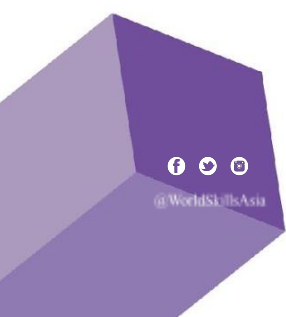


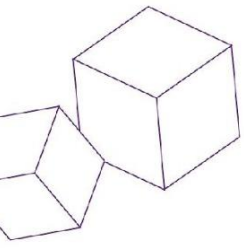
WorldSkills Asia Online Friendly Skills Games 2021

IT Software solutions for business

TEST PROJECT

Session 1





Subject area description

Our client is the owner of a logistics centers network, parking lots for trucks and a service for the delivery of small goods to end users. To optimize and automate all the company's activities, it was decided to develop software that would solve certain tasks.

The client outlined the main principles of activity and their interaction for a better understanding the entire company activities:

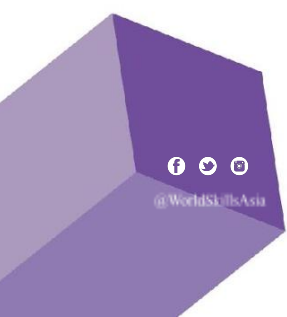
“Our company is divided into three large areas that are related to each other. Based on this, it is very important not to duplicate the same data in different subsystems and use a single data source.

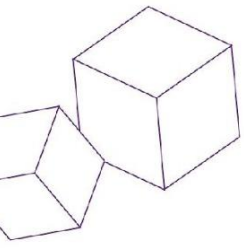
One of the main directions is logistics. We are engaged in planning and organizing cargo transportation within the country, as well as between the nearest countries. For that we have to store a large amount of information about trucks (manufacturer, model, and number, year of manufacture, mileage, type, and assigned drivers background). Since we transport various goods, we need to pay attention to trailers types that may differ not only in load capacity and dimensions, but also in the permitted type of cargo (for example, trailers for fuel, radioactive waste, livestock, etc.). Moreover, not every type of trailer can be attached to a truck (it depends on the type of truck). Since some goods may be too dangerous to transport, a certain level of danger may be assigned to the trailer. Not all drivers can be allowed to transport dangerous goods - only drivers with the appropriate level can get access to such trailers. The level of admission depends on the age and the number of trips that the driver has made.

In order to deliver goods to neighboring countries, we are preparing a special list of documents for a specific trip. We add all the information about the cargo, the driver and the truck to the package of documents.

Our company closely monitors the health of employees, preparation of schedule, taking into account medical indications. On the way, drivers must strictly follow planned time intervals (of working shifts, sleep, rest, food, etc).

Since drivers need to make regular stops during long routes, over time we have developed a network of our own parking centers, where drivers can reserve places for taking a rest, pass a cargo check, medical examination, maintenance and a number of other services. All these services are provided to their employees on preferential terms, so parking centers are often crowded, which can cause difficulties for managers.

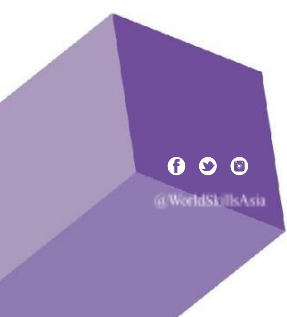


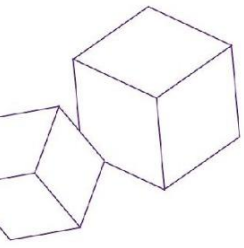


As an additional direction, our company is engaged in the delivery of small goods to end consumers. We have our own small truck fleet that is used every day to deliver small goods. Our interest is to load the car with cargo as fully and optimally as possible, so that the maximum amount of cargo can be delivered to consumers in one trip. In order not to get confused in all the logistics, we also store information about each cargo (its article number, dimensions, weight, info whether it is fragile or not, the possibility of putting another cargo on it, the recipient, the desired delivery date) and use this information when loading up the car. Creating a route for a car, we build a map to cover the shortest possible distance and deliver the cargo to the maximum number of consumers. We are going to develop a special mobile application for the driver, in which they could see information about all parcels on the current route, their articles, a list of customers and customer contact details. To speed up the work and search for goods and customer information, we are going to use a system with QR-codes so that the driver can point the camera at the code and get all the information about this parcel.

We expand our service area and organize our small deliveries to an increasing number of cities. In this regard, we are building a logistics chain for small goods in our country and we need to record all the cargo movements (who were the sender, in which center the parcel was received, through which centers it passed and in which car it was). Also, our security service requires information about the drivers who transported this parcel between logistics centers.

We also noticed that some customers order goods very often and for several items at once, which is important to take into account forming delivery and prioritizing transportation”.





Designing

Use Case

To understand the general context of the entire system functionality, you need to design a use case diagram, and further use it offering options for automating certain business processes to the customer.

Save the diagram file in the .vsdx and .pdf formats.

Entity Relationship Diagram and Data Dictionary

Due to the customer's requirement to use a single data source, you need to develop an ER-diagram for the entire subject area, as well as a data dictionary with important comments for developers of the system.

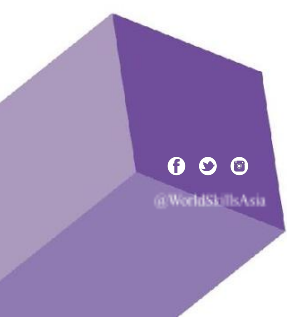
For the ER diagram the 3rd normal form with link integrity is mandatory. When creating your diagram, ensure you have a consistent and meaningful naming scheme, create the necessary primary and external keys, and take note of links (identification, multiplicity). The data types of the fields are not so important in the ER diagram, so it will be enough to display the entities with the required fields.

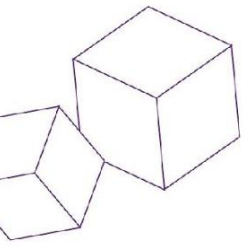
Keep in mind that stage is just designing (without physical realization in DBMS), so you don't have DB access at this session and need to use standard installed designing tools.

Save the ER diagram file in the .vsdx and .pdf formats.

For the ER diagram, you need to create a data dictionary with a set of information describing what type of data is stored in the database, as well as the format, structure and ways of using the data. Pay attention to the correspondence between your diagram and data dictionary. Use appropriate data types, constraints and formats. Describe the constraints defined in the tables, including the primary keys, external key relationships with other tables, and non-null constraints. Do not forget to provide any necessary clarification and comments for ambiguous fields. Please use the DataDictionary_Template.xlsx file as the dictionary template.

Save the data dictionary file in the .xlsx format.





Mobile layouts and API Documentation

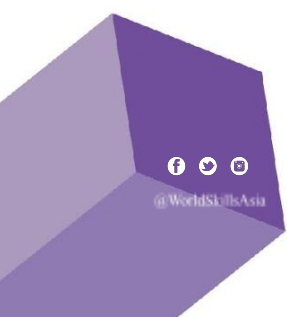
Since the process of delivering small goods to end users is developing at an active pace, the issue of early development of this module is quite acute.

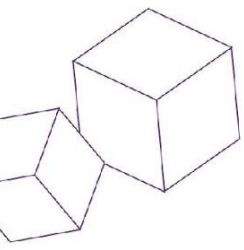
Before starting development, it is necessary to perform a number of design stages. At the initial stage, it is necessary to develop a mobile application schematic interface for a truck driver and API documentation to ensure the connection of mobile devices with a central data server.

As the initial data, use the description of the company's activities from our client. It is important to make the interface convenient and understandable for the common user. If necessary, use the notes / comments directly on the layout.

Developing API documentation, make sure that it includes the following information:

- The path to call each endpoint (endpoint),
- HTTP methods that are used for each endpoint,
- Description and purpose of each method,
- Query data and where it should go (path, query-string, or body),
- Specifying whether the request fields are mandatory or optional.





Library

To reuse the functionality described below in different projects, you need to create an external library that can be saved as a file .dll (.NET) or .jar (Java) or .cf or .epf (1C) or folder with .py file (Python). Two methods should be implemented in the library:

- 1) to calculate the distance between two coordinate points,
- 2) to get free parking spaces in a given period of time on a certain day.

It is necessary to follow the rules for naming libraries, classes and methods in them. In case of errors in the naming framework, your work cannot be verified and your result will not be credited. Classes and methods must contain the public modifier (if it is implemented within the framework of the platform) so that external applications can access them. The method specifications will be presented in a separate file.

1. GetDistance()

Input:

- the latitude of the first point,
- the longitude of the first point,
- the latitude of the second point,
- the longitude of the second point.

Output:

- the number of meters rounded according to mathematical rules.

To calculate the distance, use the following formula:

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

, where

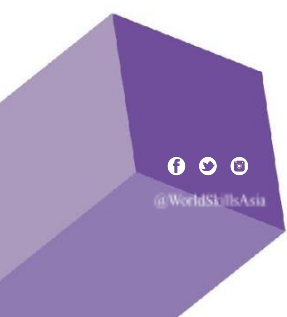
- r is the radius of the sphere (for the earth, take the average rounded value of 6370 km);
- φ_1 and φ_2 — the latitude of the first and second points in radians;
- λ_1 and λ_2 — the longitude of the first and second points in radians.

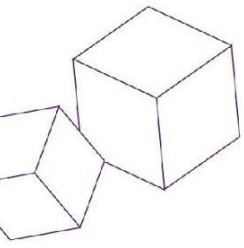
To convert degrees to radians, you can use the following formula:

$$\text{Radians} = \frac{\pi}{180} \times \text{Degrees}$$

For example, for the points (55.7539° N, 37.6208° E) and (59.9398° N, 30.3146° E), the distance according to the above formulas will be 634469 m or 634.47 km.

You need to implement the proposed algorithm as a program code. Using the built-in functionality in the platform to calculate the distance between two coordinates is prohibited.





2. AvailableParkingSpaces()

Input:

- the size of the Parking lot (number of columns and rows);
- list of busy periods (three arrays: parkingSpaces - room spaces, startTimes - the beginning of the Parking lot, EndTimes - completion of Parking);
- the time as new Parking (start time and duration in total minutes).

Output:

- list of parking space numbers that will be available in the specified time period. The result list should be ordered by parking space number (at first by column letter and after that by row number).

For example, there is a parking lot with 3 rows and 3 columns - that is, 9 parking spaces.

Busy time intervals:

B1 10:00-14:00

C3 15:30-20:00

A2 06:00-12:20

B3 09:00-10:00

C3 08:00-11:30

A1 04:00-16:45

B1 14:00-18:00

C2 10:00-22:00

A3 05:00-17:00

The desired time of the new parking:

11:00 - 330 minutes

Thus, the following parking numbers should be available at the exit during the specified time period:

B2, B3, C1.

