A Windows watermark is visible in the bottom right corner.

8.After filling in the booking details, if the user clicks save and make payment he will be directed to the **payment form**.

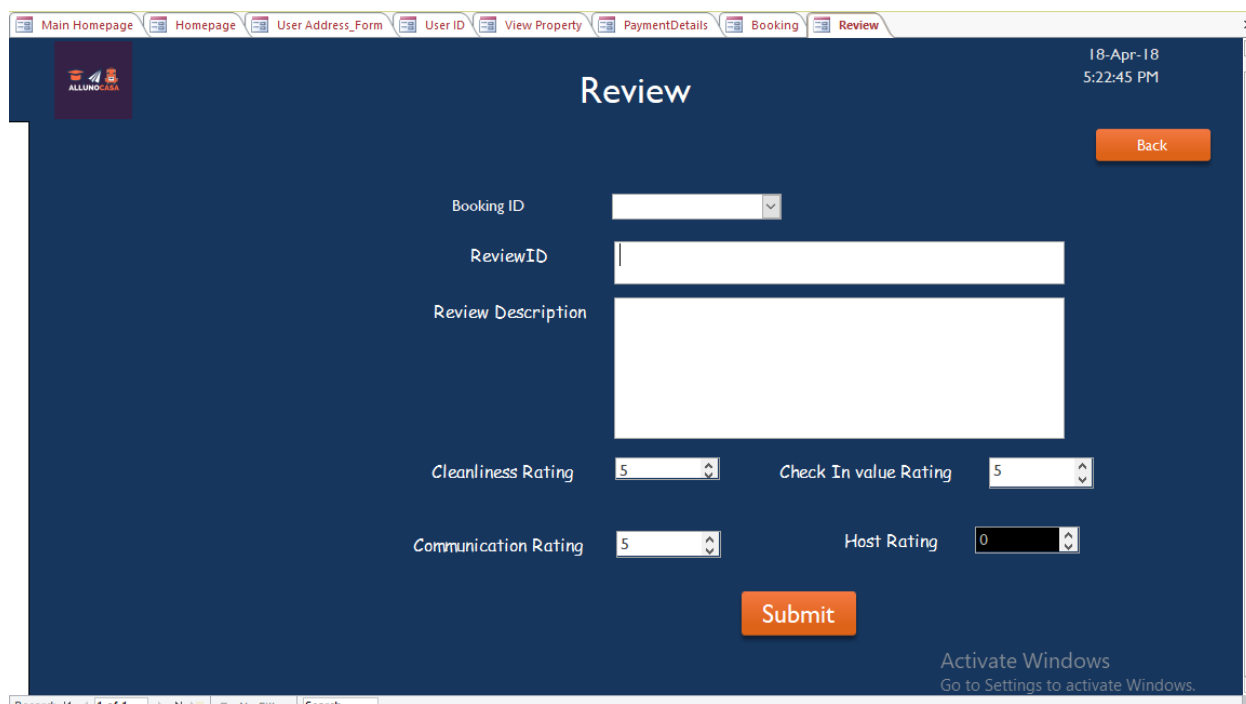
The screenshot shows the 'PAYMENT' interface. The navigation bar is the same as the previous form, but 'PaymentDetails' is now the active page. The main header area displays the title 'PAYMENT' and the date '18-Apr-18 5:20:28 PM'. The form fields are arranged as follows: 'PaymentID' (text input), 'Card No' (text input), 'Card_type' (dropdown menu showing 'Visa'), 'Expiry Date' (text input), 'Address' (dropdown menu), 'User ID' (dropdown menu), and 'Name On Card' (text input). At the bottom, there are two orange buttons: 'Pay Now' and 'Back'. A Windows watermark is visible in the bottom right corner.

9. After filling the payment details and clicking on **pay now**, payment successful message pops up and the user is navigated to homepage.



The screenshot shows a web application interface with a dark blue background. At the top, there is a navigation bar with tabs: Homepage, Host, Address_Form, Properties_Form, Amenities_Form, and PaymentDetails. A modal popup titled "Payment Details" is displayed in the center, showing a message "Record now saved" and an "OK" button. The background form, titled "PAYMENT", contains the following fields: a card number field (partially visible as "656757"), an "Expiry Date" field with the value "3/9/2019", an "Address" dropdown menu with the value "U102", a "User ID" dropdown menu with the value "U102", and a "Name On Card" field with the value "Deepanshi". At the bottom of the form are two orange buttons: "Pay Now" and "Back".

10. From here the User can add review by clicking on **add review** button from the user homepage.



The screenshot shows a web application interface with a dark blue background. At the top, there is a navigation bar with tabs: Main Homepage, Homepage, User Address_Form, User ID, View Property, PaymentDetails, Booking, and Review. The main heading is "Review". In the top right corner, the date and time are displayed: "18-Apr-18 5:22:45 PM". A "Back" button is located in the top right corner. The form contains the following fields: a "Booking ID" dropdown menu, a "ReviewID" text input field, a "Review Description" text area, a "Cleanliness Rating" dropdown menu with the value "5", a "Check In value Rating" dropdown menu with the value "5", a "Communication Rating" dropdown menu with the value "5", and a "Host Rating" dropdown menu with the value "0". A large orange "Submit" button is located at the bottom center. At the bottom right, there is a message: "Activate Windows Go to Settings to activate Windows."

11. Once the booking ID and review details are filled, if user clicks on **submit** then success message pops up and data is saved into review table.

The screenshot displays a web application interface with a dark blue background. At the top, a navigation bar contains links: Main Homepage, Homepage, User Address_Form, User ID, View Property, PaymentDetails, Booking, Review, and Review. The date and time '18-Apr-18 5:24:09 PM' are shown in the top right corner. A Microsoft Access dialog box is open in the center, displaying the message 'Review Submitted Successfully' with an 'OK' button. Below the dialog, the 'ReviewID' field contains 'R114'. The 'Review Description' field contains 'Good experience'. There are four rating fields: 'Cleanliness Rating' (5), 'Check In value Rating' (5), 'Communication Rating' (5), and 'Host Rating' (0). A 'Submit' button is located at the bottom right of the form area. An 'Activate Windows' watermark is visible in the bottom right corner.

12. Clicking on **OK** will navigate the user to homepage.

The following forms would appear if the user chooses to be a host:

1. User can become a host by clicking the button 'Be a Host' on homepage, a new **Host form** would open.

The screenshot shows the 'HOST' form in the Alluno Casa application. The breadcrumb trail at the top includes: Main Homepage, Homepage, User Address_Form, User ID, View Property, PaymentDetails, Booking, Review, Review, and Host. The form has a dark blue background with the 'ALLUNO CASA' logo in the top left. The title 'HOST' is centered at the top, with the date and time '18-Apr-18 5:26:25 PM' in the top right. The form contains three input fields: 'HostID', 'Background Check?' (with a checkbox), and 'UserID'. Below these fields are two orange buttons: 'Add Property Details' and 'Back'. At the bottom right, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.'

2. To list a new property, the host has to click on **add property** button, navigating to **address details form**.

The screenshot shows the 'ADDRESS DETAILS' form in the Alluno Casa application. The breadcrumb trail at the top includes: Main Homepage, Homepage, User Address_Form, User ID, View Property, PaymentDetails, Booking, Review, Review, Host, and Address_Form. The form has a dark blue background with the 'ALLUNO CASA' logo in the top left. The title 'ADDRESS DETAILS' is centered at the top, with the date and time '18-APR-18 5:27:12 PM' in the top right. The form contains seven input fields: 'AddressID', 'Line1', 'Line2', 'City', 'State', 'Zipcode', and 'Country' (which is pre-filled with 'United States'). Below these fields are two orange buttons: 'Next' and 'BACK'. At the bottom right, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.'

3. Clicking **next** would enable the host to fill in new property details in the **property details form**.

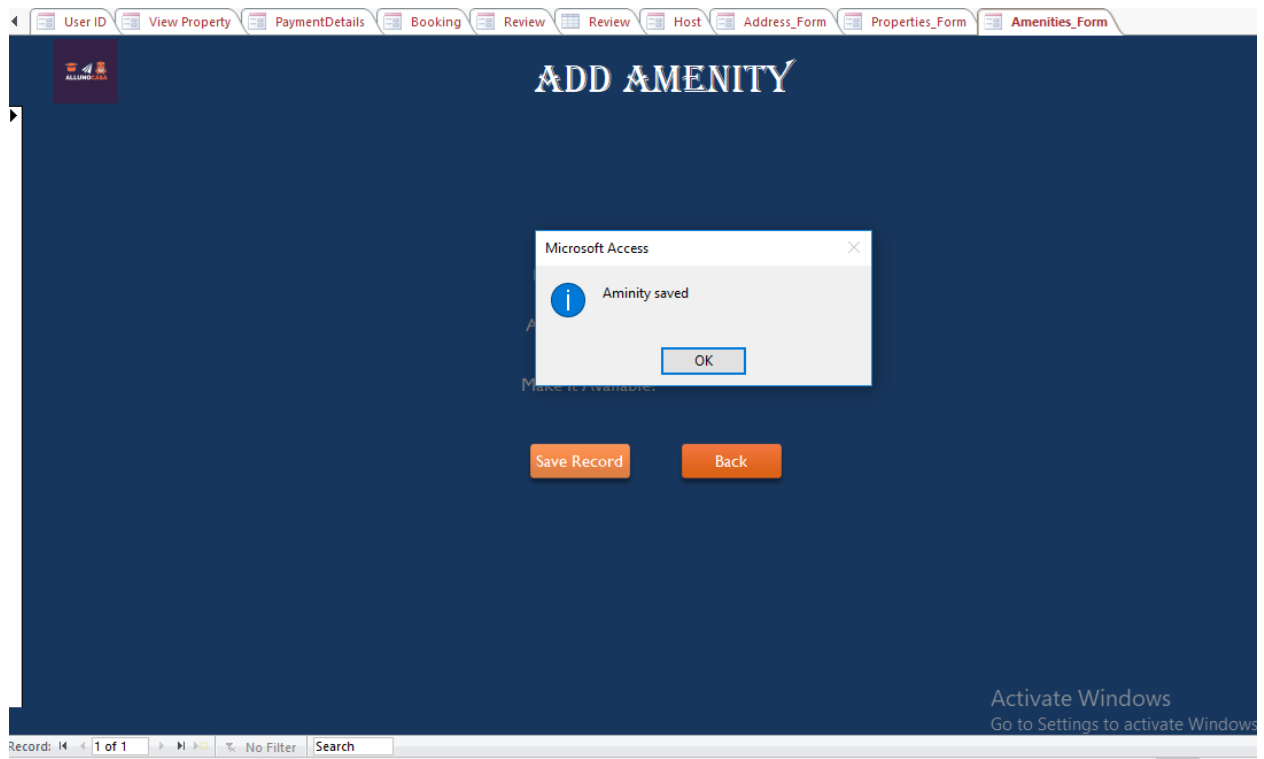
Also, fill in the Host ID from the **dropdown** corresponding to the new Property ID and the Address ID (provided in the previous form) from the **dropdown**.

The screenshot shows a web application interface with a dark blue background. At the top, there is a navigation bar with tabs: Homepage, User Address_Form, User ID, View Property, PaymentDetails, Booking, Review, Review, Host, Address_Form, and Properties_Form. The main heading is 'PROPERTY DETAILS' in white, serif font. To the right of the heading, the date and time '18-Apr-18 5:27:51 PM' are displayed. Below the heading, there are several input fields with labels: 'PropertyID_Property', 'Area in sqft', 'NoOfRooms', 'NoOfBaths', 'PropertyType', 'Host ID', 'Address ID', and 'PricePerDay'. The 'Host ID' and 'Address ID' fields are dropdown menus. At the bottom left, there is a button labeled 'Availability' with a small square icon next to it. At the bottom right, there is a message: 'Activate Windows Go to Settings to activate Windows.'

4. After filling the property details and clicking on **next** the Host can add amenities available in his properties. These details will save to the amenity table.

The screenshot shows a web application interface with a dark blue background. At the top, there is a navigation bar with tabs: User ID, View Property, PaymentDetails, Booking, Review, Review, Host, Address_Form, Properties_Form, and Amenities_Form. The main heading is 'ADD AMENITY' in white, serif font. To the right of the heading, the date and time '18-Apr-18 5:29:08 PM' are displayed. Below the heading, there are three input fields with labels: 'Property ID' (a dropdown menu), 'AmenityID' (a text input field), and 'Make It Available?' (a checkbox). At the bottom, there are two buttons: 'Save Record' and 'Back'. At the bottom right, there is a message: 'Activate Windows Go to Settings to activate Windows.'

5. By clicking **'save record'** the amenity is saved for a particular property. Success message would appear and data is saved to HaveAmenity table.

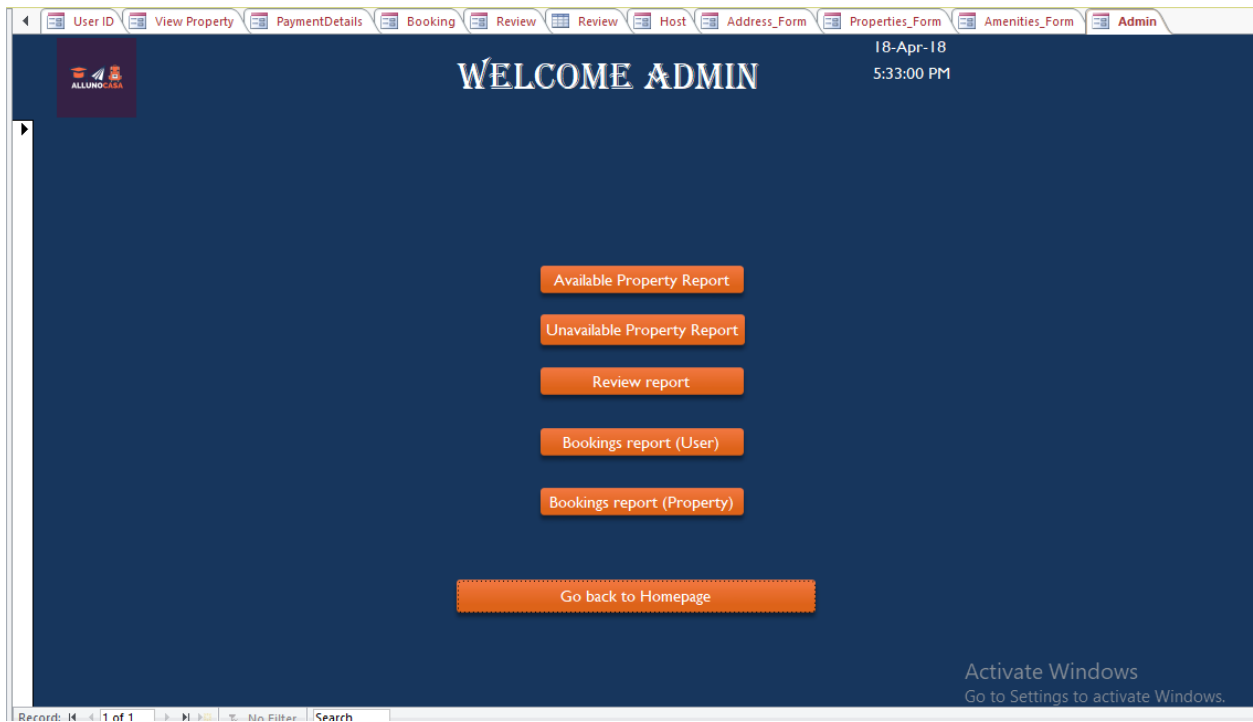


- Host can also directly add property details by clicking add property button on homepage.

The following reports would appear if the admin button is clicked:

Sample Reports:

1. If the admin opens up the ALLUNOCASA DBMS and clicks on '**admin**' button on main homepage- then the following window will open up and then can **view reports**.



2. The following report shows the available properties by clicking on **available properties** button.

Available Properties

Wednesday, April 18, 2018
5:33:29 PM

| PropertyID | Area in sqf | NoOfRooms | NoOfBaths | PropertyType | HostID | AddressID | TotalPrice |
|------------|-------------|-----------|-----------|--------------|--------|-----------|------------|
| P101 | 1500 | 2 | 2 | Apartment | H101 | A101 | 50 |
| P103 | 1000 | 2 | 1 | Apartment | H103 | A103 | 40 |
| P104 | 1500 | 3 | 1 | Apartment | H104 | A104 | 35 |
| P105 | 3500 | 4 | 4 | Villa | H105 | A105 | 75 |
| P109 | 3000 | 4 | 4 | Apartment | H106 | A111 | 60 |

Back

Activate Windows
Go to Settings to activate Windows

3. The following report shows the unavailable properties by clicking on **unavailable properties** button:

Unavailable Properties

| PropertyID | Area in sqft | NoOfRooms | NoOfBaths | PropertyType | HostID | AddressID | TotalPrice |
|------------|--------------|-----------|-----------|--------------|--------|-----------|------------|
| P102 | 2000 | 3 | 2 | Apartment | H102 | A102 | 60 |
| P106 | 3000 | 5 | 3 | Apt | H104 | A107 | 60 |
| P107 | 2500 | 4 | 3 | Apartment | H105 | A108 | 45 |
| P108 | 1500 | 3 | 3 | Villa | H101 | A109 | 80 |

Back

4. The following is a report for reviews for a particular booking which can be viewed by clicking on **Review report** button.

ReviewID: R102

ReviewDescription: Not a good experience

CleanlinessRating: 3

CheckInValue: 3

CommunicationRating: 2

HostRating: 3

BookingID: B102

Back

ReviewID: R103

Activate Windows
Go to Settings to activate Windows.

5. On clicking the **booking report (user)** button the User booking report will open up, which shows number of booking per user.

USER BOOKING COUNT

Wednesday, April 18, 2018
5:35:53 PM


| userID | lumberOfBooking |
|--------|-----------------|
| U101 | 3 |
| U102 | 2 |
| U103 | 2 |
| U104 | 1 |
| U105 | 3 |
| U106 | 2 |

Back

Page 1 of 1

6. On clicking the **booking report (property)** button the Property booking report will open up, which shows number of booking per property.

Navigation: Properties_Form | Amenities_Form | Admin | Available Properties | Unavailable Properties | Review report | User Booking Count | **Property Count Form**

 **PROPERTY BOOKING COUNT**

Wednesday, April 18, 2018
5:36:19 PM

| PropertyID | NumberOfBooki |
|------------|---------------|
| P101 | 3 |
| P102 | 3 |
| P103 | 2 |
| P104 | 2 |
| P105 | 3 |

Total Properties

5

Back

Page 1 of 1

Link to Demo Video:

https://www.youtube.com/watch?v=JcRNE_whguY&feature=youtu.be

Contributions:

Aishwarya Nandapurkar:

- Proposed the idea of the project
- Wrote first draft of Project Report
- Worked on Data Entry
- Designed the initial draft of ERD (on paper) and finalized the same with my project mates
- Designed and developed the database architecture
- Worked on validating the fields and improvised the attribute/column sizes

Deepanshi Puri:

- Wrote first draft of Project Report
- Designed and developed the database architecture
- Finalized ERD for the project on My SQL Workbench
- Formulated Create Table statements
- Formatted Final Word Report
- Spoke the introduction part of the video

Dhairya Pawar:

- Wrote first draft of Project Report.
- Created initial ERD in Lucidchart.
- Designed and developed the database architecture.
- Created and edited the UI for the access application.
- Created, designed and formatted the forms and reports.
- Created queries and developed reports based on these queries.
- Explained the UI flow (Application flow) in the video.
- Created and edited the entire video.

Pawan Dandekar:

- Wrote first draft of Project Report
- Screenshots of forms and reports
- Created various reports such as Property available, unavailable property, review report, booking report (User), Booking report(property) etc.
- Write SQL code to generate above reports
- Added macros to save and go back to home page
- Spoke the report part of the project video
- Generated the popup message for save data in the database table