Software Applications Development
WorldSkills 2026 National Competition
HUNGARY

Round 3 - Session 3

Submitted by:

Skills IT

Contents

1. Introduction

In this session you have been provided multiple services to integrate with, on top of that you must develop exact APIs based on the provided swagger and this documentation.

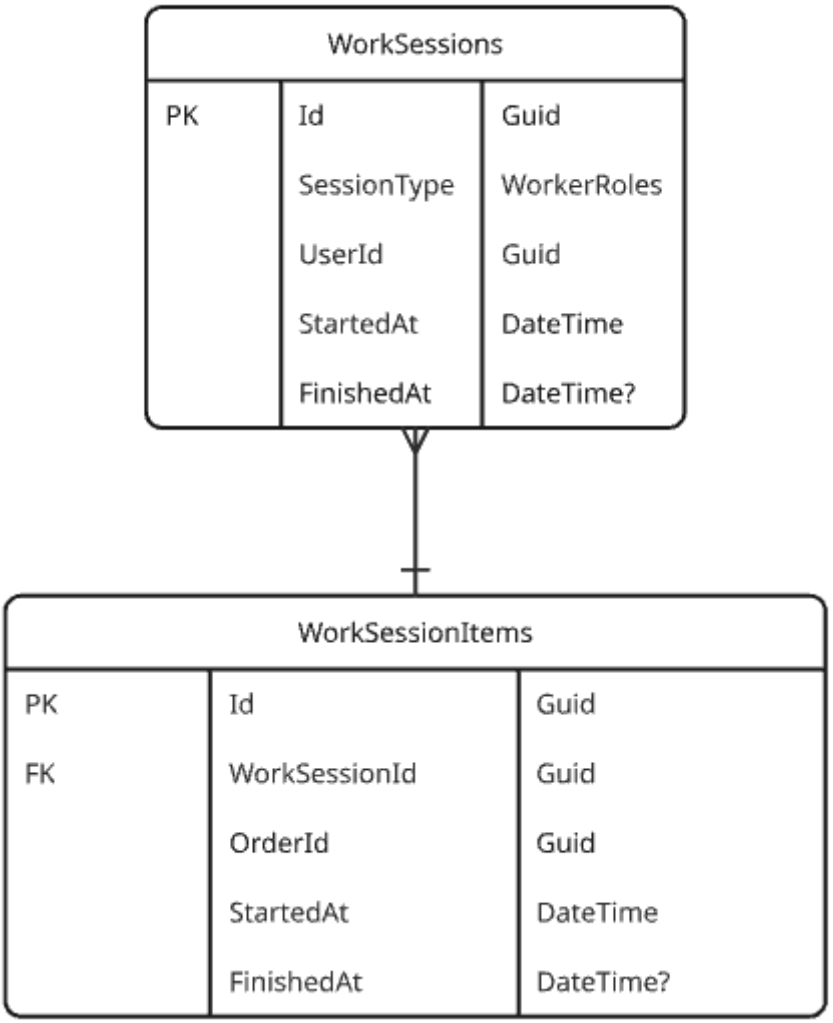
1.1 Description of project and tasks

Your task will be to implement an application that will provide warehouse workers with work sessions and work items to work on.

1.2 Database

You have been provided with a MySQL database, you must use this database. You can find connection information on the info page.

The database has the following structure. You must **not** change the schema and use this as-is.



1.3 How to submit your work

1. You have to submit all your in the git repository you have been provided with.
2. Your **README** file has to contain instructions on how to start your backend and frontend. We prefer you provide executable files or a built/deployed solution.

2. Part 1 – Work sessions

In this section, you need to create a backend in your chosen technology to help warehouse workers manage their work. When an employee starts their work they will start a work session. This will follow all work items they have. An employee must have at most 1 work session active. When the worker asks for a new work item to do, they will be provided with one order to work on, this will be a **work session item**. A work session item is associated with an order in the ERP system. The ERP system has a status API, which you can use to lock the order and follow the work it has had. When a work session is stopped the work session item also must be stopped and the order released, the order statuses are the following in order: **Created, Started, Gathering, Gathered, Delivering, Delivered**. In case a **Gatherer** worker requests a new work session item, all **Created** and **Started** orders must be considered (more logic later), when a **Gatherer** work session item is stopped the new status must be set to **Started**, when a work item session is finished then the **Gathered** status must be used. **Transporter** roles work the same way only In case a work session is requested all **Gathered** orders must be considered and set to **Delivering**, when a session item like this is stopped **Gathered** status must be set and when it's finished the **Delivered** status must be used.

2.1 The user information

The user information is handled the same way as in session 1, refer to that session in case of questions.

2.2 Start work session

Starts a new work session for the logged-in worker. If there is an already running session for the worker, **stop it** and start a new session. Make sure to also stop the session item, if there is one and release the order.

API Schema

POST api/work-sessions/start

Response body**:**

```
Unset
{
  "id": "123e4567-e89b-12d3-a456-426614174000",
  "startedAt": "2025-04-26T11:39:40.293Z",
  "sessionType": "Gatherer"
}
```

Session type is the same as the worker role: **Gatherer, Transporter.**

2.3 Get new work session item

The worker will periodically request new work items to work on. The following business logic must be used for selecting the next order to work on:
All roles must prioritize orders by descending priority, meaning that higher priority orders should be first worked on. In case of matching priority the order date must be checked. The oldest order should come first.
Gatherer: Gatherers can work on **Created** and **Started** orders. A **Started** order should always be prioritized over **Created** orders. In case there are multiple Started orders the same generic prioritisation logic should be applied.
Transporter: Transporters can work on **Gathered** orders. The generic prioritisation logic should be applied. The open orders can be loaded from the **erp-service:**
GET erp-service/api/orders/open
Response body**:**

```
Unset
[
  {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "partnerId": "123e4567-e89b-12d3-a456-426614174000",
    "orderedByUserId": "123e4567-e89b-12d3-a456-426614174000",
    "orderedAt": "2025-04-26T16:03:20.049Z",
    "status": "Created",
    "workerUserId": "123e4567-e89b-12d3-a456-426614174000",
    "priority": 1,
    "items": [
      {
        "id": "123e4567-e89b-12d3-a456-426614174000",
        "productId": "123e4567-e89b-12d3-a456-426614174000",
        "quantity": 1,
        "price": 1
      }
    ]
  }
]
```

After finding the proper order it must be locked by setting the orders status in the **erp-service**. For a **Gatherer** session type the status must be set to **Gathering**, for **Transporter** session type the **Delivering** status must be used.
PATCH erp-service/api/orders/{id}/status
Request body:

```
Unset
{
  "status": "Created"
}
```

API Schema

GET api/work-sessions/{id}/items/next

Response body:

```
Unset
{
  "startedAt": "2025-04-26T16:05:34.021Z",
  "orderId": "123e4567-e89b-12d3-a456-426614174000",
  "partnerName": "…",
  "orderItems": [
    {
      "productName": "…",
      "quantity": 1
    }
  ]
}
```

In case there are no orders to work on for the given role, a **204 No content** should be returned.

In case the worker already has a working session item, the same worker session item should be returned.

If there is no active work session for the logged in user a **404 Not found** should be returned with the following content:

```
Unset
"No active work session found"
```

In case there is an active session but the id is mismatching with the provided one a **422 Unprocessable entity** should be returned with the following content:

```
Unset
The work session ID does not match the current work session
```

2.4 Mark work session item as done

When a work session is done by the worker it must be marked as done in the system. If the work session type is **Gatherer** the ERP order's status must be set to **Gathered**, if the session type is **Transporter** the ERP order's status must be set to **Delivered**.

API Schema

POST api/work-sessions/{id}/items/done

If there is no active work session for the logged in user a **404 Not found** should be returned with the following content:

```
Unset
"No active work session found"
```

In case there is an active session but the id is mismatching with the provided one a **422 Unprocessable entity** should be returned with the following content:

```
Unset
The work session ID does not match the current work session
```

2.5 Stop work session

When a worker is done with their work they must stop their work session. In case a work must stop their work while they have a work session item assigned to them, the work session item also must be stopped.

API Schema

POST api/work-sessions/{id}/stop

If there is no active work session for the logged in user a **404 Not found** should be returned with the following content:

```
Unset
"No active work session found"
```

In case there is an active session but the id is mismatching with the provided one a **422 Unprocessable entity** should be returned with the following content:

```
Unset
The work session ID does not match the current work session
```

6. Additional information

- Some media, icons and text have been provided for you in the media files. You are free to use these, but you can also create your own, as long as the application is still fit for purpose. **You should not use any other media files (e.g. downloaded videos, images, icons, etc.)**.
- Clean code and user interface accessibility are also important considerations.
- Do not hardcode API responses as another database will be used for testing.