Criterion B: Design overview

Table Structure

Commented [B1]: Database tables shown clearly which depicts the overall structure of the database. ERD shown to explain relationship between tables.

Routes	
route_id	Integer
bus_id	Integer
start_time	Integer
end_time	Integer

Busses	
bus_id	Integer
bus_name	String
reg_no	String
contact_no	Integer
seat_limit	Integer

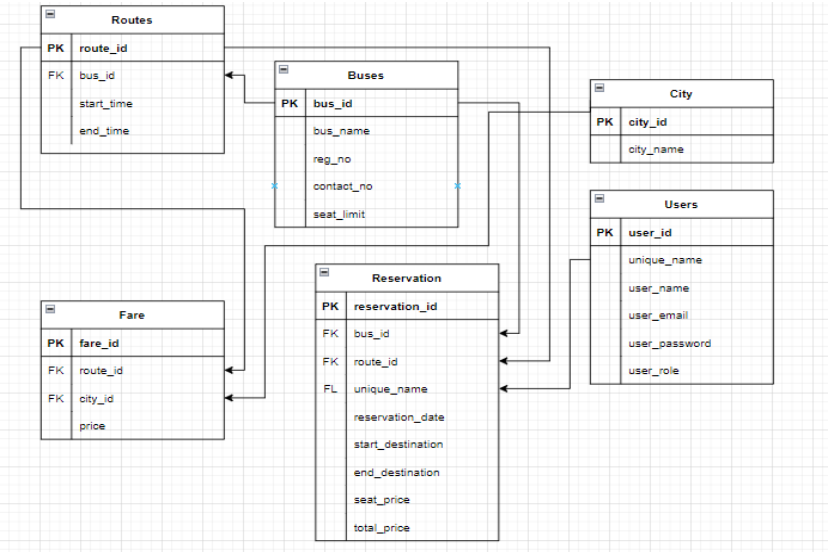
Reservation	
reservation_id	Integer
route_id	Integer
bus_id	Integer
unique_name	String
reservation_date	Integer
start_destination	String
end_destination	String
seat_price	Floating Point
total_price	Floating Point

Fare	
fare_id	Integer
route_id	Integer
city_id	Integer
price	Floating Point

Users	
user_id	Integer
unique_name	String
user_name	String
user_email	String
user_password	String
user_role	Boolean

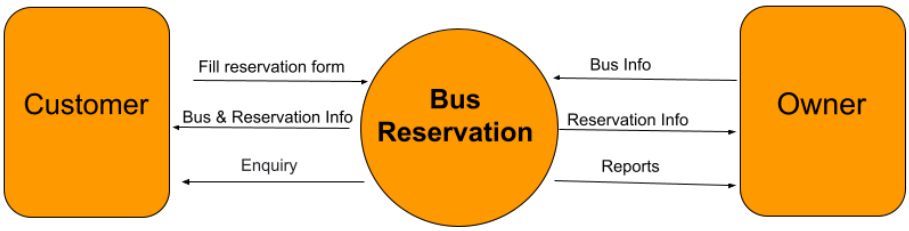
City	
city_id	Integer
city_name	String

Entity relationship diagram (ERD)

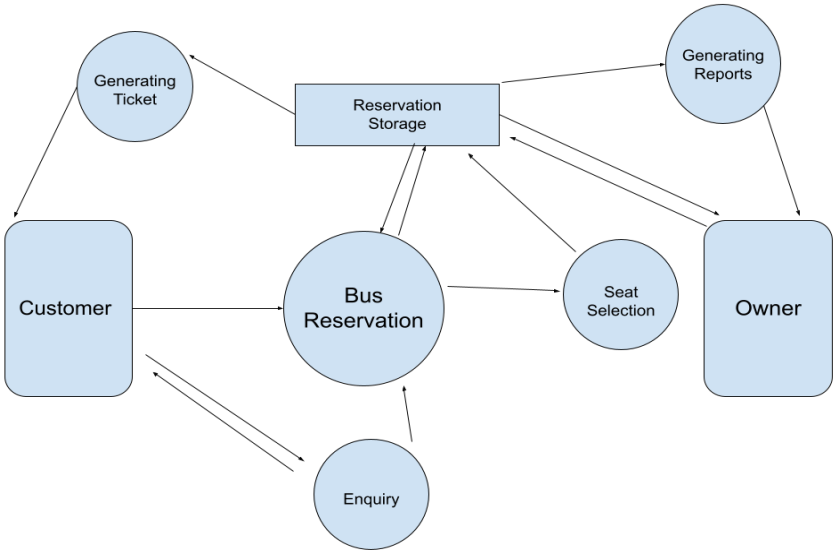


Data flow diagram level 0

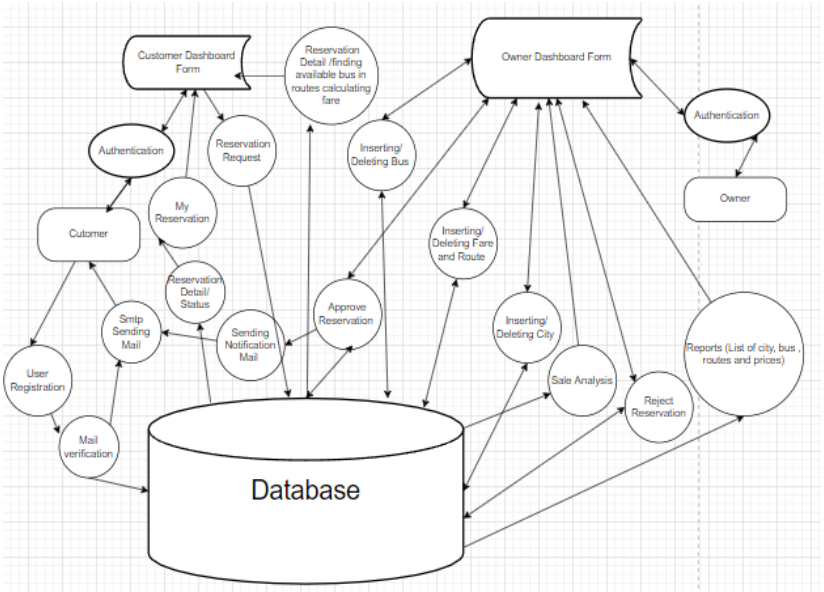
Commented [B2]: Different levels of Data flow diagrams are stated.



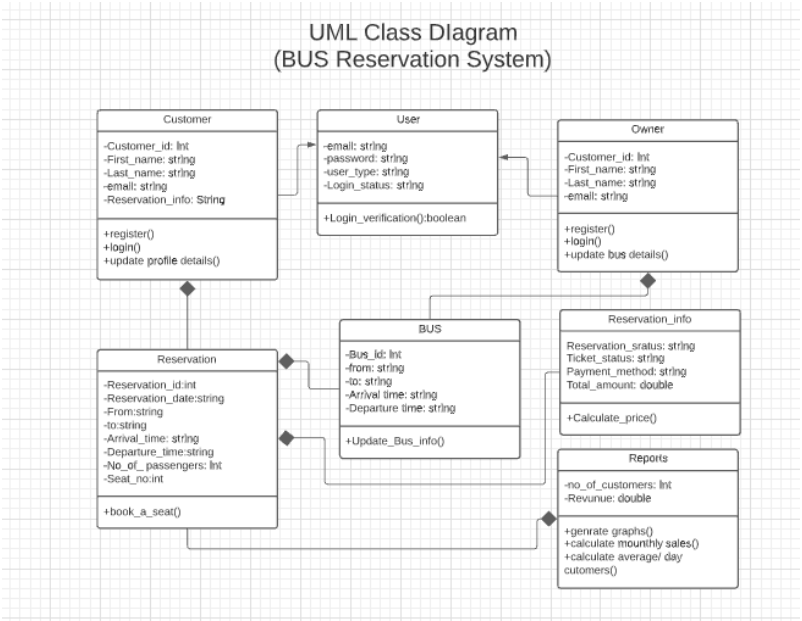
Data flow diagram level 1



Data flow diagram level 2

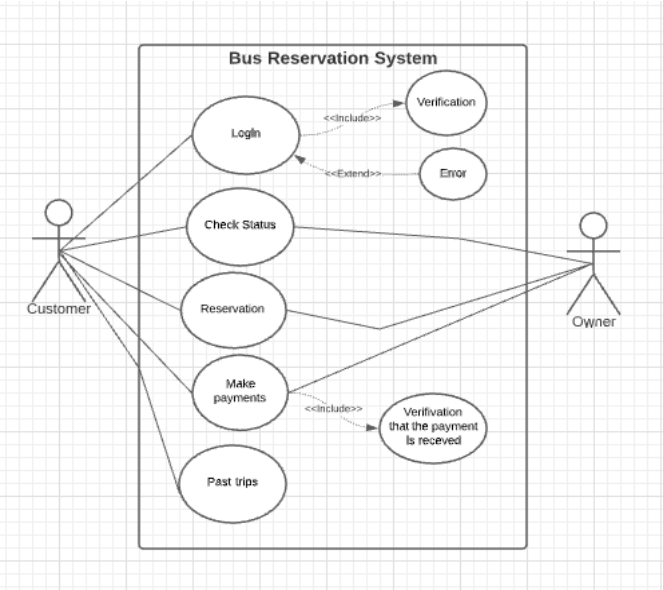


UML Class Diagram



Commented [B3]: UML diagrams which explains about the design of the system.

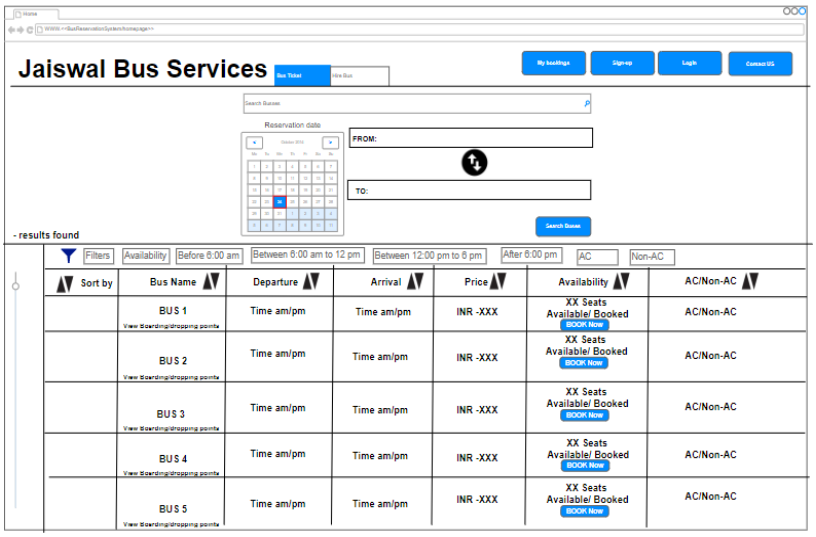
UML Use Case Diagram



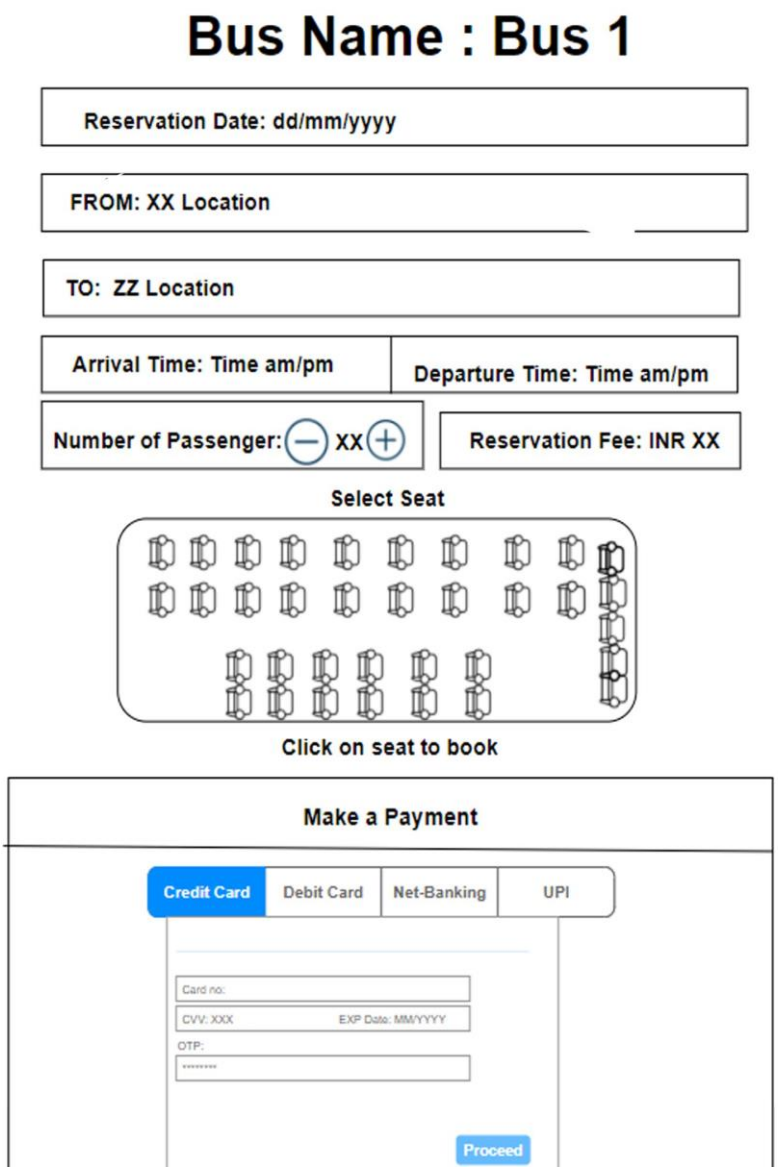
Wireframes

Commented [B4]: Wireframes showing layout and Graphical User Interface elements.

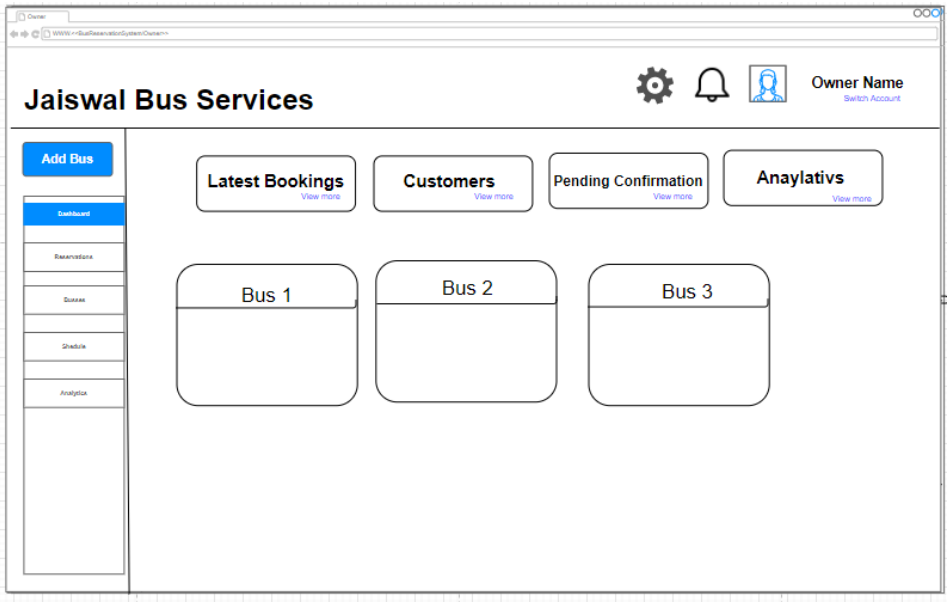
Customer Home page



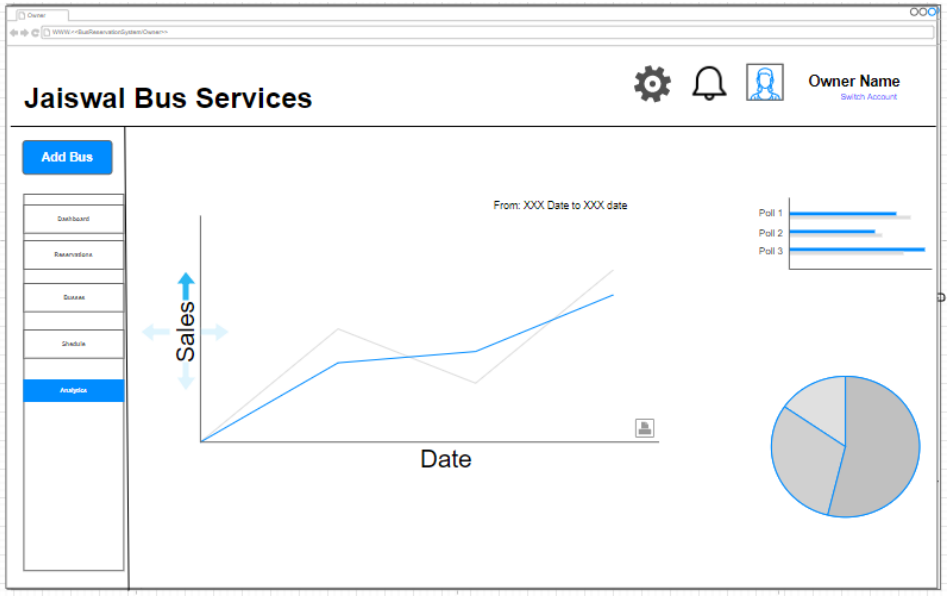
Customer Reservation Form



Owner Admin Page



Owner Analytics



Pseudocode for Fare calculation

```
City [ ]: [ 'Indore', 'Khudel', 'Double Chouki', 'Chapda', 'Badi',
'Bijwar', 'Kanod']
Fare [ ]: [ 0 ,  30  ,  40 , 60 , 100 , 110 , 130]

Start_city =  input('Enter Start Destination')
End_city =  input('Enter End Destination')
Seat_req = input('Enter Seat Required')

If Start_city and End_city and Seat_req == true
i=0
Loop i from 0 to 8

If Start_city = City[i]
City[i] = Start_city
Int_start = i

If Start_city = City[i]
City[i] = Start_city
Int_end = i
endloop

Start_Price = Fare[Int_start]
End_Price = Fare [Int_end]
Final_Price = End_Price - Start_Price
Total_Price = Final_Price * Seat_req
else
return 'Please Give the Input'
```

Pseudocode for sending reservation invoice via mail

```
if reservation created = true    then
reservation_id = reservationid()// this will store the reservationid
of reservation in variable
user_email = useremail()// this will store the current user email in
variable
invoice_data [ ] = [select from table='reservations' where
reservation_id = "reservation_id"]

bus_id =  invoice_data [ 'bus_id' ]
bus [ ] = [select from table='bus' where bus_id = "bus_id"]
bus_name = bus ['bus_name']



mail() to 'user_email'
      output 'You have successfully reserved the bus seat, refer to the
details below'

      Receipt Name: invoice_data [ 'unique_name' ]
      Bus Name: bus_name
      From: invoice_data [ 'start_destination' ]
      To:invoice_data [ 'end_destination' ]
      Total Price: invoice_data [ 'total_price' ]

else output 'error: failed to reserve the seat(s)'
```

Test Plan

Commented [B5]: Test plan outlined which addresses all criteria of success.

Action to be tested	Nature of the test	Expected Result
The system will allow the user to sign up and login into the system.	In this test the application will be tested by giving input to sign up and login form which will test that the user is able to sign up and login into the system.	The user will be able to sign up and login into the system
The system will verify the user via email	In this test when the user signed up then how does it verify the email of the user will be tested.	The system is able to verify the user
The system will allow the user to search busses for chosen date and locations	In this test the input will be given to form of date and locations, and the test will be conducted to check whether the buses are searching on given input.	The busses are searchable on chosen destinations and date
The system will allow the user to reserve the bus by sending a request to the admin.	In the test when the reservation is requested then the action needs to be tested that the system is sending the request of reservation to admin.	The request is sent to the admin when users reserve seats.
The system will notify the user about their reservation via email.	In this test, when the reservation request is approved by the admin then, the system will send notification mail to the user who reserved the seats for the bus. The action of the mail needs to be tested.	The mail is sent about the reservation to the user
The system will allow users to make payments for reservations.	In this test the system will be tested whether the system allows the user to make payments for the reservations on the application.	The user is able to make payments for the reserved seats in the application.
The system will allow the user to view upcoming and previous trips.	In this test, when the user is logged into the system the previous and upcoming trips are viewable that will be checked.	The upcoming and previous trips are viewable to the user.

The web-application will be responsive working on all the resolutions.	In this test the application will be tested in different resolutions where it is tested that the application is displayed correctly.	The application is displayed correctly on different resolutions.
The system will allow admin to add,delete and update the buses, routes, and cites.	In this test the admin account will be logged in and the test will be conducted whertere the admin is able to add , delete and update the buses, routes and cites.	The admin is able to add, delete and update busses, routes and cites
The system will allow admin to create the fare chart.	In this test the admin account will be logged in and it will be checked whether the admin is able to design the fare chart.	The admin is able to design the fare chart for particular route
The system will allow automatic calculation of fare after the route had been created by admin	In this test when the bus is searched, then the pricing will be tested, whether the calculations are made, if made they are accurate.	The automatic calculations are made successfully with accurate results.
The system will allow the user to select the seats for the reservation from the seat selector.	In the test the seat selector is working will be tested	The user is able to select the desired seats.
The system will allow admin to view the reservation requests	In this test when reservation requests are sent to the admin then the admin can view the requests or not will be checked.	The admin is able to view the reservation request sent by the customer
The system will allow the admin and user to view reports with sorting and searching.	In this test the actions will be tested whether the user or admin is able to search and sort the data.	The user and admin is able to search and sort the data
The system will allow admin and user to download and print the reports.	In this test, the reports are downloadable and printable will be checked.	The reports are able to download and print.
The system will allow user to personalize their profile by adding avatars	In this test the customer account will be logged in and it will check whether the customer is able to add avatar	The user is able to add the profile avatar.

	to their profile.	
The system will allow admin to approve or reject reservation request	In this test where the admin can view the request, it will be checked that whether or not the admin is able to approve and reject the requests sent by the customer for reserving seats.	The admin is able to approve or reject the request.
The system will generate graph for sales	In this test it will be tested that the admin is able to view the sales graph on the dashboard and the graph is working as the reservation increases and the data is reflected on the graph	The graph is viewable to the admin with all the appropriate data. As the data increases, it is reflected on the graph.