# 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”.

## 5th December, 3pm - Task 1 & 2

Your work will be submitted as a group via Bb. Your submission must include 5 files: UseCaseDiagram.png, UseCaseDescriptions.docx, EntityRelationshipDiagram.png, SampleData.png, and PeerAssessment.docx. The peer assessment form can be found [here](https://docs.google.com/document/d/1naf5rOP4J6Pugbr9zzLgeh6KO60PwVWphLuPzpFhC0U/edit?usp=sharing) and should be filled in as a group. If the images of your models are not legible then we will not be able to mark your work - make sure your images are full resolution and are readable.

## 27th January, 10am - Task 3 & Individual Report

Task 3 will be submitted as a group via Bb. Your submission must include ClassDigram.png and PeerAssessment.docx. The peer assessment form can be found [here](https://docs.google.com/document/d/1naf5rOP4J6Pugbr9zzLgeh6KO60PwVWphLuPzpFhC0U/edit?usp=sharing) and should be filled in as a group. Again, make sure your image is readable. A separate individual submission point will be available for the individual report.

# 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%)

## Task 1 - Use Case Model and Descriptions (25%)

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

1. a use case diagram with all actors and use cases (15%)
2. two use case descriptions with normal course and alternate courses (10%)

## Task 2 - Entity Relationship Diagram (25%)

Identify the entities and relationships needed in a database to effectively store SHUber’s data and produce a high quality, complete, entity-relationship diagram. You should also include a few rows of sample data for each entity.

## Task 3 - Class Diagram (30%)

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

# Individual Task (20%)

Investigate and write a referenced report, around 2 sides of A4, regarding two topics:

1. The data protection laws changed recently. What considerations should SHUber make regarding the usage and sharing of its data? (15%)
2. Relate to your models and explain what happens when requesting a taxi. What processing happens on your mobile device? What information is created and modified? How is a driver and route selected? (5%)

# 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 presented, 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 the difference between this and ERD.