# Assignment 1 - Modern Taxi System

## General Information

This assignment represents **50% of the module**. You will work in **groups of 3 or 4**. There is an **individual element** to the assignment. This assignment addresses learning outcomes 1 and 3 from the module specification: “students should be able to identify and discuss the main aspects of UML and entity-relationship modelling, and use them to model system requirements” and “students should be able to discuss the appropriateness of object-oriented and data driven modelling for given system requirements”.

## Submission - 25th January - before 4pm

Your work will be submitted as a group via Bb. Your submission must include images of your models which do not require any special software to open (for example - “Class Diagram.png”, “ERD.png”). If the details of the models can not be viewed then we will not be able to mark your work - make sure your images are readable.

## Group Walkthrough

You are expected to justify and answer questions about your work during a group walkthrough shortly after submission. Failure to attend your walkthrough will result in 0 marks (unless there is an extenuating circumstance supported by documentary evidence). We will provide a Peer Assessment Form which you are to fill in as a group and hand in during the walkthrough. The peer review form will help us to assess individual contribution to the work and will be taken into consideration when marking. If you suspect any issues with group contributions during the assignment please contact the module leader as soon as possible.

# The modern taxi system

SHUber is modern taxi service making it easier to book a taxi.

Before using the service customers must download the app, signup, and register their card details. All payments are handled automatically by the app, meaning both customers and drivers need not carry cash. Handling the actual payment transaction is outside the scope of your assignment, but the cost of journeys must be recorded.

After signing up, customers can use the app to request a SHUber pick them up. The current location of the customer will be used as the pick up location, and they can manually set their drop off location.

Once a customer has created a request, the job will be accepted by a nearby SHUber driver and they will begin making their way to the pick up location. At this point the customer gets some extra information about their driver: they can track the driver’s location on a map, see their name, profile picture, average rating, and vehicle details.

When the driver arrives the customer will receive a notification from their app and they can then jump in the vehicle.

After the journey the customer receives a summary of the route taken, and the cost of the trip. A copy is also sent to their registered email address. For each job the driver will rate the customer out of 5 which will impact their average rating - this way other drivers will know if a customer is likely to cause trouble. The customer can also rate the driver after the journey to help SHUber maintain driver standards.

# Group Tasks (80 marks)

## Use Case Models (25 marks)

Identify the actors and use cases that form the SHUber system and create:

1. a use case diagram with all actors and use cases
2. a use case description with normal course and alternate courses

## Entity Relationship Diagram (25 marks)

Identify the entities and relationships needed in a database to effectively store SHUber’s data and produce a high quality, complete, entity-relationship diagram.

## UML Class Diagram (30 marks)

Identify the classes and relationships required to enable the SHUber system to function and produce a high quality, complete, UML class diagram.

# Individual Task (20 marks)

Investigate and write a report regarding two topics important to SHUber: methods for programmers to set up communication between the software and the database; and data protection (laws are changing in 2018). Each is worth 10 marks. The report should be in layman's terms, 1-2 sides of A4, and it must be referenced.

1. What is an ORM (object-relational mapping) and how does it differ to using a DAO (data access object)? What are the pros and cons for the programmer? How can SHUber use these technologies?
2. Which considerations should SHUber make regarding the usage and sharing of its data as far as the [Data Protection Act 1998](http://www.legislation.gov.uk/ukpga/1998/29/contents) is concerned?

# Marking

Please see the [Marking Grid](https://docs.google.com/a/my.shu.ac.uk/document/d/10GjQ3HEdkfbhuR8P8Stz49DvSrHfaCCypTnyRG3cMTQ/edit?usp=sharing) for more information. Advice is given below.

## What we’ll be looking for:

**Use Case Models and Use Case Descriptions** - chosen use cases and actors, associations between actors and use cases, use of include and extend, the system boundary, conformance to the use case description template provided, a realistic collection of alternate courses.

**Entity Relationship Diagram** - chosen entities, choice of attributes and primary keys, relationships, cardinality and naming of relationships, consistency between relationships and foreign keys.

**UML Class Diagram** - chosen classes, attributes and operations within each class, accuracy of relationships and multiplicities, logical positioning of classes to improve readability.

## Common mistakes:

**Use Case Models and Use Case Descriptions** - insufficient detail, missing actors, lack of clarity over where the boundary of the system responsibilities lie, no examples of include and extend, lack of understanding of the difference between include and extend, unclear formatting of use case descriptions, missing alternate courses.

**Entity Relationship Diagram** - missing entities, incorrect or irrelevant attributes, poor choice of primary and foreign keys, mistaken or unclear cardinality of relationships, incorrect modelling method - make sure to use crow’s foot notation.

**UML Class Diagram** - missing classes, errors in the relationships between the classes, lack of detail, unclear purpose or responsibilities of classes, lack of understanding of difference between this and ERD.