

## PHASE 2 PROJECT

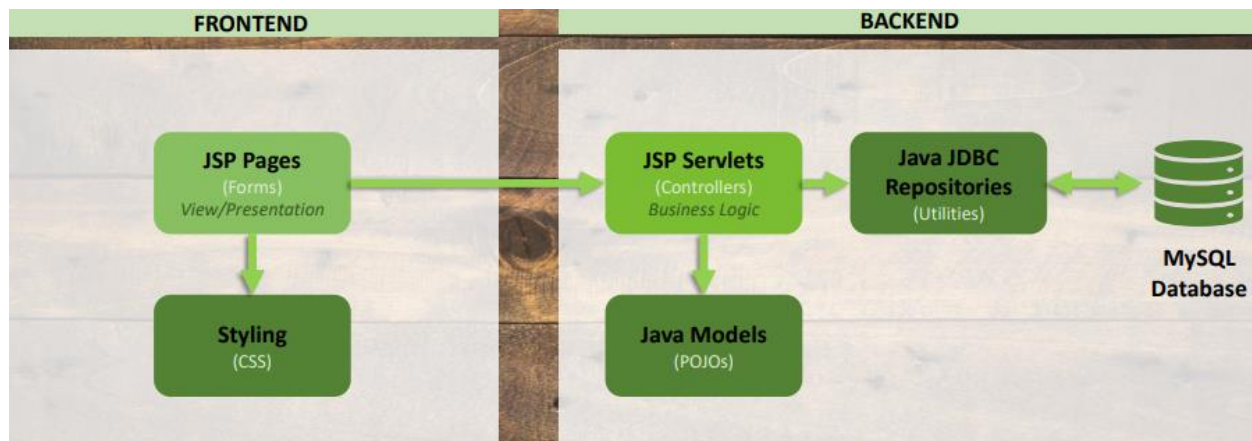
### NIVYA S

FlyAway is an airline ticket-booking portal that lets people book flights on their website. They decided to hire a Full Stack Developer to design and develop the portal. The goal of the company is to deliver a high-end quality product as early as possible. This project focuses on developing a backend project using JDBC, JSP, and Servlets, although a minimal frontend should be implemented. You can create frontend forms on your own or use online resources. It should include a search form to allow entries of travel details, list the available flights with ticket prices, and a confirmation of ticket and the journey details.

Technical Components – Backend & Database Management

JSPs, Servlets, and JDBC to implemented a fully connected application to a MySQL Database .Implement frontend Form Design . Implement Database CRUD features using JDBC . Database design and scripting . Designed a normalized database structure . Created a database initialization script . Populated tables via scrip.

### DESIGN



List Users

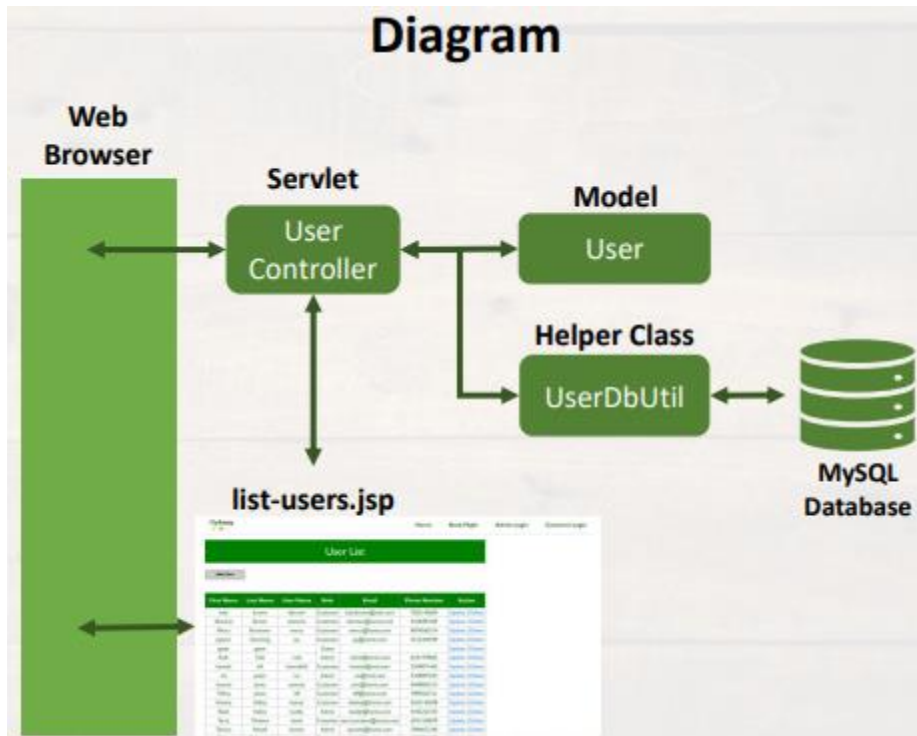
To Do List 1.

Create User.java

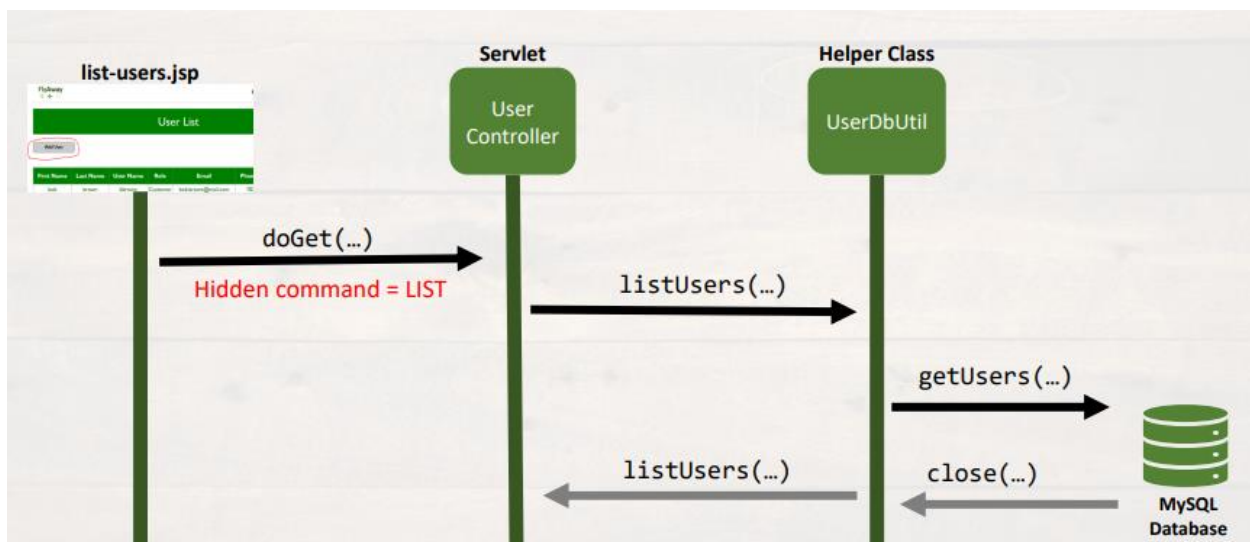
2. Update UserControllerServlet.java to handle the request to add a User (using LIST command)

3. Create UserDbUtil.java to add a JDBC getUsers() method

4. Create JSP page: list-users.jsp



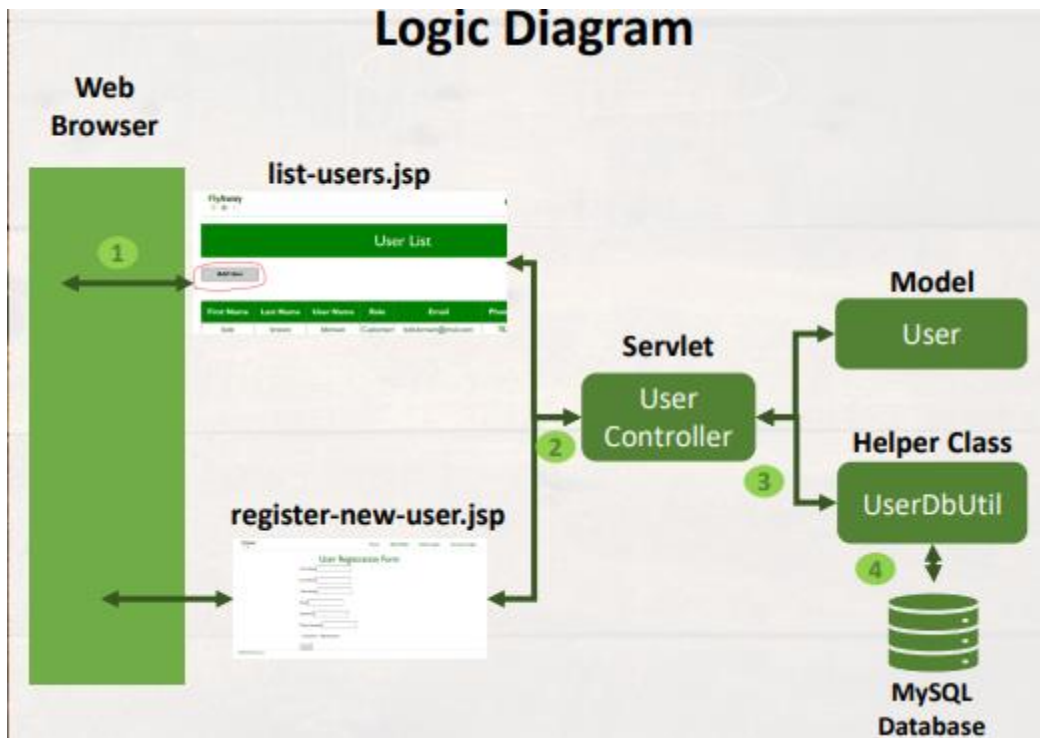
List Users: Sequence Diagram



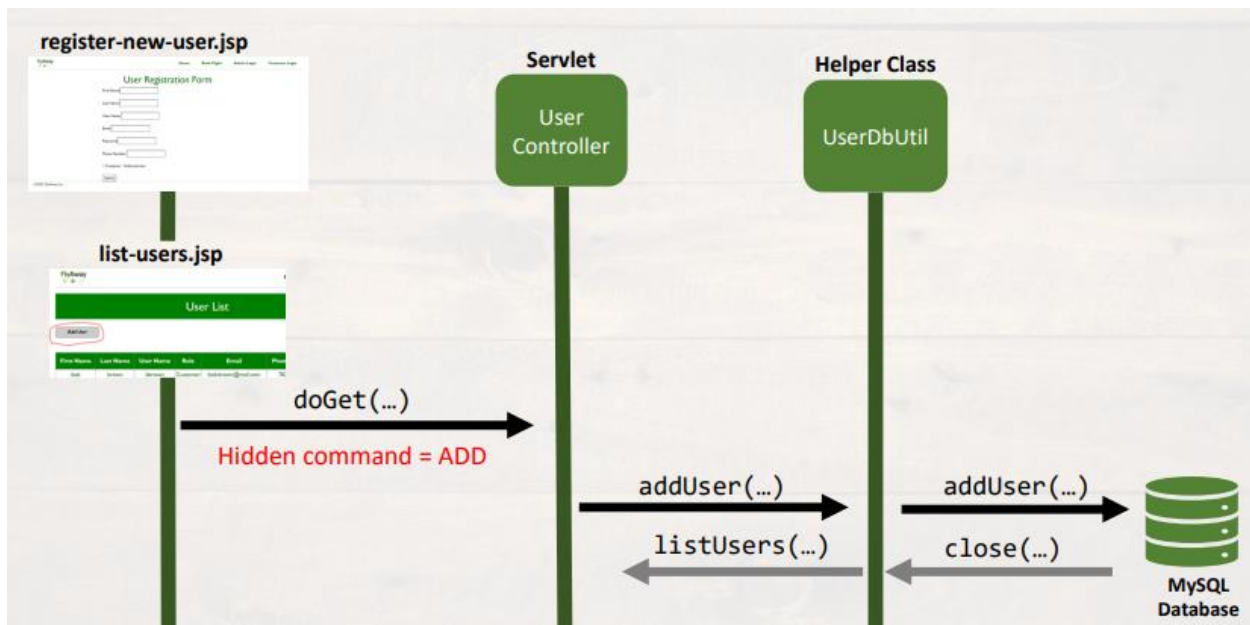
### Add New User

- To Do List 1. Update list-users.jsp to add the "Add User" button
2. Update Landing Page to include a link to the "Add User/Register" form
3. Create Form (JSP page) : registernew-user.jsp
4. Update UserControllerServlet.java to handle the request to add a User (using ADD command)

5. Update UserDbUtil.java to add a JDBC addUser() method



Add New User: Sequence Diagram



Update Existing User

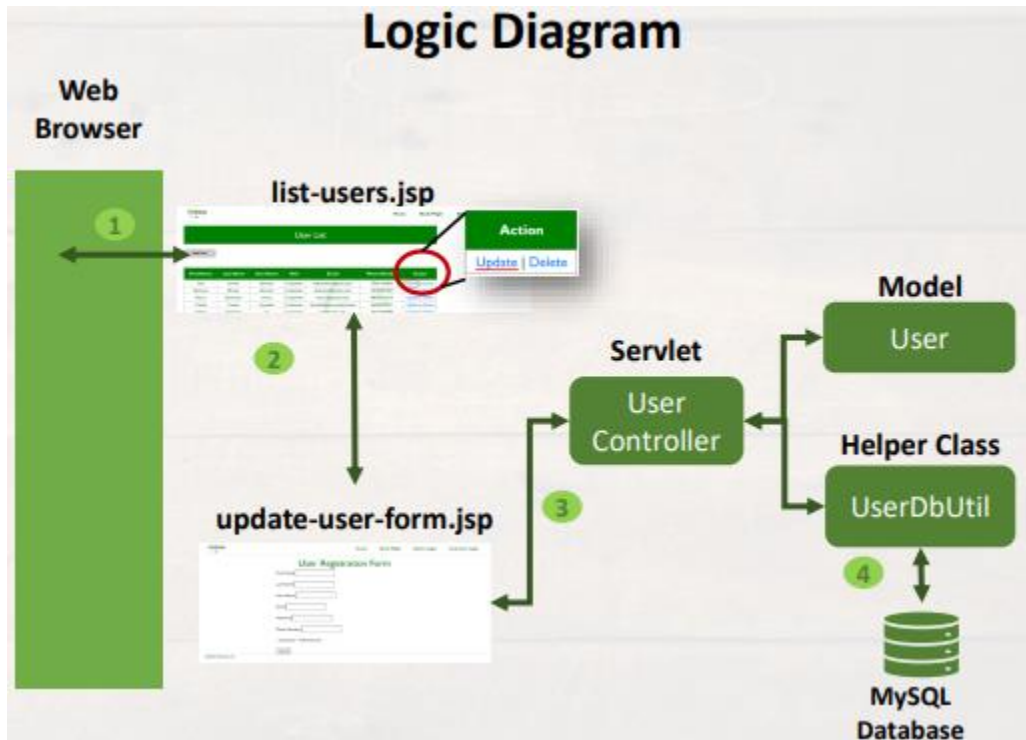
To Do List

1. Update `list-users.jsp` to add "Update" link

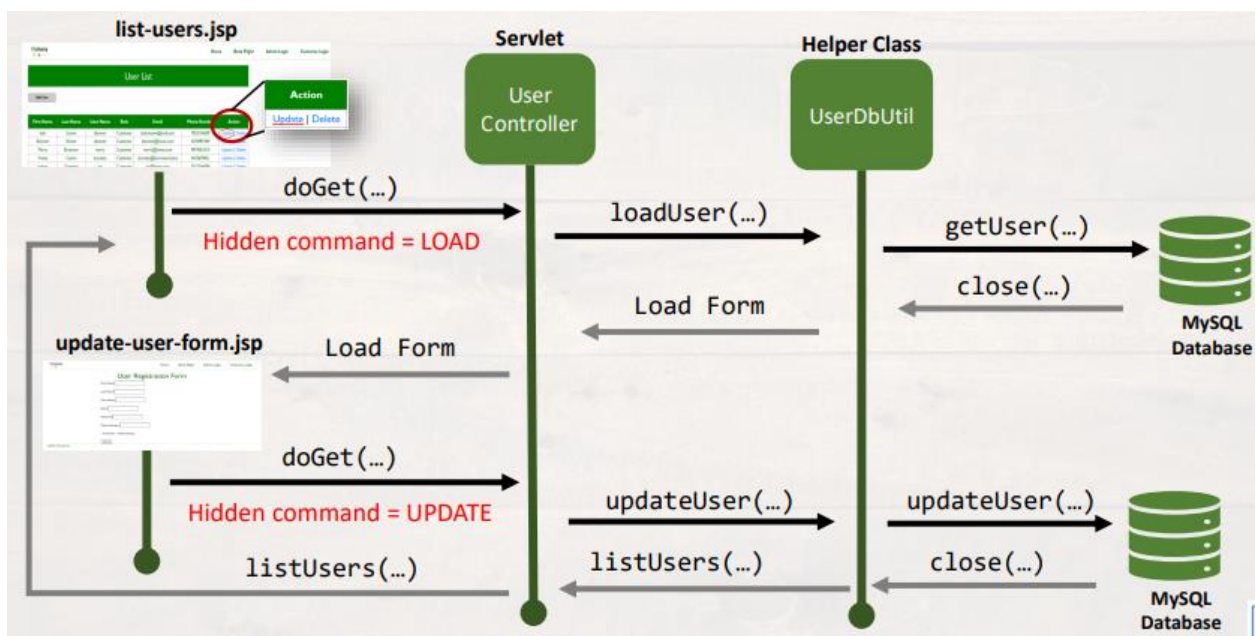
2. Create Form (JSP page) : updateuser-form.jsp

3. Update UserControllerServlet.java to handle the update request • Pre-populate the update-user-form with existing database information (using LOAD command) • Add logic to update the requested User, using the form data (using UPDATE command)

4. Update UserDbUtil.java to add a JDBC method for updating a User



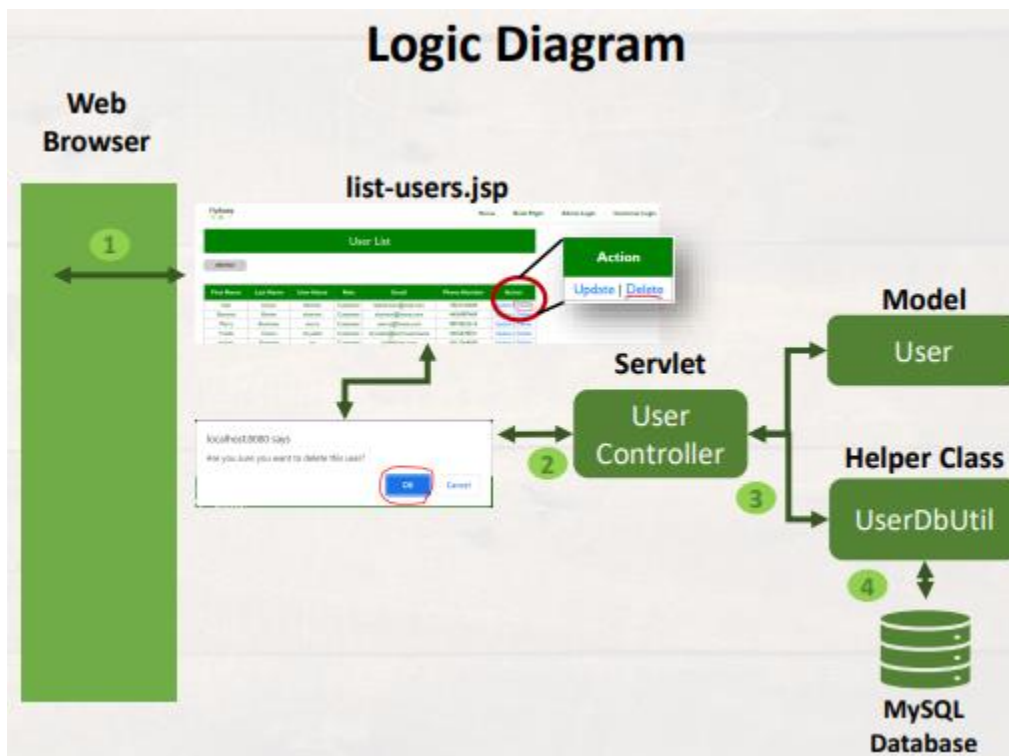
Update Existing User: Sequence Diagram



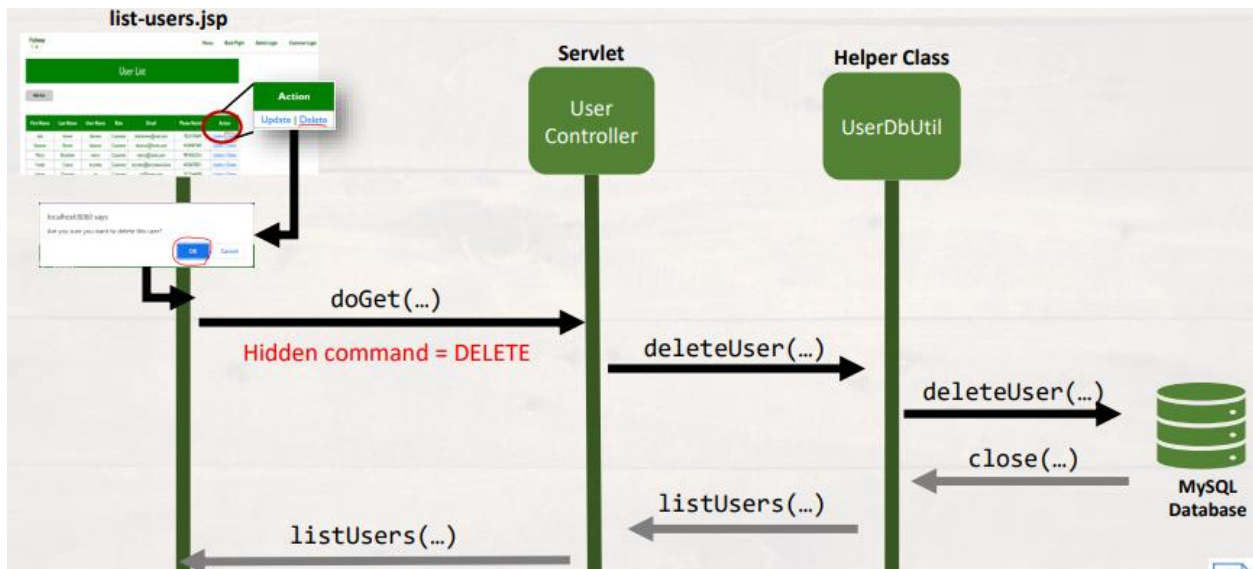
## Delete Existing User

### To Do List

1. Update list-users.jsp to add "Delete" link • Add logic to display a confirmation window
2. Update UserControllerServlet.java to handle the request to delete a User (using DELETE command)
3. Update UserDbUtil.java to add a JDBC deleteUser() method



### Delete Existing User: Sequence Diagram

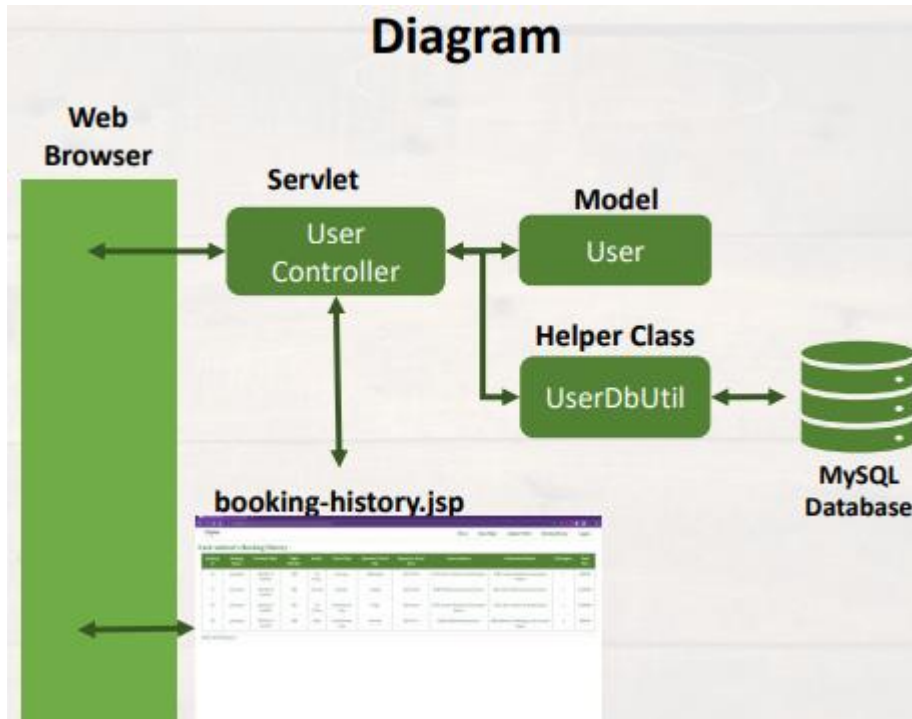




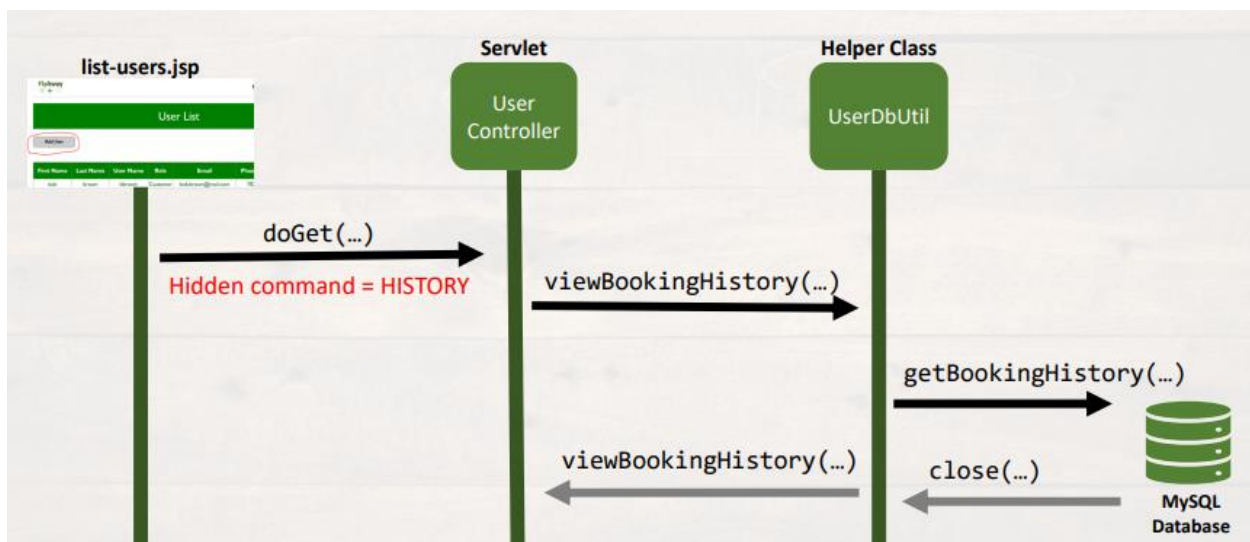
## Booking History (Single Logged-In User)

### To Do List

1. Update UserControllerServlet.java to handle the request to add a User (using HISTORY command)
2. Create a new object to hold the booking history results – BookingHistory.java
3. Create UserDbUtil.java to add a JDBC getBookingHistory() method
4. Create JSP page: bookinghistory.jsp.



### Booking History: Sequence Diagram (Single Logged-In User)



GITHUB LINK