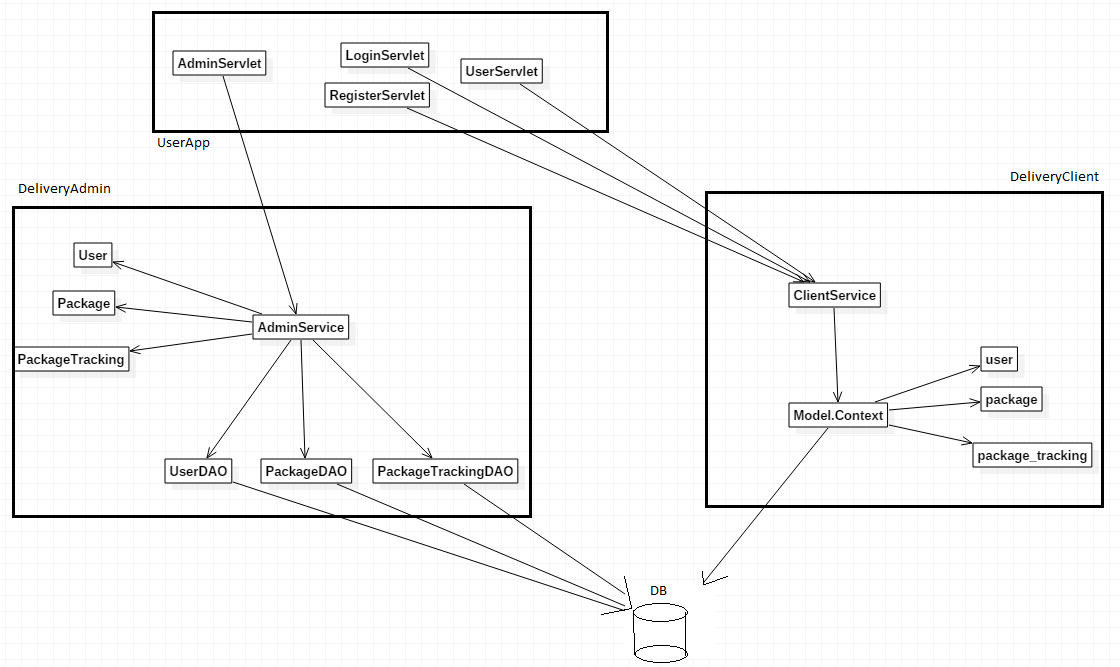
DISTRIBUTED SYSTEMS

Assignment 4

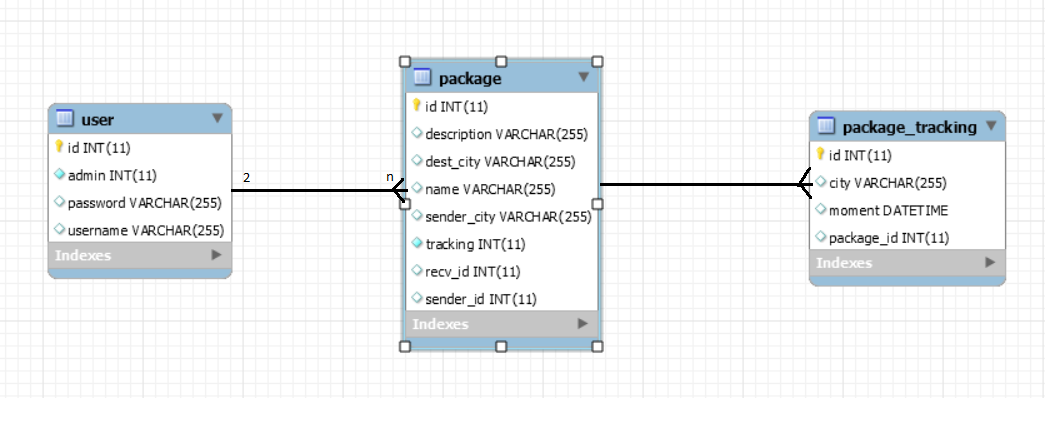
Service Oriented Distributed Systems

A4: SOA web services

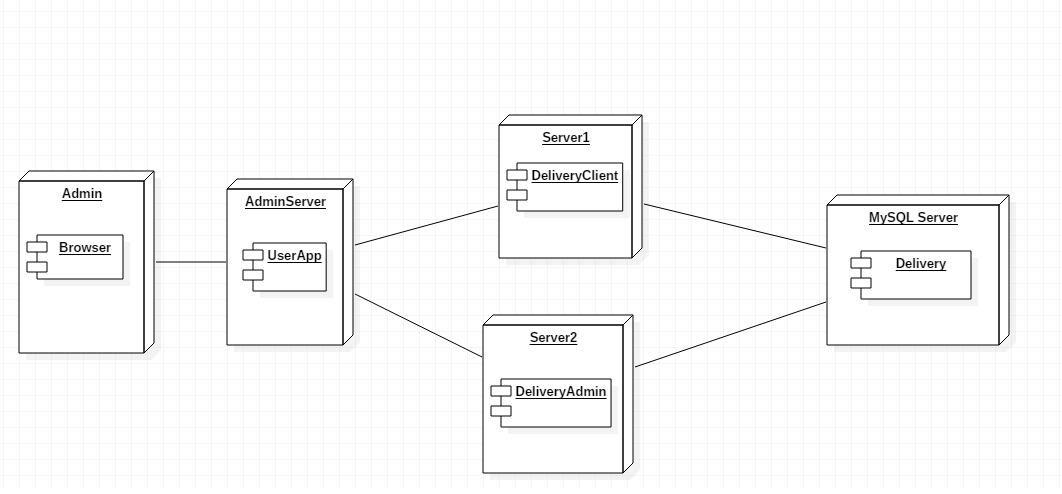
* Conceptual architecture of the distributed system

The distributed system is conceptually split into three components: the front-end UserApp, composed of the servlets that handle user interaction and input-output. There are four main servlets: LoginServlet (handles login), RegisterServlet (handles user registration), UserServlet (handles user input) and AdminServlet (handles admin input). The next component is the DeliveryClient part, composed of the ClientService class which exposes and implements the services exposed to regular users. This part is implemented in C#. The final component is the DeliveryAdmin part. The AdminService exposes and implements the services available to the admin, using the UserDAO, PackageDAO and PackageTrackingDAO classes to access the database.

* Database diagram:

The database is composed of three tables: the user table stores all the users of the system, the package table stores the packages (which have references to the sender and receiver, both users) while the package\_tracking table is used to composed the route of a package. Entries in the last table have references to the package table.

* Deployment diagram:

The administrator uses a browser from their computer to access the adminapp web application which resides on yet another computer. The application then calls one of the two available web services, each residing on different computers: ClientService and AdminService. Both these service access a database which could be stored remotely. Currently, everthing runs on the same machine but this could easily be changed with minimal effort.

* Build and execution considerations
  1. The DeliveryClient app:
     1. Open Visual Studio
     2. Open | Project/Solution
     3. Browse and select the project
     4. Go to “ClientService.cs”
     5. Click “IIS Express”
  2. The DeliveryAdmin app:
     1. Open IntelliJ IDEA
     2. File | Open
     3. Browse and select the directory containing the project
     4. Click “Open”
     5. Go to pom.xml file, double click to open it and right click on the editor
     6. Select Maven | Reimport to make sure that all the required libraries are included
     7. Open the hibernate.cfg.xml file from src/main/resources from the clientapp module
     8. Modify the appropriate fields with the data required to access the database schema
     9. Go to “AdminServicePublish”
     10. Run | Run “AdminServicePublish”
  3. The UserApp web application:
     1. Navigate to the “LoginServlet” class from the UserApp module
     2. Go to Run | Edit Configurations
     3. Click on the Green plus (+)
     4. From the dropdown list, select Tomcat Server | Local
     5. (Optional) Assign a name to the run configuration
     6. Change the field from Open browser to “http://localhost:8080/admin”
     7. Click on the “Fix” button from bottom right
     8. Select (Project name):war
     9. Click “Ok”
     10. Click Run | Run (Configuration name) – first option
     11. A browser page should automatically open with the default URL (<http://localhost:8080/> is default). If not, open a browser and go to the aforementioned URL.