MINISTRY OF EDUCATION AND SCIENTIFIC RESEARCH

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA

FACULTY OF AUTOMATION AND COMPUTER SCIENCE

COMPUTER SCIENCE DEPARTMENT

DISTRIBUTED SYSTEMS

Assignment 4

Scridon Beniamin
Grupa: 30643 TI

A4: SOA web services
using Java and .NET

Content

1. Requirements

Design, implement and test a distributed system that uses web services to expose the server functionalities to its clients

1.1 Functional requirements:

Consider a distributed application called “Online Tracking System” that has a GUI which exposes the following functionalities to its users:

- The application has two types of users: administrators and clients.
- After the login, the user is redirected to its corresponding page
- If the user does not have an account, it can register and become a simple user (client)
- The Administrator can:
    1. Add/remove package. The package has the following characteristics:
        o Sender – Client
        o Receiver – Client o Name
        o Description
        o Sender City
        o Destination City
        o Tracking – Boolean – initially false
    2. Register package for tracking
        o The package becomes tracked, and a route is associated to it. This route represents the path of the package to the destination, as pairs of (City, Time).
    3. Package status updating
        o A new entry (City, Time) is introduced to the route
- The Client can:
    1. List all its packages
    2. Search packages
    3. Package status checking

1.2 Implementation technologies:

- HTML, Java Servlets, Hibernate ORM, JSP, .NET, .NET MVC, Entity Framework ORM, Soap Services

2. Conceptual architecture of the distributed system

The application has 3 modules:

1. assignment4 JAVA maven module used for admin operations and contains the following:
    a. Entities package:
        i. BaseEnity class that has a fiels Id
        ii. Package class that extends BaseEntity and contains all the details about a package.
        iii. User class that extends BaseEntity and contains the basic information about the user and the user type.
        iv. RouteItem class extends BaseEntity and contains information about the route of the package.
    b. Repository package: Classes used for sending and receiving data to/from database using Hibernate Framework
    c. Service package that contains the application business logic.
    d. Dto package contains, dto classes sent by soap service(entities that contains Lists inside cannot be sent via soap).
    e. Assembler package map a class to its corresponding dto and a dto to a class.
2. Ass4 JAVA maven projectis the main application, users access this application via loginServlet at localhost:8080/ass4/loginServlet. This module uses the other 2 modules:
    - Assignment4_WS2 deals with basic user operations, login and register operations.
    - Ass4 deals with admit operations.

    Using Eclipse helper for web services are generated the classes for calling the 2 web services used, for each web service a proxy class is created and in order to call a web service method it is enough to instantiate the proxy and call the desired method.

    This module contains the following servlets: AdminAddPackage, AdminDeletePackage, AdminHomeServlet, AdminPackageServlet, AdminUpdateTracking, LoginServlet, LogoutServlet, RegisterServlet, UserPackageServlet, UserServet.
3. Assignment4_WS2 is a .NET MVC web project, and this module contains 2 web services:
    - LoginService for login and register operations

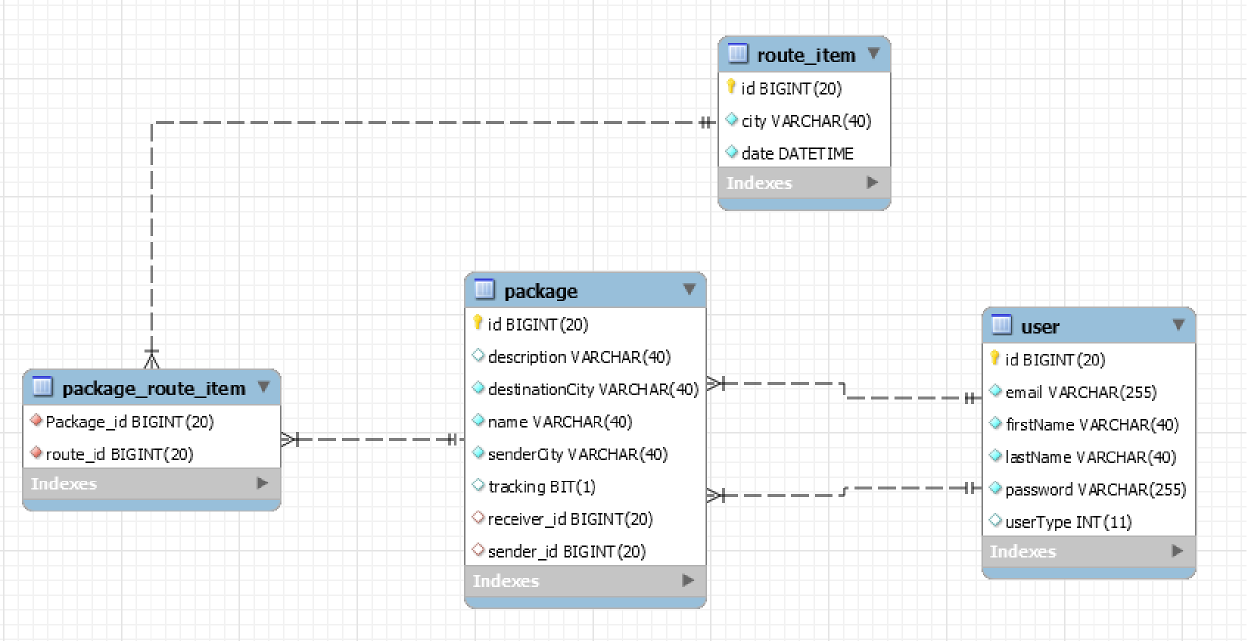
    - ClientService used for basic client operations: get packages for a user, getPackagesByNameOrId.
    Entity Framework is user as ORM framework.

3. Database

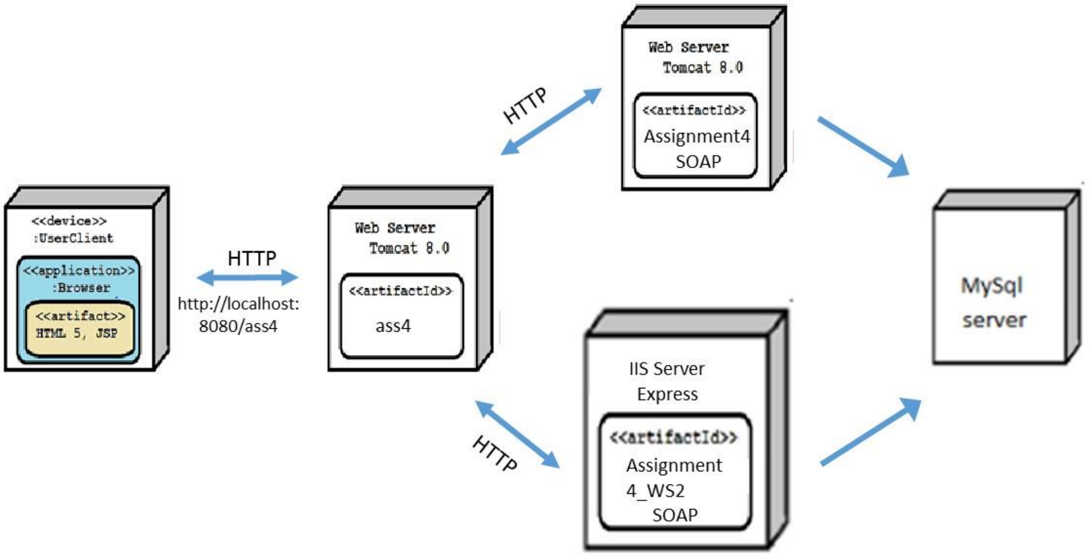
MySql database that has 4 tables:

- Package: contains information about a package
- User: contains the basic information about an user
- Route_item: each package has a route associated and in this table is kept an item of that list.
- Package_route_item is used to connect package to its route_items.

The schema looks like in the next picture:



4. UML Deployment diagram:



5. README file:

Git repository: https://rusiacob@bitbucket.org/rusiacob/sd_a1.2.git

Build:

JAVA:

* IDE(ex: Eclipse, Intellij etc.)

* Apache Maven installed: maven clean, maven install

*JDK 1.8

Deploy:

*deployment server (ex: Tomcat, Glassfish)

*JRE 1.8

.NET:

* VisualStudio 2015 Community

* Clean, Build

* .NET Framework 4.0

Deploy:

*deployment server (Visual Studio embedded IIS, IIS Express)

6. Bibliography

1. http://www.java2blog.com/2013/03/soap-web-service-example-in-java-using.html

2. http://www.javatpoint.com/hibernate-tutorial

3. http://stackoverflow.com/questions/15940234/how-to-do-a-soap-web-service-call-from-java-class

4. http://www.coned.utcluj.ro/~salomie/DS_Lic/LabAndProject/DS_Lab_Resources.pdf

5. https://www.youtube.com/watch?v=D720X8pnXGs

6. http://www.c-sharpcorner.com/article/integrating-net-web-services-with-java-and-classic-asp-clie/