BUE
The British University in Egypt
الـجامعة البريطانية في مصر

Group 9 Phase1 Feedback

General Comments	- Good Job! - The class diagram should be consistent with the Netbeans project.
Class Diagram	• There is no relationship between class "Passenger" and "Train". • There should be a relationship between class "Admin" and "Train". • You should represent the booked tickets in class. You may add a class "Booked_Tickets" in-between passenger and Ticket.
General Class implementation • Relevant attributes • Inheritance/Aggregation/Composition/Interface • Implemented based on Object Oriented Concepts.	• Timeslot should be represented as an Arraylist. • Routes should be represented as an Arraylist. • Arraylist of tickets/seats should be in class Route.
Methods that should be provided for Admins	*Consider passing parameter for your methods. Ex. "Add Train()"*
• Manage Account - Search for a specific admin from the collection and update the required information.	- √
• Add Train: - The method should take the information of the new train. - Add the train as an object in the Trains collection.	- √
• Remove Train: - Search for the train to be removed within the collection. - Remove the train from the collection.	- √
• Update Train stored data: - Search for the specified train within the collection.	- √

- Update the train data.	
• Generate Reports: - The method should search in the booked collection for the booking of a certain route. - The method should calculate the total fare of this route by summing up the fares collected. - The method should display the total number of bookings and the total fare collected.	- √
Methods that should be provided for Passenger.	
• Create Account: - The method should add the new Passenger in the collection of Passengers.	- √ - You should implement this method in its class not in the main.
• Manage Account - Search for a specific passenger from the collection and update the required information.	- √
• Book a ticket: - The passenger should select origin and destination of a route, timing slot and seat number from the available seats. - The method should add the new booking to the Booked collection. - The passenger should be allowed to reserve more than one seat.	- √ - You should consider if the passenger needs to reserve more than one ticket. - You mentioned different types of seats (High, Medium, low) but you do not mention how the passenger will know the type of seat while booking.
• Cancel Booking: – The method should search the booked collection for the one to be canceled. – The canceled booking should be removed from the booked collection. – The seats should be made available again.	- √
• Update Booking: – The method should search the booked collection for the one to be updated. – Cancel booking method could be called	- √

(or it is implemented again). – Book a ticket method could be called (or it is implemented again)	
• View tickets: – The method should display for the passenger his list of booked tickets by displaying his collection of booked tickets.	- √
Proper use of Collections	
Collections needed: - A collection for Admins - A collection for Passengers - A collection for Trains - A collection of Train_routes - A collection of tickets_per_route. - A collection for Booked tickets (different implementations are accepted)	- A collection of Routes is needed. - A collection of tickets_per_route is needed.
Class Implementation and Testing in main	
• Login: - Displaying menus should be based on the correct login	- √
• Test all methods : - A scenario for calling 11 essential methods - Insert meaningful data for testing. - Using appropriate display methodology	Good job!