# Instructions

* Ensure you have forked the Trip Planner repository from the SAD Diploma GitHub organisation
* Ensure you have cloned your fork to your desktop
* Ensure you have a copy of the Visual Paradigm Trip Planner project

## Data Access

1. All work for this section should be under a new feature branch called **data-access-model**
2. Create the data access model in the class library **TripPlanner**. Ensure you use an appropriate folder structure

Note: You can find the Entity Relationship Diagram for the database structure in the Visual Paradigm project, under **Physical Model**

1. As soon as you have created your first code-first class:
   1. Do a commit to the **data-access-model** branch
   2. Push your commit to your fork on GitHub
   3. Open a pull request from your **data-access-model** feature branch to the main repository
2. Ensure all code-first classes are created, and you have a context class called **TripPlannerDbContext**

## Services

1. All work for this section should be under a new feature branch called **trip-service**. Create this branch from **data-access-model**
2. Create the services in the class library **TripPlanner**. Ensure you use an appropriate folder structure

Note: You can find the Class Diagram for the services layer in the Visual Paradigm project, under **Application Services**

1. As soon as you have created the class for **TripApplicationService**:
   1. Do a commit to the **trip-service** branch
   2. Push your commit to your fork on GitHub
   3. Open a pull request from your **trip-service** feature branch to the main repository
2. Ensure all the methods are implemented correctly (your teacher will provide feedback via the pull request)
3. Ensure the class takes **TripPlannerDbContext** as a constructor parameter

Testing

1. All work for this section should be under a new feature branch called **trip-service-tests**. Create this branch from **data-access-model**
2. Create test services in the class library **TripPlanner.Tests**. Ensure you use the appropriate **project.json** dependencies from xUnit website. <https://xunit.github.io/docs/getting-started-dotnet-core.html>
3. As soon as you have created and completed the class for **TripApplicationServiceTests** including tests for createTrip, UpdateTrip and DeleteTrip.
   1. Do a commit to the **trip-service-test** branch
   2. Push your commit to your fork on GitHub
   3. Open a pull request from your **trip-service-test** feature branch to the main repository
4. Ensure all the methods are implemented correctly (your teacher will provide feedback via the pull request)

## Angular

1. All work for this section should be under a new feature branch called **trip-ui**. Create this branch from **master**
2. Create the angular code in the **js** folder under **wwwroot**. This folder is located in the **TripPlanner.UI** project