

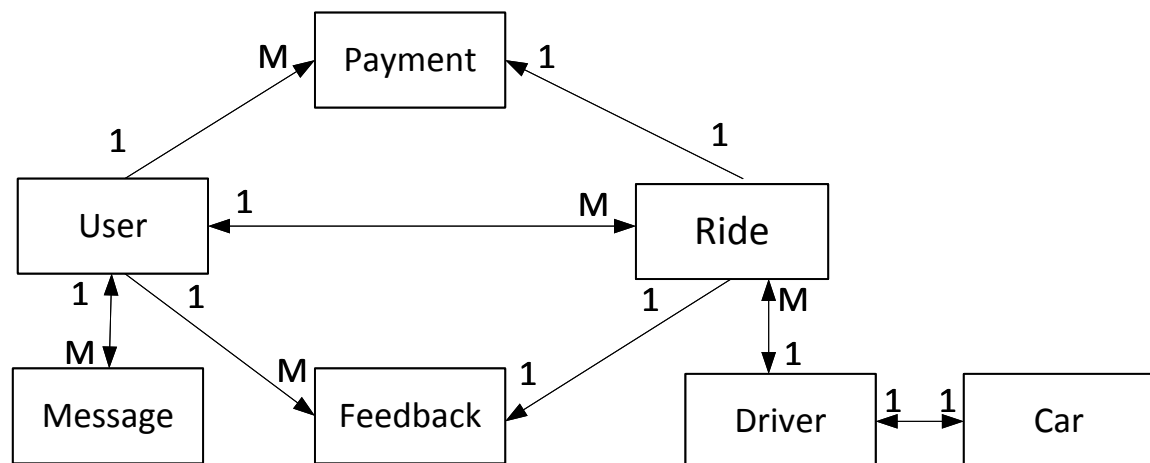
Part 2: Enterprise JavaBeans TBS: Taxi Booking System

Total Marks: 30

DOMAIN INFORMATION

PickMe is a new taxi company in Singapore. Due to the immense popularity of online taxi service apps, PickMe intends to implement a new Taxi Booking System (TBS) app to increase the market share. The app will help the customers to book taxi rides, taxi drivers to update status and the admin staff to monitor the services.

From a detailed study of the system, an entity relationship diagram is provided:



Each entity in the entity relationship diagram has a set of attributes (i.e. properties or data fields) that describes its static information. The list below defines the attributes. *Note that the list does not include data fields for managing entity relationships. The set of attributes provided is sufficient for this assignment.*

User

- User Name (unique identifier)
- Password¹
- Contact Number
- Email
- Address

Driver

- Driving License Number (unique identifier)
- Name

¹ There is no requirement for the password to be encrypted in this assignment

- Contact Number
- Address
- Current Latitude²
- Current Longitude

Car

- Registration Number (unique identifier)
- Car Brand
- Car Model
- Manufactured Year

Ride

- Ride Id (unique identifier)
- Start Time
- Start Latitude
- Start Longitude
- End Time
- End Latitude
- End Longitude
- Fee

Payment

- Payment Id (unique identifier)
- Payment Time (date and time of payment)
- Card Type
- Card Number
- Card Holder Name

Feedback

- Feedback Id (unique identifier)
- Feedback Time (date and time of request)
- Rating³
- Comment

Message

- Message Id (unique identifier)
- Time (date and time of message)
- Content
- Status
- Comment

SYSTEM REQUIREMENTS

Part 2a: Taxi Booking System (for the system administrator) (14 marks)

1. Given the above domain information, produce an EJB 3.0 application for the Taxi Booking System with a *non-web-based client administration application*.
2. The application will be called “TBS: Taxi Booking System”.
3. The entities given above are sufficient for your application. You DO NOT HAVE TO CREATE ADDITIONAL ENTITIES FOR THIS ASSIGNMENT.
4. The client application that you build will be used to test your EJB 3.0 application. For evaluation purpose, you are required to produce the following functionalities⁴ in the client application⁵:

² All latitude and longitude fields are integer values with each in the range of 0 to 100

³ Integer rating value of 1 to 5 stars

⁴ For part 2a and 2b, all validation checks for data input should be implemented. For example, the system should not accept string values for integer fields or a negative value for fields such as manufactured year. In addition, proper error messages should be displayed for all errors and exceptions. Further, if a functionality call doesn't return any results, a suitable message should be displayed.

- a. Add user accounts: Allows an administrator to create new user accounts for new customers. If the account already exists, an error message will be shown.(1 mark)
- b. Delete user accounts: Allows an administrator to remove user accounts from the system. User name is used as input. If the user does not exist, an error message will be shown. A user account cannot be deleted if it is associated with rides, payments, feedback or messages.(1 mark)
- c. Add car: Allows an administrator to add new cars into the system. If the car already exists a suitable error message should be shown.(1 mark)
- d. Delete car: Allows an administrator to remove cars from the system. Registration number is used as input. If the car does not exist, an error message will be shown. A car cannot be deleted if it is associated with a driver.(1 mark)
- e. Add driver: Allows an administrator to add a new driver into the system. If the driver already exists an error message is shown. **A driver must associate with one car**.(1 mark)
- f. Update driver: Allows an administrator to update driver information. The driving license number is used as input. If the driver does not exist, an error message will be shown. **If the driver is associated with a ride; driving license number and the associated car cannot be changed**. Otherwise, all fields can be changed. (2 marks)
- g. Delete driver: Allows an administrator to remove drivers from the system. The driving license number is used as input. If the driver does not exist, an error message will be shown. A driver cannot be deleted if it is associated with a car or rides. (1 mark)
- h. Feedback report: Allows an administrator to get a feedback report on drivers. The driving license number of the driver is used as input. A summarized report should show the counts of each feedback category (i.e. how many 1 star feedbacks, how many 2 star feedbacks and so on). (1 mark)
- i. View idle drivers: Allows an administrator to view the currently idle drivers who are not on a ride. The report should show the driver's license number, name, contact number and current location.(1 mark)
- j. Monthly performance report: Allows the administrator to view the performance report of each driver. The report should show the total revenue earned, number of rides, total ride time and the total distance of the rides of each driver for the current month. (1 mark)
- k. View unpaid rides: Allows an administrator to view detailed information of all the rides currently not paid by users. In addition to the details of the ride, the details of the associated user (username, contact number and email) should be shown.(1 mark)
- l. Process messages: Allows an administrator to view messages sent by the users. The list of messages is displayed on a first-come-first-served basis. The administrator can select a message by message Id and update the message status from "unread" to "processed". All "processed" messages will not be displayed in the next query. The administrator can also add some comments to the message. (2 marks)

⁵The client application execution flows is left for your own design. Bonus marks will be given for good user interface design.

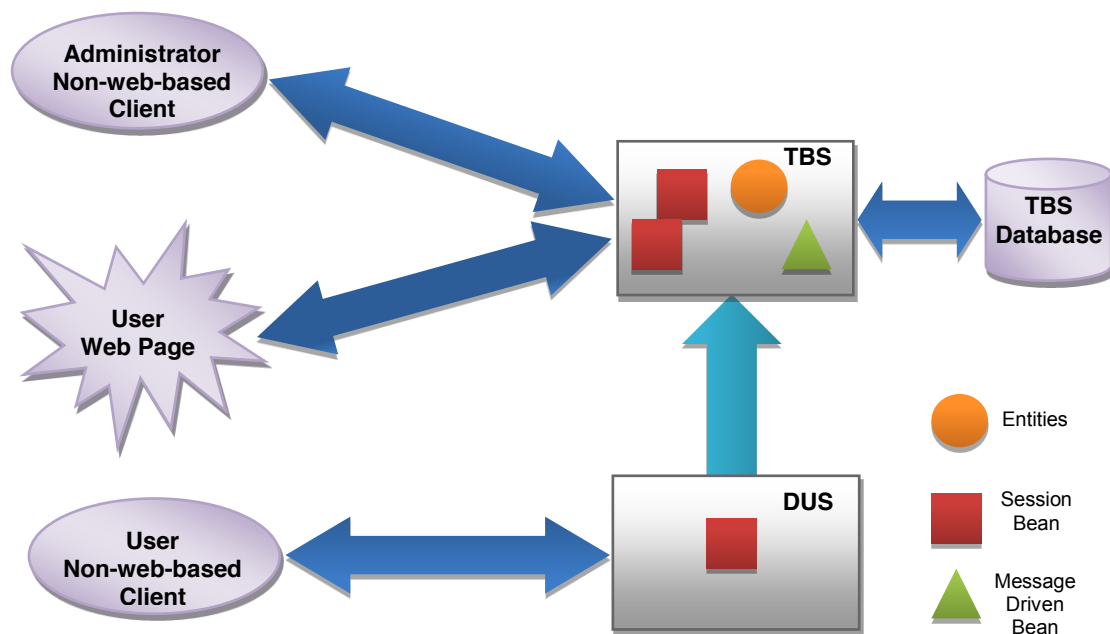
Part 2b: Taxi Booking System (for the system users/customers) (12 marks)

1. Produce a *Web-based client application using JSP and Servlet* for the same Taxi Booking System in Part 2a.
2. The entities given above are sufficient for your application. You DO NOT HAVE TO CREATE ADDITIONAL ENTITIES FOR THIS ASSIGNMENT.
3. The Web-based client application will be used concurrently with the non-web-based client application to test your EJB 3.0 application. For evaluation purpose, you are required to produce the following functionalities in the Web-based client application especially designed for customer's use:
 - a. Register as a member: A user can register himself/herself as a user by providing his/her particulars.(1 mark)
 - b. Login to the system: A user has to login using the user name and password before he/she can use the system functionalities.(1 mark)
 - c. Update profiles: A user can change the password and update his/her profile (email, contact number, address) on the system.(1 mark)
 - d. Book ride: A user can book a ride with this feature. In order to book a ride, the user should enter the current location as latitude and longitude. The closest driver who is currently idle will be assigned to the user. Refer appendix on distance calculation. Users cannot book rides if they have unpaid prior rides or is currently on a ride. If a suitable driver cannot be found, the user should be notified by the system with a suitable message. Once the booking succeeds a record for the ride is created on the database. The start time of the ride will be time of booking. The start latitude and longitude of the ride is the user's location. (2 marks)
 - e. Cancel ride: The user can cancel rides that are booked. If the cancellation is within two minutes of booking, no fee will be charged. If the cancellation is after two minutes, a fee of \$5 will be charged. The end time of the cancelled ride will be the cancellation time. The end latitude and longitude of the cancelled ride will be same as the start latitude and longitude. (1 mark)
 - f. View previous rides: A user is able to view a list of all past rides. For each ride, in addition to the details of the ride, the details of the driver (name and contact number) and feedback details (if available) should be shown.(1 mark)
 - g. Facilitate payment: The user can pay for his/her rides by providing credit card information (card type, card number and holder name). The user must pay only the exact amount of the respective ride. Any payment that is less than or exceeding the ride amount or multiple payments, should be rejected with a suitable error message. (2 marks)
 - h. Provide feedback: The user can provide feedback to his/her rides. In order to provide the feedback, the ride should be fully paid. (1 mark)
 - i. View payments: A user is able to view details of all past payments on this page. (1 mark)
 - j. Facilitate sending and viewing messages: A user is able to send messages to contact the administrative staff. The user should be able to view the content, status and any comments of his/her previous messages. (1 mark)

Part 2c: DUS: Driver Update System (4 marks)

1. Produce another EJB 3.0 application on a **separate NetBeans project** to facilitate driver updates.
2. The application will be called “DUS: Driver Update System”.
3. This application will have a **non-web-based client** to facilitate sending updates.
4. No login is required. A driver only needs to input his/her driving license number to access the system. The DUS will support two functionalities as explained below.
 - a. Update current location: The driver can update the current location by providing the latitude and longitude.
 - b. End current ride: If the driver is on a ride, the driver can end the ride by providing the current latitude and location. Upon ending, the fee for the ride will be calculated. Refer appendix for fee calculation.
5. DUS has a session bean that receives the requests from the users and sends a message to TBS. A message-driven bean in TBS receives the requests and updates them into a database. For simplicity, no acknowledgement is required.

Pictorially, TBS and DUS communication is as follows:



IMPLEMENTATION

1. In your implementation of the TBS, **your system should have the following data prior to evaluation**:
 - a. At least 3 users.
 - b. At least 3 drivers and 4 cars.
 - c. At least 5 rides, 2 payments, 1 feedback.
 - d. At least 2 messages.

EVALUATION

1. There will be an evaluator assigned to assess your work during the assignment evaluation week.
2. You will be required to deploy the EJB application on a computer in the laboratory or on your own laptop.
3. Tests on the functionalities of the non-web-based and web-based client applications will be carried out by the evaluator simultaneously. The evaluator will be examining the source code of the client and server logs of the EJB 3.0 application to ascertain the proper use of Enterprise JavaBeans 3.0 concepts.
4. Note that **all database queries within your EJB application should be done using EJB-QL. Marks will be deducted for the use of native SQL in database queries.**
5. Marks⁶ will be allocated in the following manner for all *executable* applications:
 - a. TBS Functionalities (up to a maximum of 26 marks)
 - TBS for administrator (up to 14 marks)
 - TBS for user (up to 12 marks)
 - b. DUS Functionalities (up to a maximum of 4 marks)
6. Bonus marks: Based on additional features such as good and nice web page design, advanced search functions, encrypted passwords, or any function that has not been specified above, up to a maximum of 3 bonus marks will be awarded.
7. If your EJB 3.0 application is unable to deploy and therefore not executable, up to 3 marks will be awarded for your Part 2 submission.
8. If your EJB 3.0 application is able to deploy and is executable, but none of the functionalities required in Part 2 works properly, up to a maximum of 5 marks will be awarded for your Part 2 submission.
9. Please be reminded that plagiarism is a serious offence. Disciplinary actions will be taken against those caught cheating.

APPENDIX

- For simplicity, each of latitude and longitude will be an integer value in the interval $[0, 100]$ (i.e. integer values between 0 and 100 including 1 and 100). Non integer values and values not within the range should be rejected by the system.
- The distance between two points will be Euclidean distance.
 - If the two points are $p = (x_1, y_1)$ and $q = (x_2, y_2)$,
 - $distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$,
 - x_1, x_2 are the latitudes and y_1, y_2 are the longitudes of the two points.
- Fee calculation will be based on the distance travelled and the time taken for the ride. Fee will be a double value rounded to two decimal points.
 - $fee = (distance \times 1) + (time \text{ in minutes} \times 0.1)$

⁶ A student can score up to a MAXIMUM of 30 marks for Part 2 (including bonus marks).