method Name: Booking()	class name:- Booking

Clients: users

Associated use case: user, customer history, court

Description: The Booking ID will be created when the user login to the application and Book the court.

Arguments received: Whenever the user has book the book and it will fetch the Booking table

Types of value returned: It will return either "Incorrect username (or) password" i.e. Not found (or) ok which is "login and succesfully; and it return 500 (or) 400 Status accordingly

Pre-condition:- For this case, there needs to be valid user whenever he tries to book a court it will create a valid booking ID under his name

Post-condition:- Booking ID will create based on the Booking user

method name: order history()	classname: history

client: user

Associated classes: user, booking

Description: By using user id the booking history will be fetched, the method will go to the database by booking iD and fetch the details based on the user ID.

Arguments received - Based on the customer booking history ID it will fetch the booking data with the date and booking attributes

Types of Values returned: It will return either "Incorrect username" (or) "password" i.e, not found (or) which is "log in and successfully and it return 500 (or) 400 status accordingly

pre condition: The user needs to book the court to fetch the order history

Post-condition: It will reflect all the booking history of the user

method name: court fee()	classname: fee

client: user

Associated classes: court

Description: Each court has a seperate fee it is named as fee id and it is fetch the data.

Arguments received: Their fee amount fee will be added to the booking table and it will store in the data base.

Types of values returned: It will return either "Incorrect username" (or) "password" i.e, not found (or) ok which is "log in and successfully and it return 500 (or) 400 status accordingly

Pre-Condition: There fee can be assigned only if there is court.

Post-condition: The fee will be adjusted in the booking table based on his court booking method name:- Add to cart()   class name:- cart()

client:- user

Associated classes:- user, booking.

Description: After sellecting the court the user needs to add the items in the court. The cart should not be null in order to book the court

Arguments received: After adding items to the court if the costomer wants his/her details he can fetch all the items in the caet.

Types of values returned:- It will return either "Incorrect username" (or)" password" i.e., not found (or) which is log inand successfully and it rettrun 200 (or) 400 status accordingly.

Pre-condition:- The court should not be null.
Caet must have items in the court

Post:- condition:- whenever the user addea the court into his caet the caet need to create under his userID

Methodname: court()	Classname:- court

Client : User

Associated classes :- User, Booking, fee

Description: when the user logged into his account and complete the process the court will be booked...

Arguments received:- The court will be assigned to the booking table under the username

Types of values returned: It will return either "Incorrect username" (or) "password" i.e., not found (or) OK which is "log In and successfully ; and it return 500 (or) 400 status accordingly.

Pre-condition: The court needs to be available for the user to book

post-condition: court will be assigned to the user that will be reflect booking table..

Method Name: Login button	Class Name: Login User	ID: 01

Clients: Users

Associated Use case: Login booking

Description of Responsibilities: Username and password information should retrive dat user from data base and they should not be null.

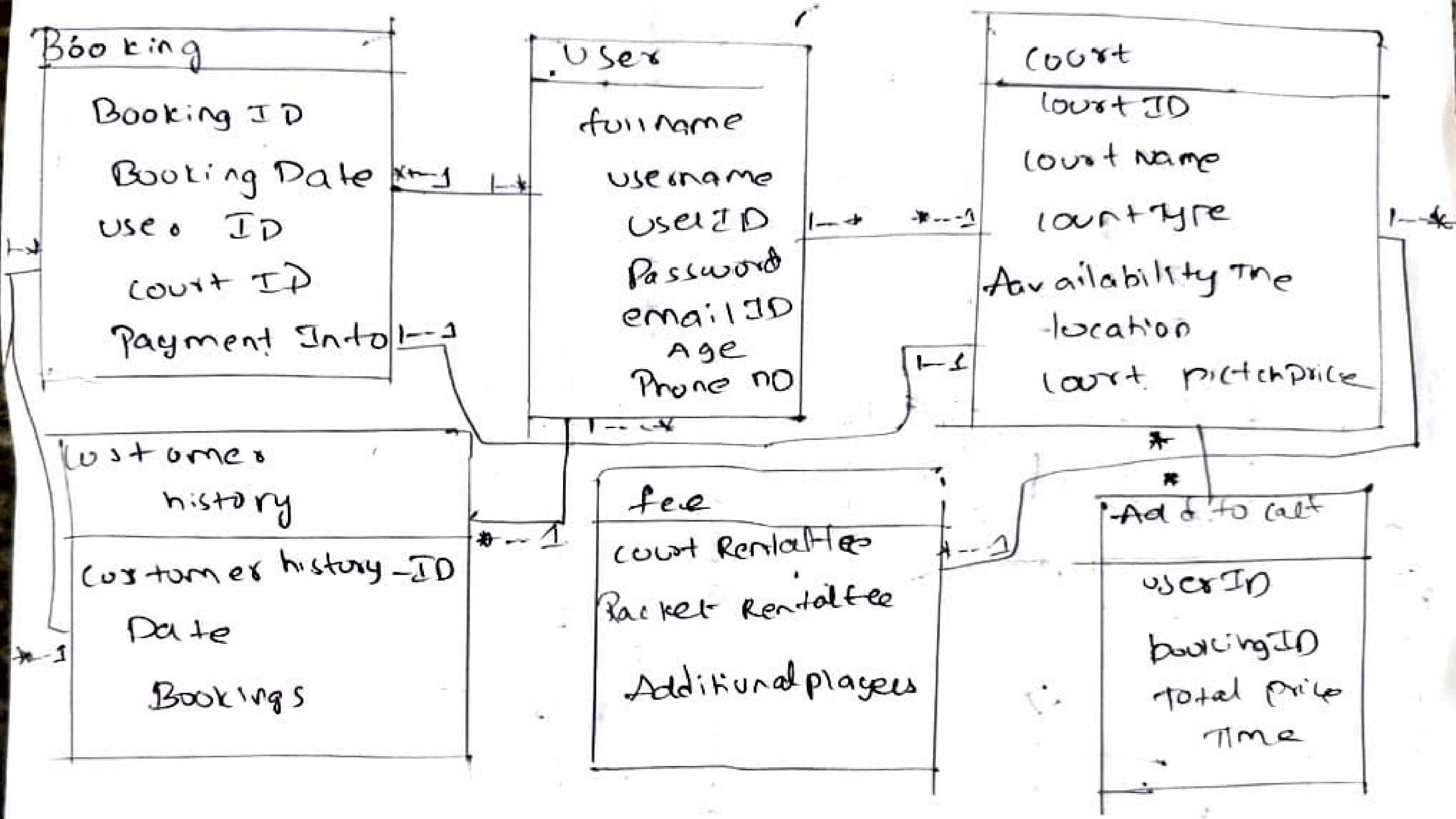
Argument Received: As a response, it receives an object and this object contains the. user information like the user name and encrypted password.

Type of value Returned: It will return either "Incorrect Username or password" i.e 'Notfound or OK which is "Logined successfully; and it returns 500 to 400 status accordingly.

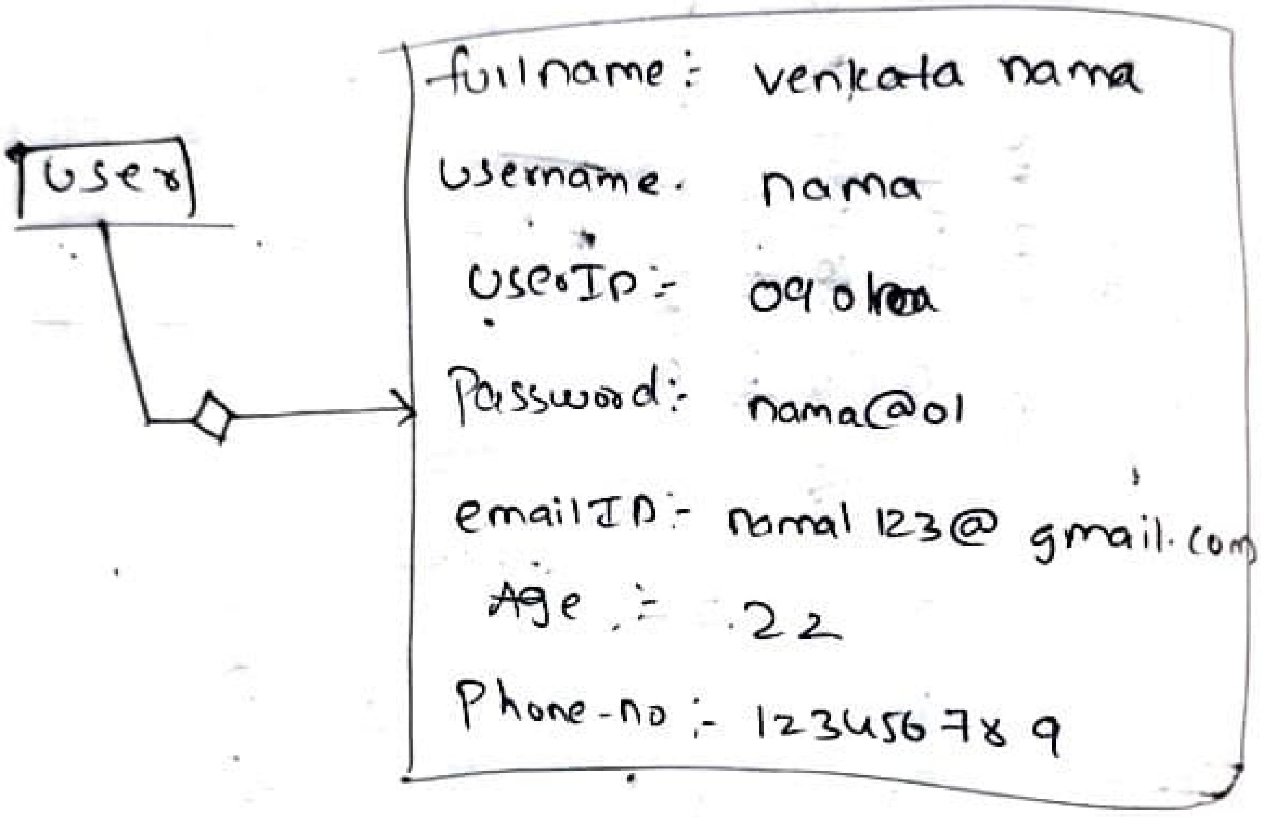
Pre-Condition: User name and the password must not be null as per the pre-condition.

Post-Condition: To verify if the respective user is existing in the database, we must check if i=0;
In the above condition, i refers to the number of rows returned from the database for the given Username and User password.

class Diagram:



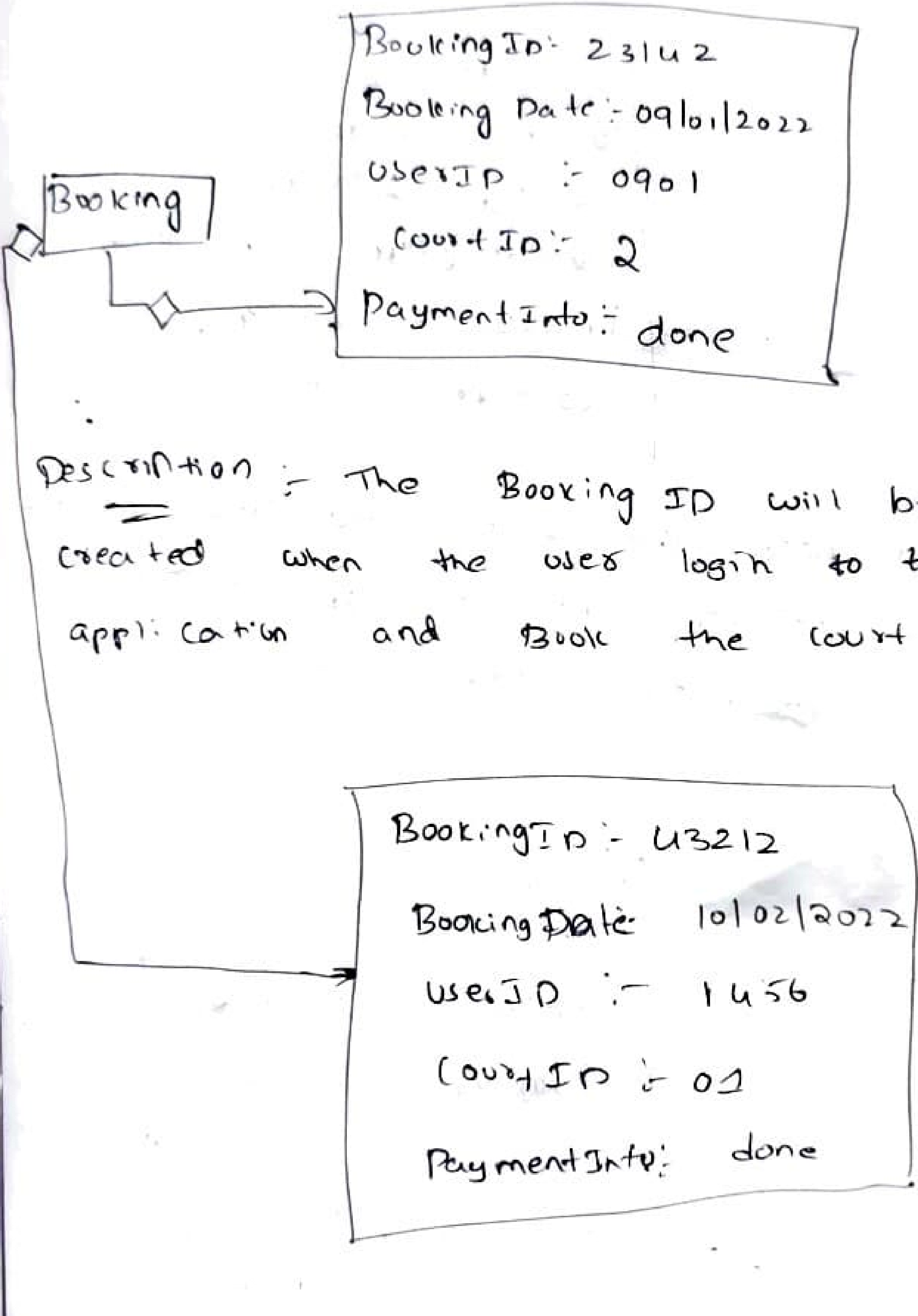
objcct Diagrams:- [User->class]



Description: username and Password inform

should retrieve user fom data base

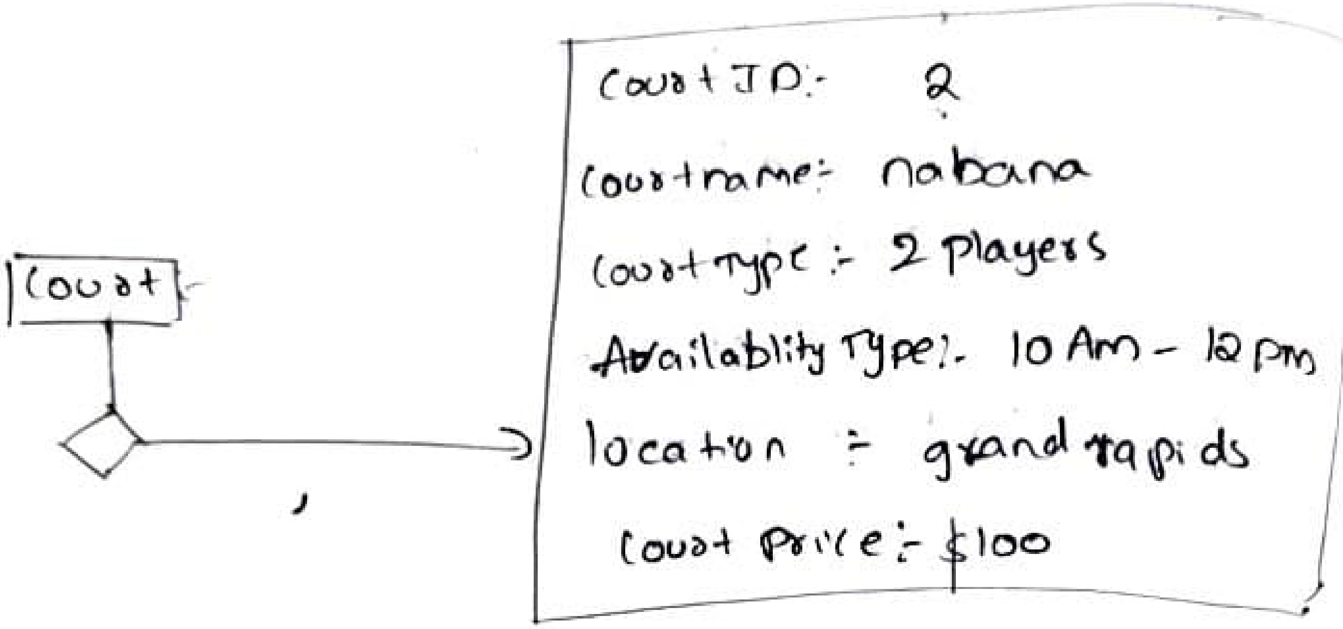
and they should be null



BookingID:- 23142
Booking Date:- 09/01/2022
UserID :- 0901
Court ID:- 2
Payment Info:- done

Description :- The Booking ID will be created when the user login to the application and Book the court

BookingID :- 43212
Booking Date: 10/02/2022
UserID :- 1456
Court ID :- 01
Payment Info: done



Court ID:- 2

Courtname:- nabana

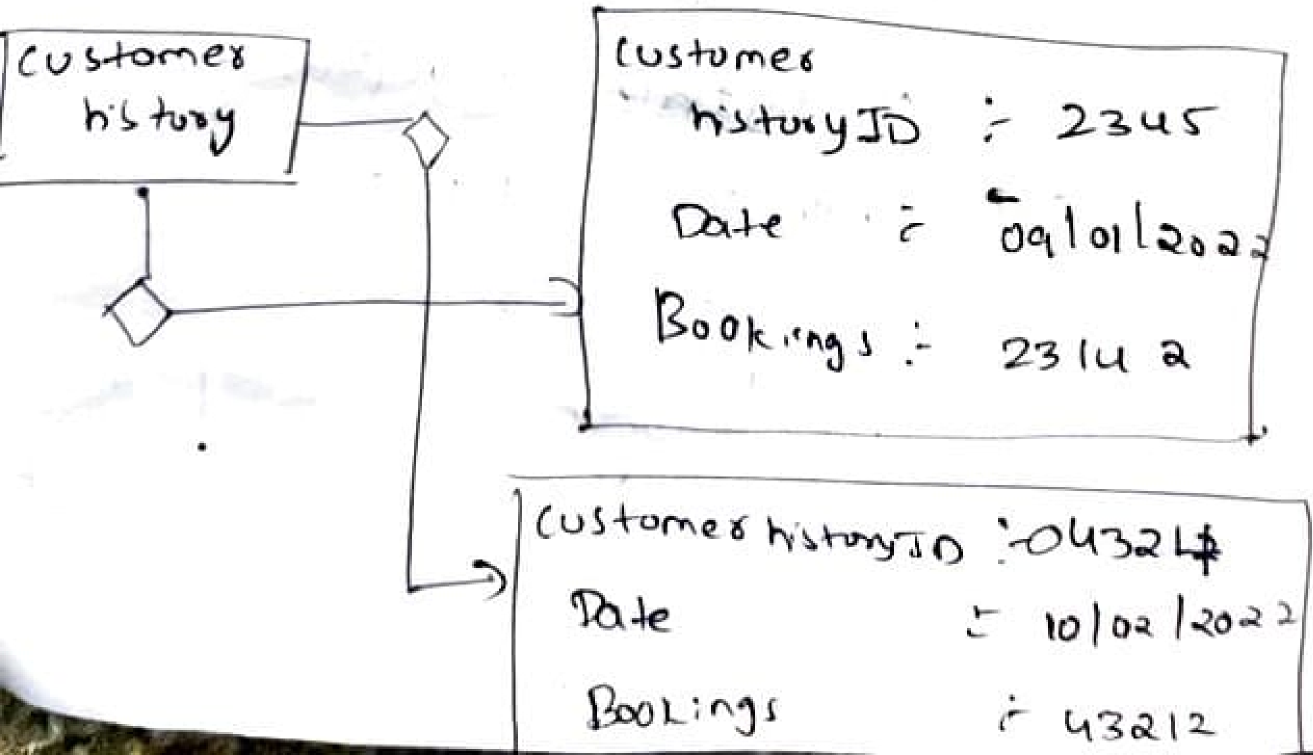
Courttype :- 2 players

Availablity Type:- 10 Am - 12 pm

location :- grand rapids

Court price:- $100

Description:- After the user logged into his/h
account selecting the court type, time,
location. court price the court will be
booked



Customer historyID :- 2345

Date :- 09/01/2022

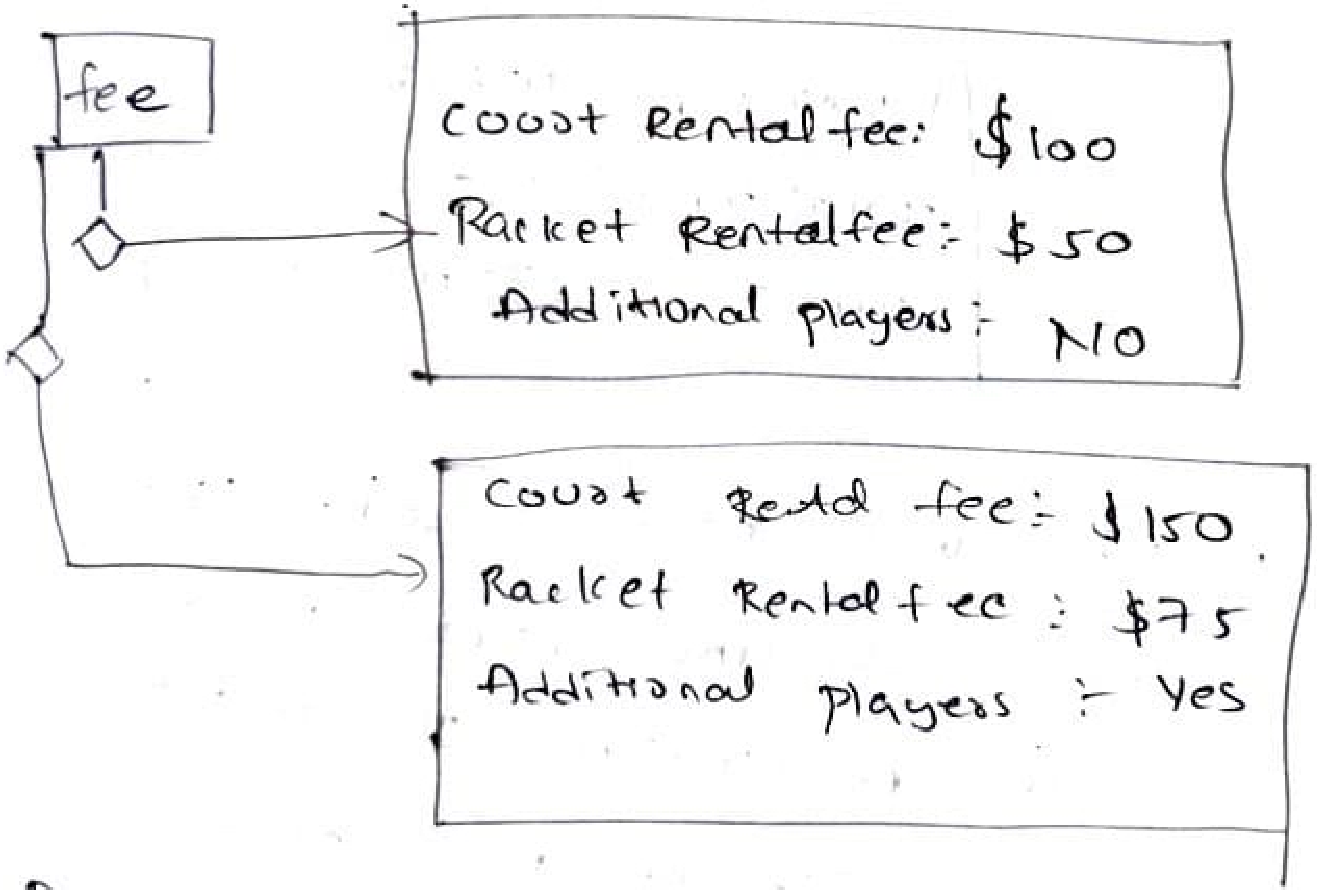
Bookings :- 2314 2

Customer historyID :-04324$

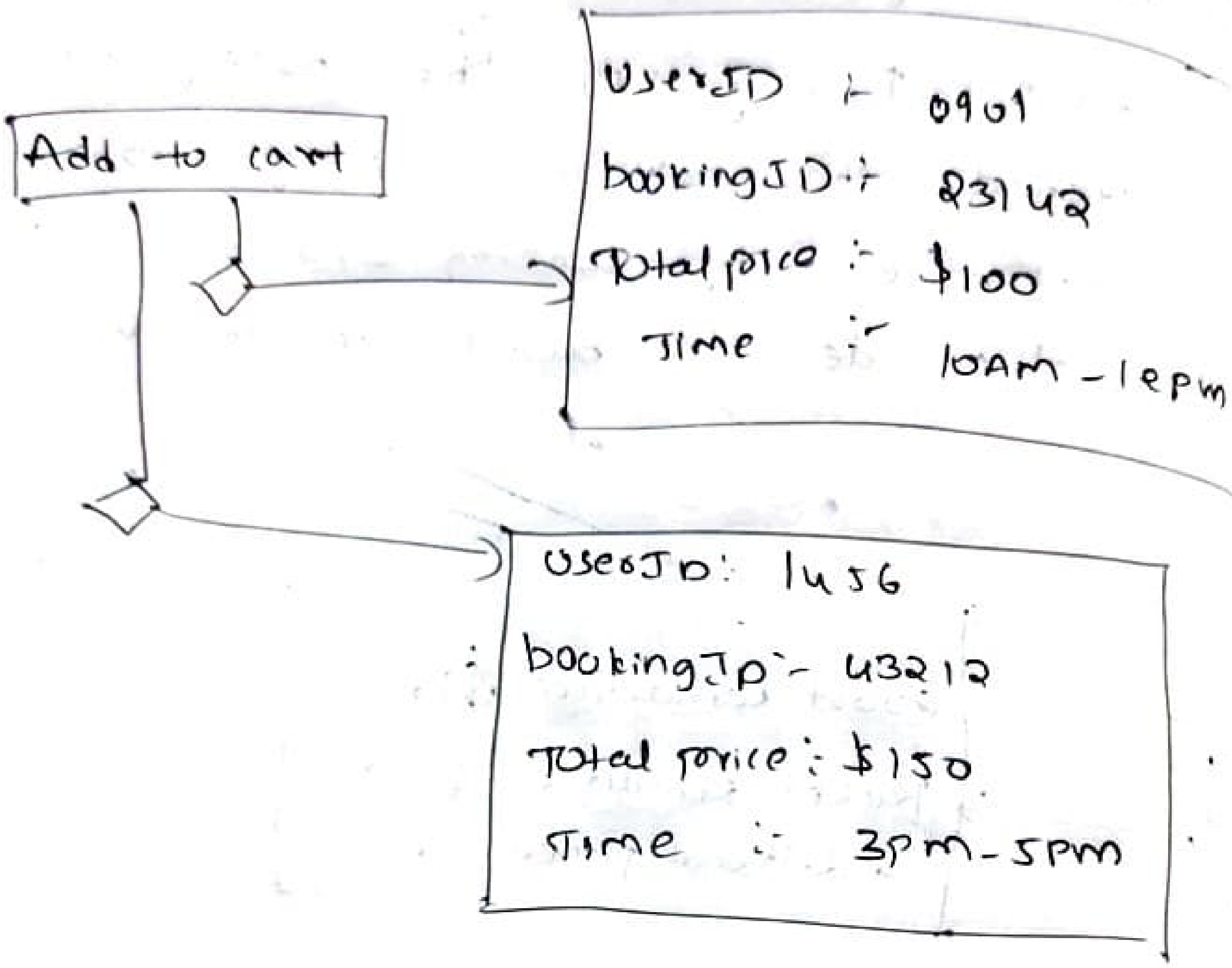
Date :- 10/02/2022

Bookings :- 43212

Description:- By useing customer history ID and Date, Booking ID the history will be fetched, the method will go to the database by booking ID and fetch the details based on the user ID.



Description: each court has a seperate fee and user needs to the select the court desired and it will ask additional Players if the user needed, it will fetch the details.



Description:- The user needs to add the selected items to the cart it will have userID, bookingID, total price, time. The cart should not be null in order to ~~totut~~ book the court