



First Year Case Study

Objectives

- **To introduce the first year case study.**
- **To study the Use Case model.**
- **To study the case study architecture.**
- **To consider the analysis model.**
- **To consider the domain classes.**
- **To view the JavaDB Database tables.**
- **To study the design model.**
- **To execute and use the case study.**

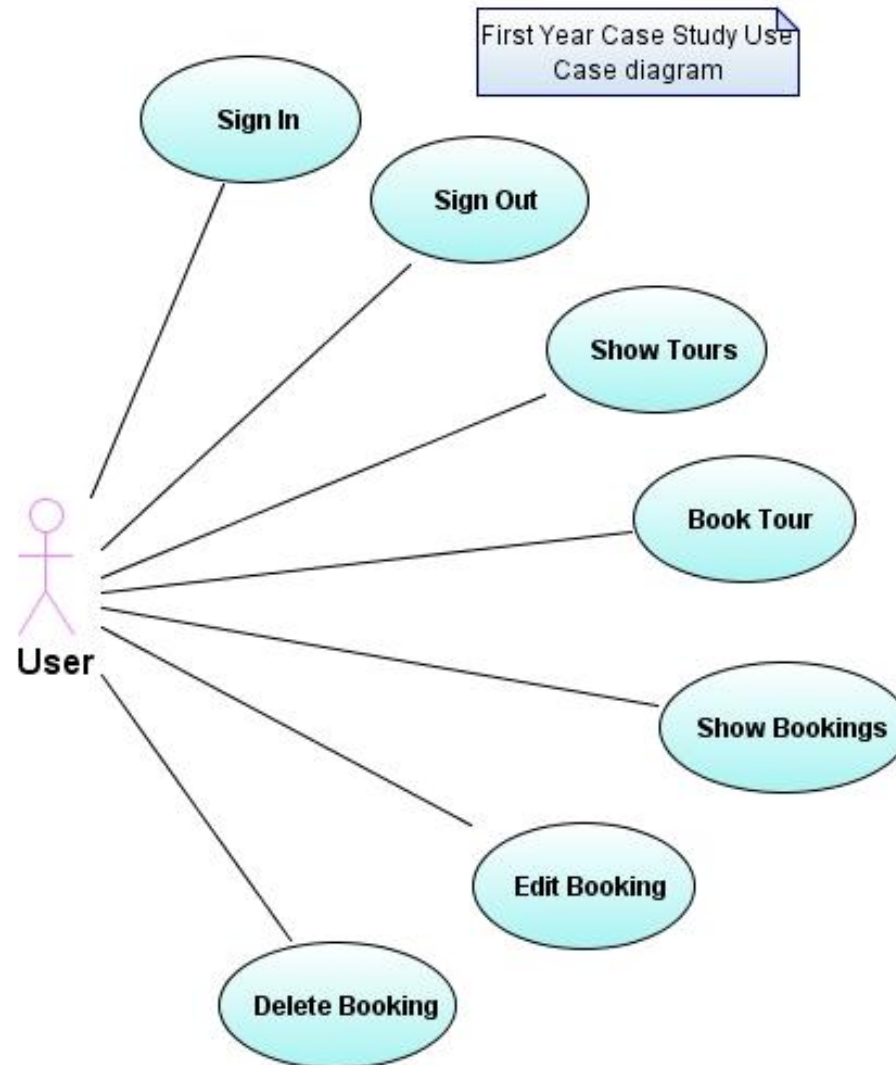
The First Year Case Study - 1

- **The case study is a simple 3-tier Web application.**
- **The application is an E-Commerce application that might be used by a travel company to allow the booking of travel tours.**
- **Case study clients are thin e.g. a Firefox or Internet Explorer Web browser.**

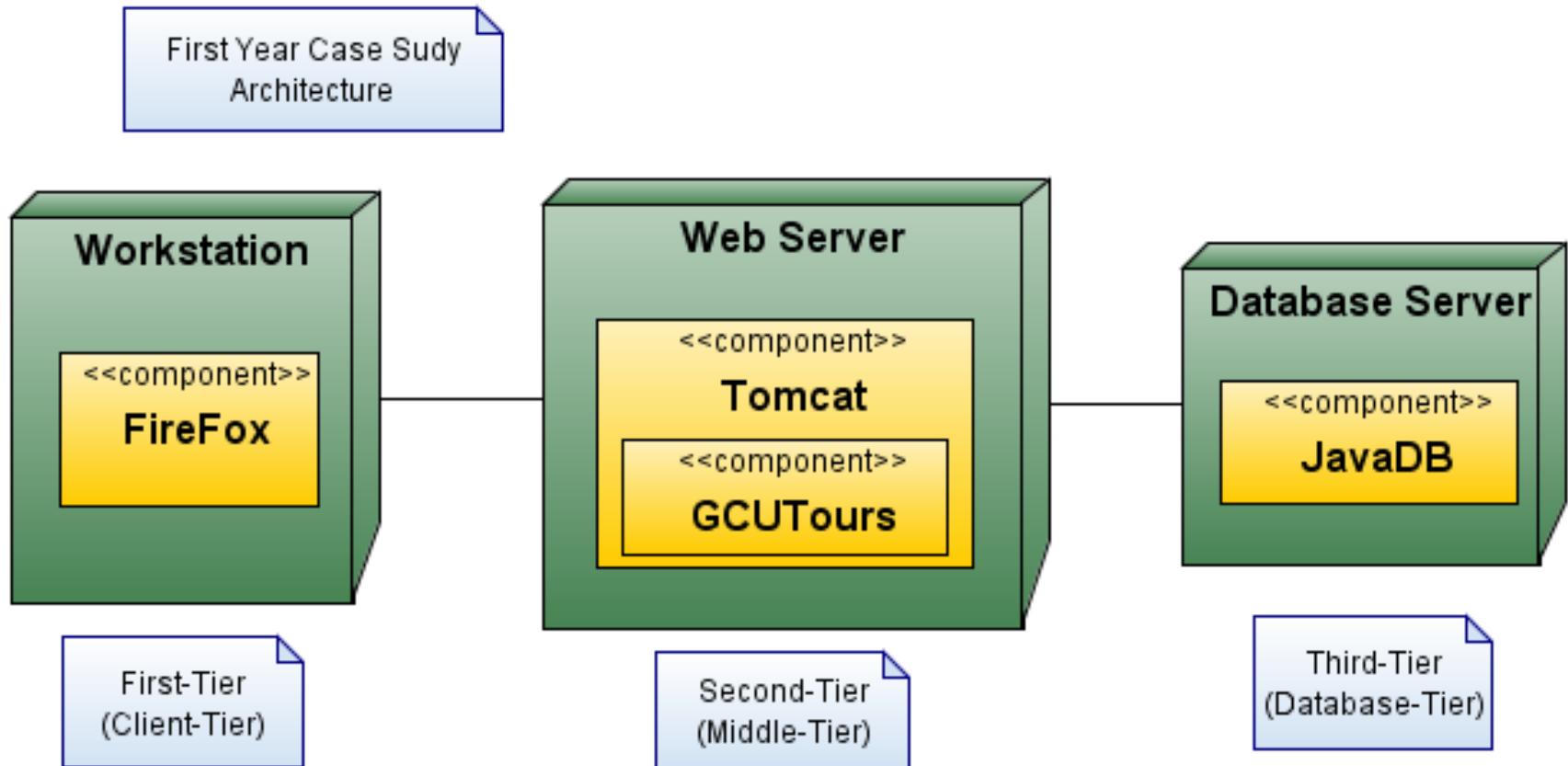
The First Year Case Study - 2

- **The middle tier is implemented using Java technology and is responsible for Presentation, Business and Data Access Logic.**
- **The middle tier uses a server called Tomcat that can serve Web pages and can also execute code in response to client requests.**
- **The JavaDB database server is the third tier**

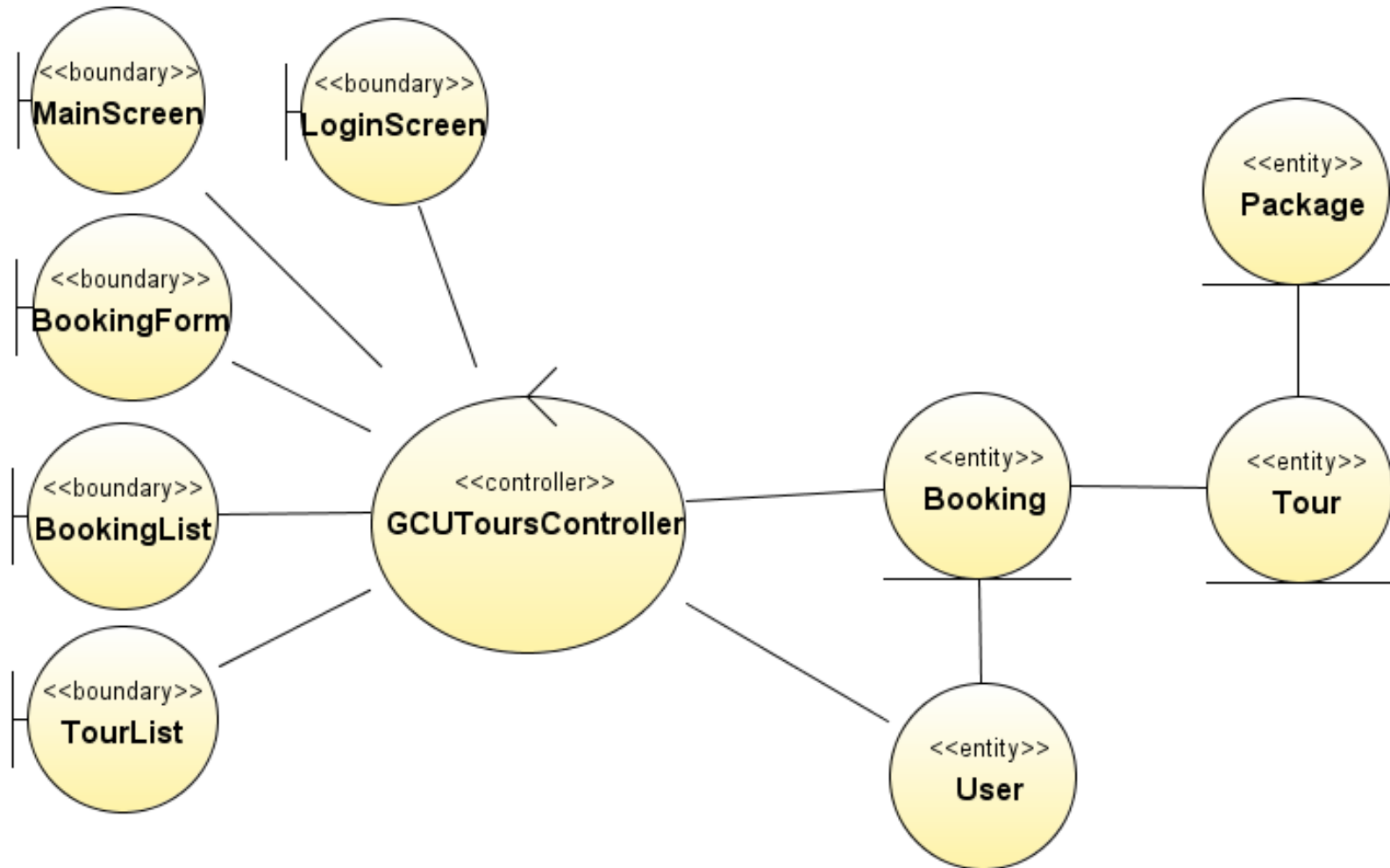
The Use Case Model



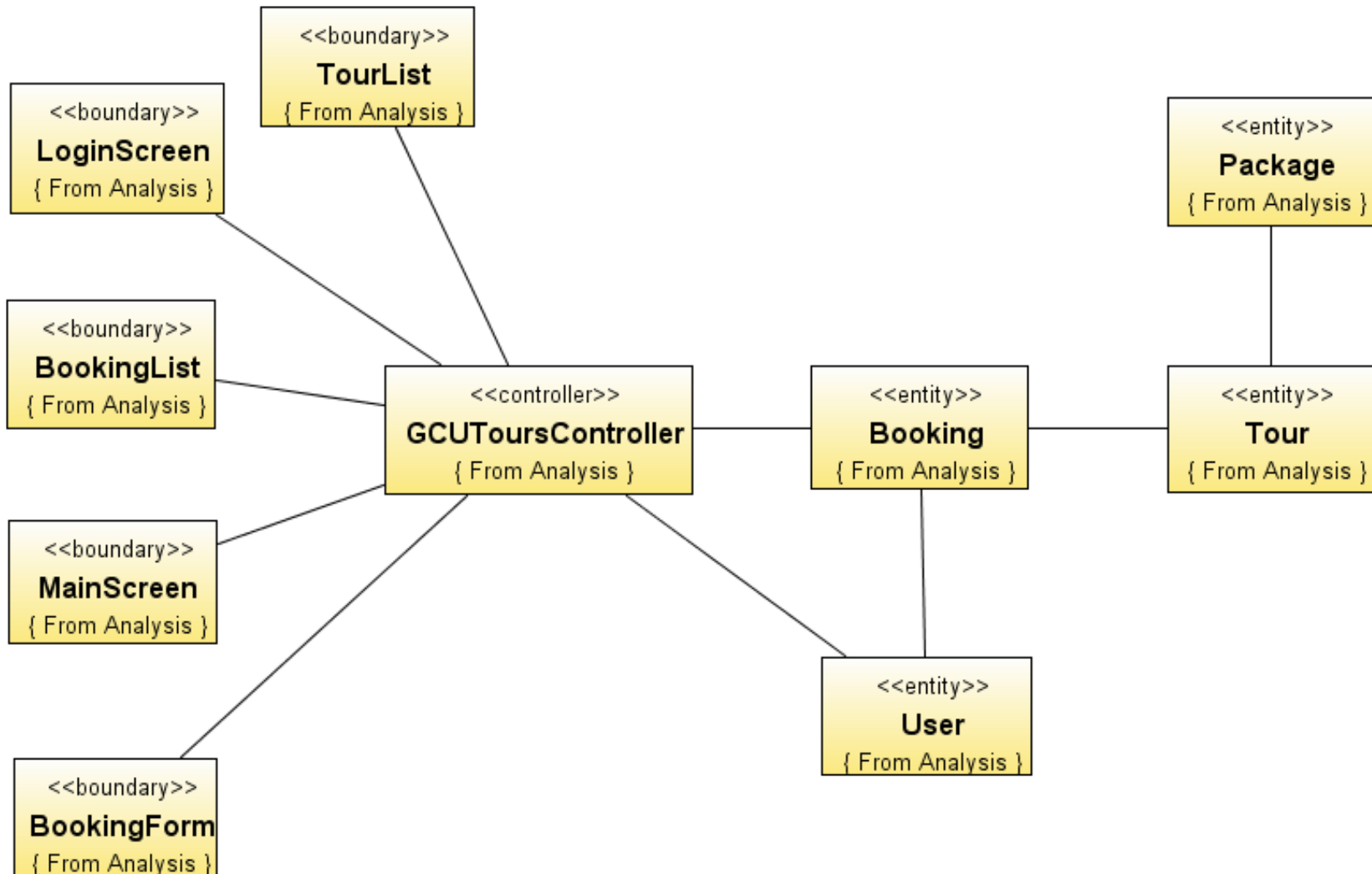
The Architecture



The Analysis Model (icon)

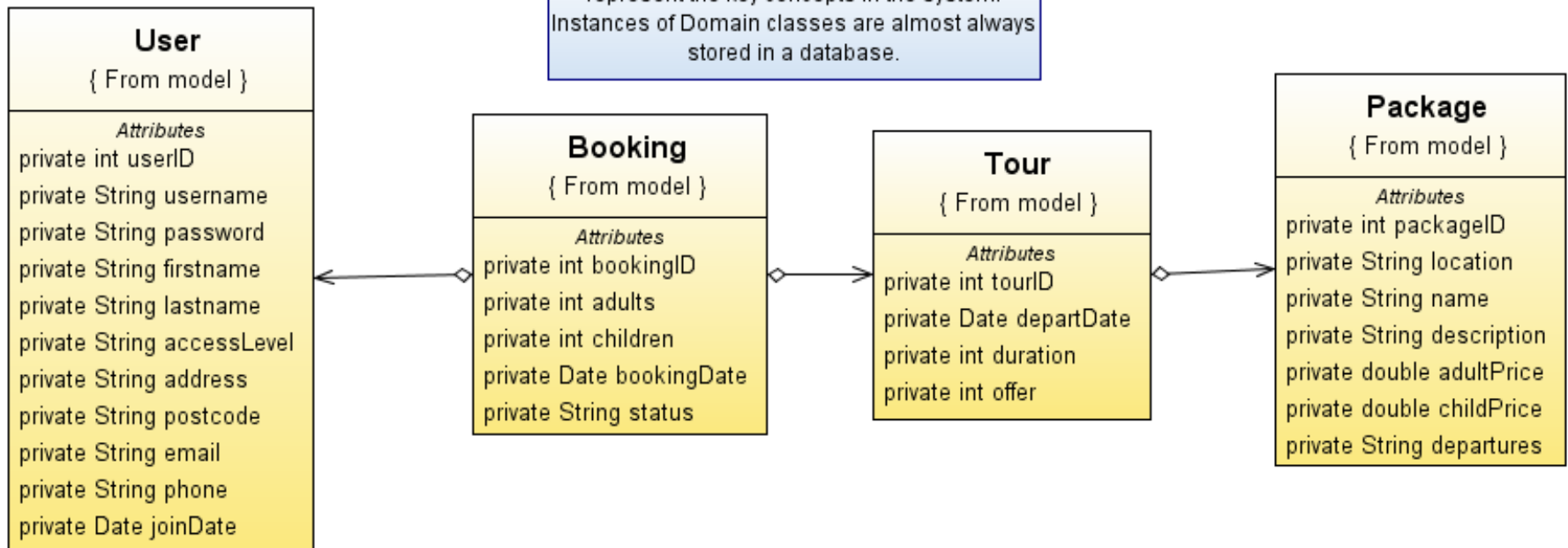


The Analysis Model (non icon)



The Domain Classes

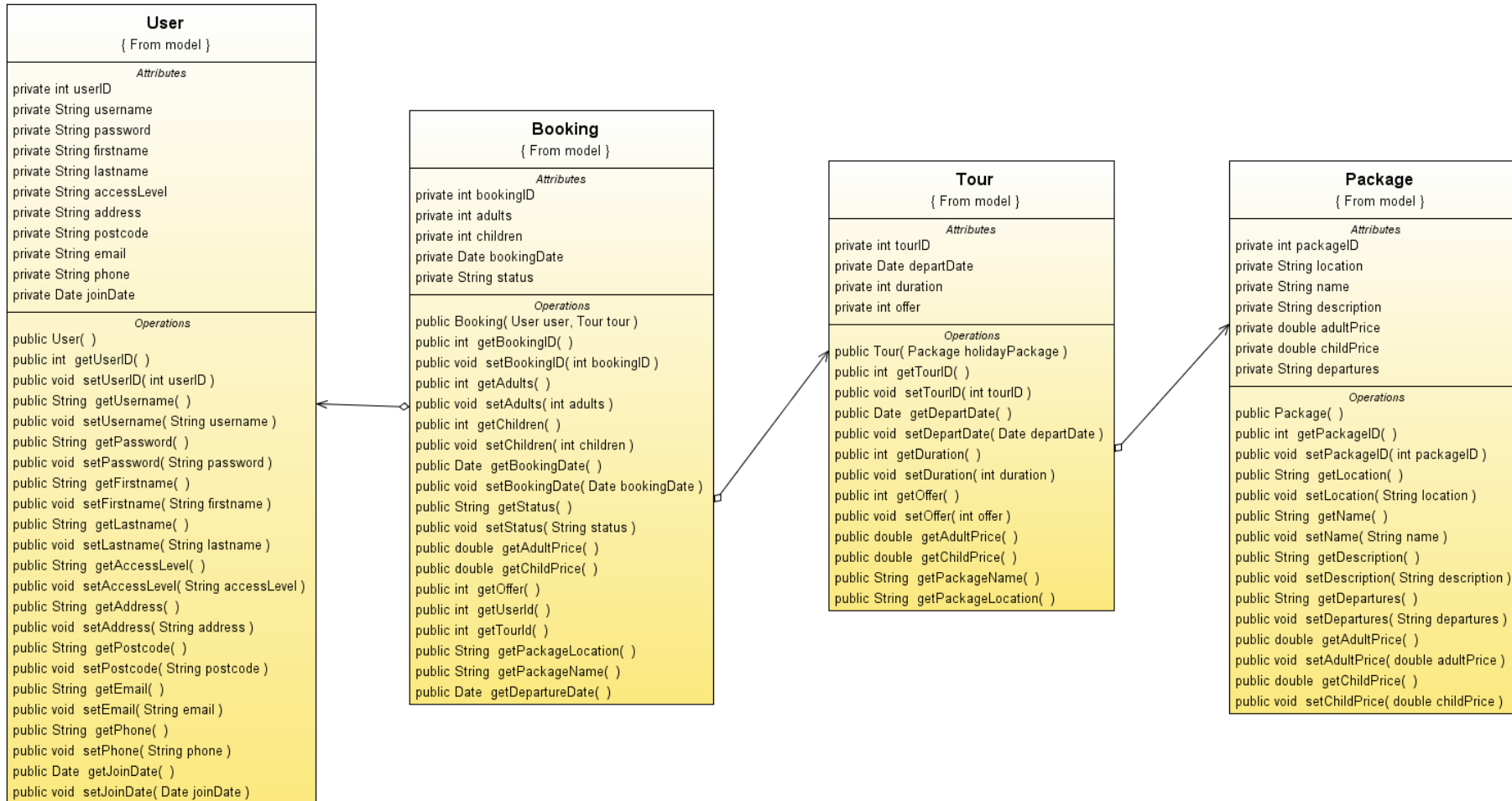
The Domain Classes i.e. the classes which represent the key concepts in the system. Instances of Domain classes are almost always stored in a database.



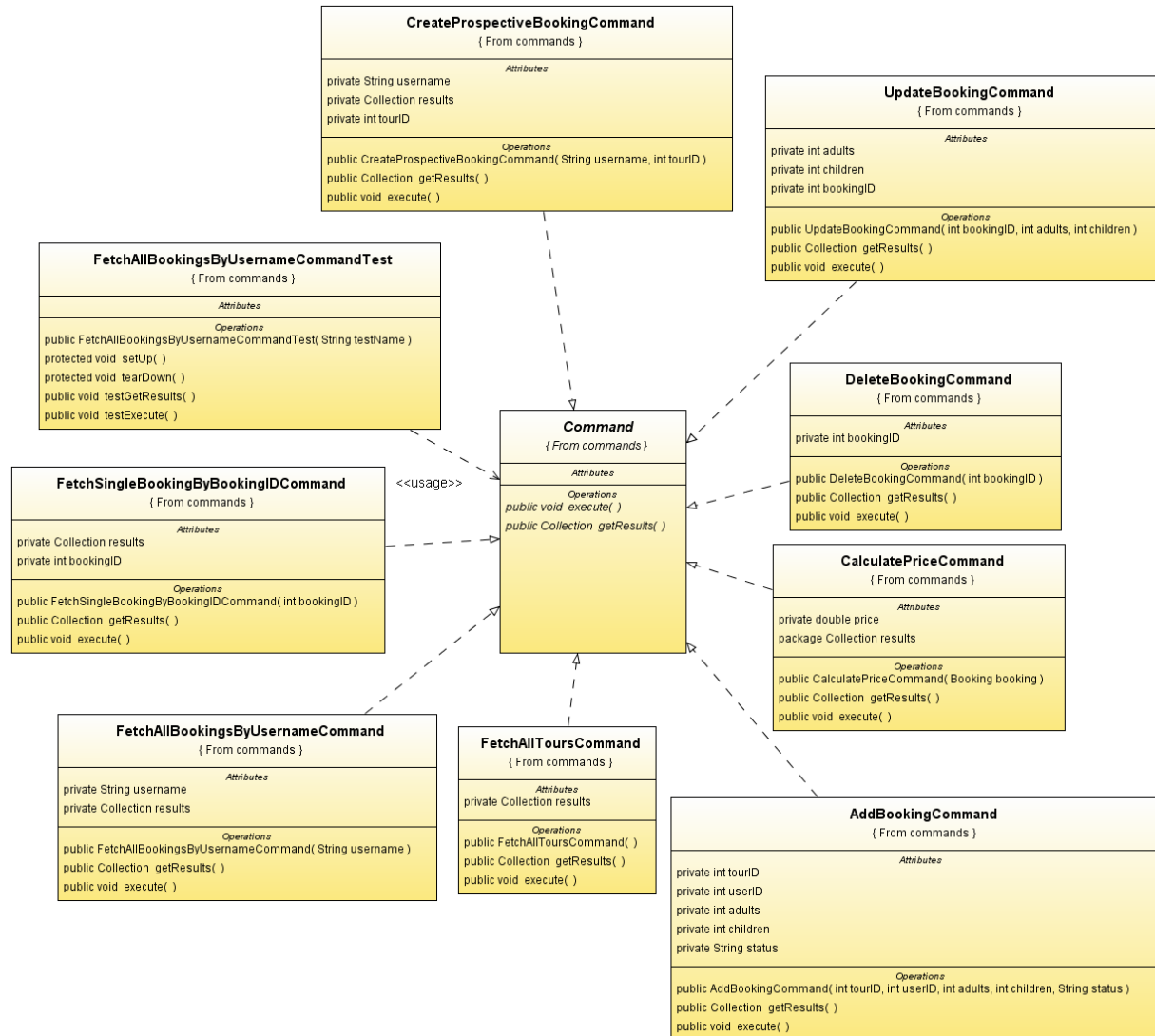
The JavaDB Database Tables

- **The JavaDB database has Four tables to support the saving of instances of the User, Booking, Tour and Package classes**
- **The tables are:**
 - **USERS** for instances of the User class
 - **TOURS** for instances of the Tour class
 - **BOOKINGS** for instances of the Booking class
 - **PACKAGES** for instance of the Package class
- **A single row in each table is used to hold each class instance.**

Design Model 1 Domain Classes



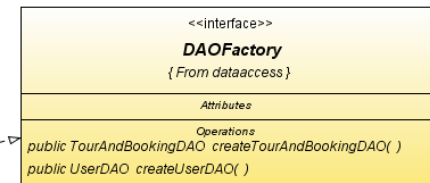
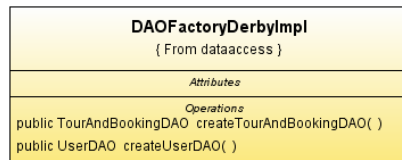
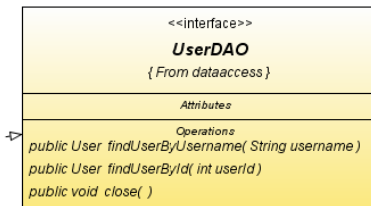
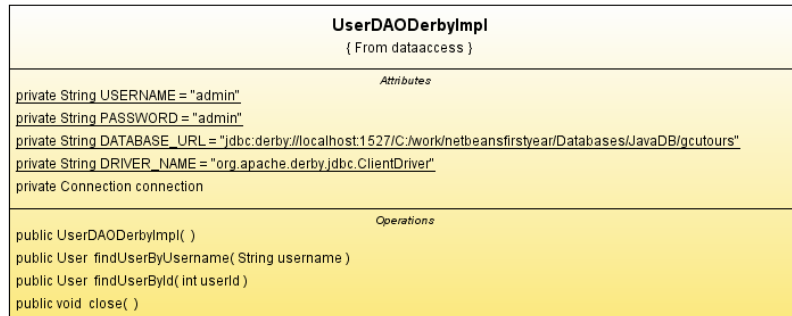
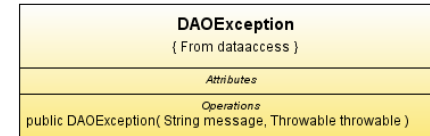
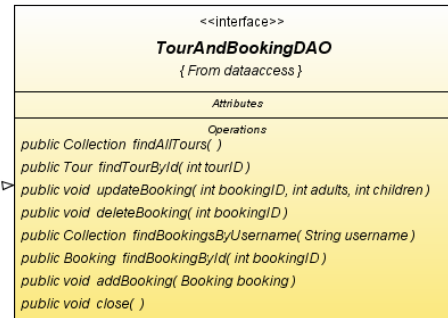
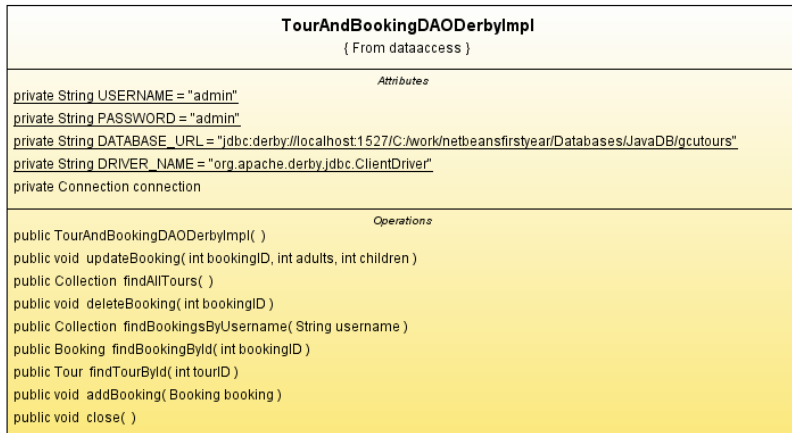
Design Model 2 Commands



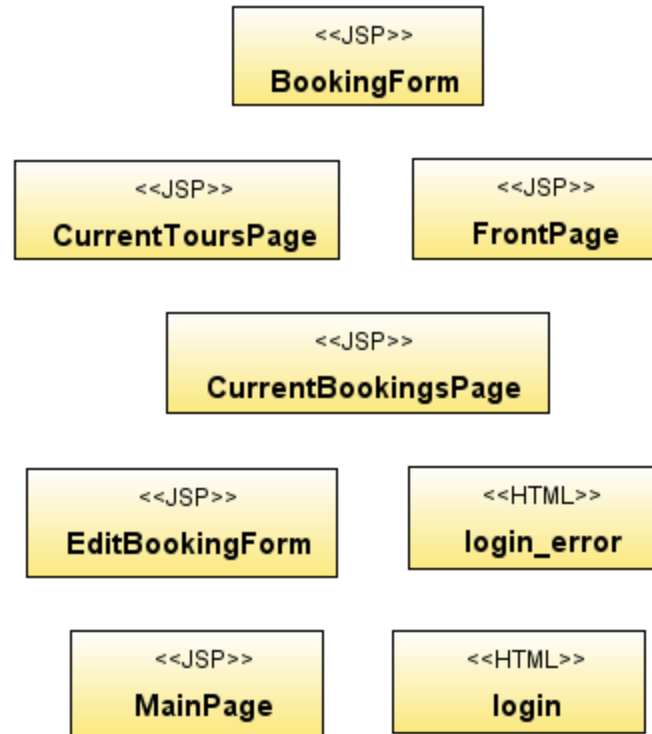
Design Model 3 Controllers

FrontControllerServlet { From controllers }
<i>Attributes</i>
<i>Operations</i>
<pre>protected void processRequest(HttpServletRequest request, HttpServletResponse response) private void showBookings(HttpServletRequest request, HttpServletResponse response) private void showEditBookingForm(HttpServletRequest request, HttpServletResponse response) private void updateBooking(HttpServletRequest request, HttpServletResponse response) private void deleteBooking(HttpServletRequest request, HttpServletResponse response) private void signOut(HttpServletRequest request, HttpServletResponse response) private void showBookingForm(HttpServletRequest request, HttpServletResponse response) private void addBooking(HttpServletRequest request, HttpServletResponse response) private void showMain(HttpServletRequest request, HttpServletResponse response) private void showTours(HttpServletRequest request, HttpServletResponse response) private void showView(HttpServletRequest request, HttpServletResponse response, String view) protected void doGet(HttpServletRequest request, HttpServletResponse response) protected void doPost(HttpServletRequest request, HttpServletResponse response) public String getServletInfo()</pre>

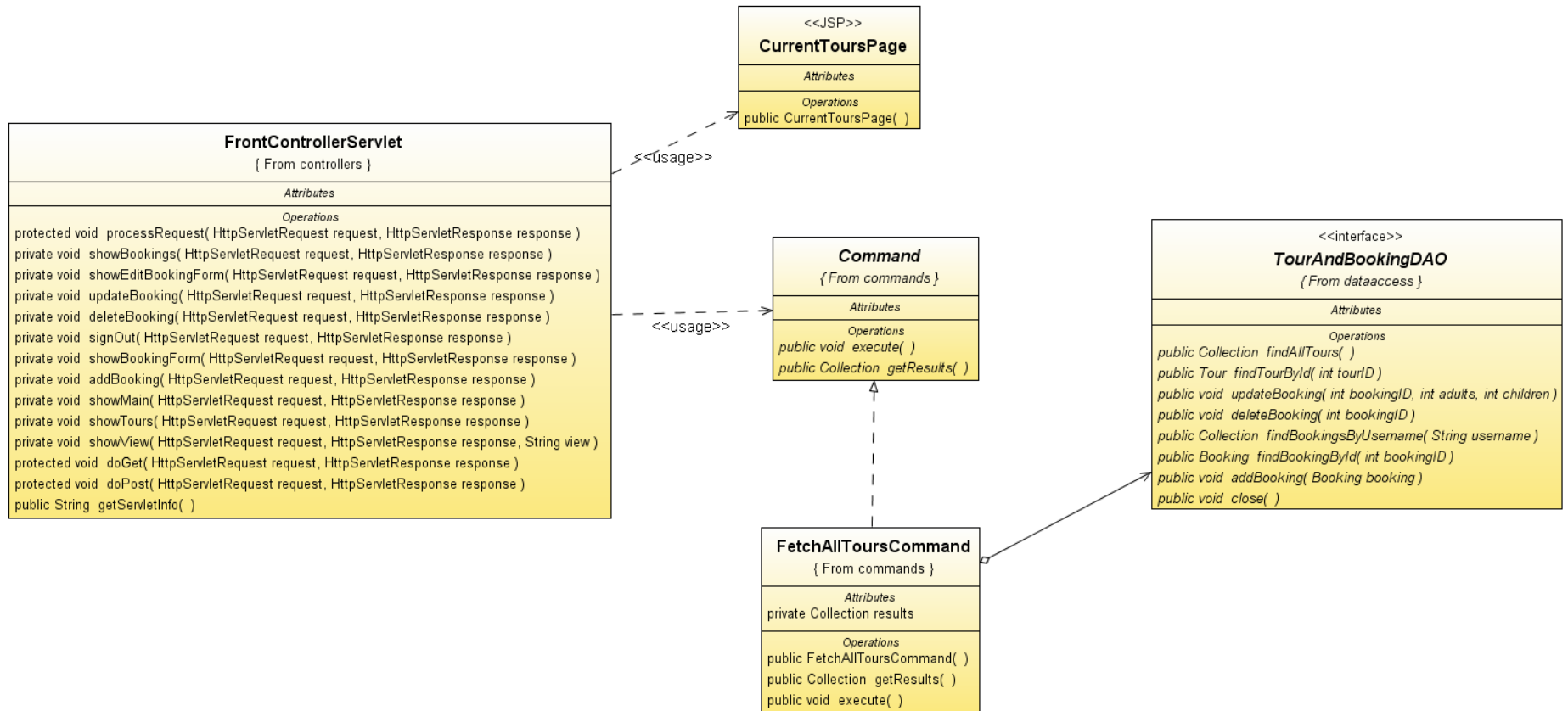
Design Model 4 Data Access



Design Model 5 User Interface



Design Model 6 Misc





Case Study Demo