

2024-2025

Programming 6



Krystal Distribution Group



Abstract	2
Our Customers	3
Supply Chain	4
Pricing and Storage strategy	6
Bounded Contexts	8
Some concepts in detail	8
User Stories (import these in gitlab)	9
Technologies and techniques	11
Project setup	12

Abstract

Krystal Distribution Group (KdG): Efficient and Seamless Mineral Logistics

Krystal Distribution Group (KdG) specializes in the efficient and seamless distribution of essential raw materials, including gypsum, iron ore, cement, petcoke, and slag. They want to renew their logistics system, as it is outdated and hard to change. They need a new system that can be adapted easily and makes data available for other systems.

The new logistics system was baptized as KdG MineralFlow, and it aims to optimize the flow of materials from arrival scheduling to final warehousing.

Currently, KdG handles the following materials, but their state-of-the-art warehouses are capable of adapting easily to all sorts of raw materials, so the system needs to be capable of handling these kinds of evolutions.

Descriptions of Raw Materials

1. Gypsum:

- **Description:** Gypsum is a soft sulfate mineral composed of calcium sulfate dihydrate. It is commonly used in the construction industry for producing plaster, plasterboard, and cement. Gypsum is also used in agriculture as a soil conditioner and fertilizer.
- **Storage Price per Ton per Day:** 1\$ / t / d
- **Price per Ton:** 13\$ / t

2. Iron Ore:

- **Description:** Iron ore is a naturally occurring mineral from which iron is extracted. It is a crucial raw material in the production of steel, which is used extensively in construction, manufacturing, and transportation industries. Common types of iron ore include hematite and magnetite.
- **Storage Price per Ton per Day:** 5\$ / t / d

- **Price per Ton:** 110\$ / t
- 3. **Cement:**
 - **Description:** Cement is a binder substance used in construction that sets, hardens, and adheres to other materials to bind them together. It is a key ingredient in concrete, mortar, and stucco. Portland cement, made from limestone and clay, is the most common type.
 - **Storage Price per Ton per Day:** 3\$ / t / d
 - **Price per Ton:** 95\$ / t
- 4. **Petcoke (Petroleum Coke):**
 - **Description:** Petcoke is a carbon-rich solid material derived from oil refining. It is used as a fuel in power generation, cement kilns, and other industrial processes due to its high calorific value. Petcoke is also used in the production of electrodes for the aluminium and steel industries.
 - **Storage Price per Ton per Day:** 10\$ / t / d
 - **Price per Ton:** 210\$ / t
- 5. **Slag:**
 - **Description:** Slag is a byproduct of the smelting process used to produce metals from their ores. It is used in construction as an aggregate in concrete, road construction, and as a raw material in cement production. Slag helps improve the durability and strength of concrete.
 - **Storage Price per Ton per Day:** 7\$ / t / d
 - **Price per Ton:** 160\$ / t

Our Customers

Description: Sellers are businesses or entities that deliver raw materials (gypsum, iron ore, cement, petcoke, slag) to Krystal Distribution Group (KdG) for storage. They pay for the storage services provided by KdG and rely on the company to manage the logistics and warehousing of their products.

Key Features:

- **Delivery management:** KdG can handle 40 trucks per hour in the planning schedule. This means that if 40 trucks (and let's say each truck is a 25 tonner) arrive at the same hour, within the hour, there is 1 kiloton of raw material added to the warehouses.
- **Payment for Storage:** Suppliers pay KdG a storage fee based on the volume and duration of storage.
- **Inventory Management:** KdG manages the inventory and ensures the safe and secure storage of raw materials.
- **Storage Space Availability:** Sellers can only make appointments if there is available storage space (less than 80% full). If, throughout the day, the warehouse reaches capacity, each warehouse has overflow capacity (for example 110%).

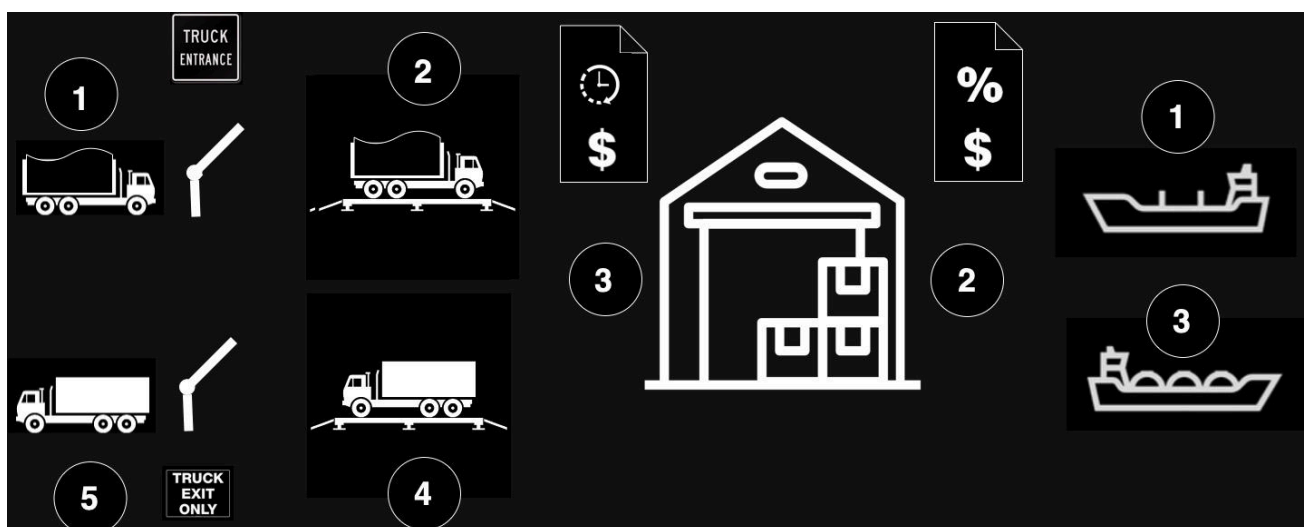
Description: Buyers are businesses or entities that purchase raw materials stored by KdG. The Buyers make arrangements with the Sellers directly, when a purchase is made, the purchase order system is seamlessly integrated with KdG, so that the ship that arrives to pick up the materials can get the amount purchased. The buyer has the shipment order (SO) with him, which should match the purchase order (PO). For each purchase that ends up in a shipment, KdG gets a commission fee of 1% as decided per year contract.

Key Features:

- **Commission-Based Pricing:** For each successful shipment, calculating a commission fee for each PO.
- **Purchase order, shipment order matching:** Shipments get prepared by KdG, loaded and the necessary paperwork, PO en SO consolidation is done by KdG.
- **Shipment handling:** Loading a ship can take up to a couple of days depending on the mandatory tasks such as refueling, safety inspection and of course loading of the materials.

Krystal Distribution Group efficiently manages the flow of raw materials by catering to these two types of customers, ensuring a balanced and profitable operation for suppliers and buyers.

Supply Chain



Trucks (Land side)

1. A **customer (selling company)** makes an **appointment** for a **truck** loaded with raw material. The type of raw material is known up front, the license plate of the truck as well. According to schedule, the truck arrives at the **gate**. The gate opens when the license plate is recognized and within the arrival window of the existing appointment. Appointments can be made 24/7.
The window of arrival is exactly an hour. If a truck is planned at 10 am, he can arrive from 10h00 to 11h00. So, if the truck arrives at 9h59, it must wait a minute to enter. If he arrives at 10h59 he is still on time. KdG can handle 40 trucks per hour.
2. The truck passes a **weighing bridge**, the weight of the truck is registered and the truck gets assigned a **warehouse** number. Each customer has a warehouse that it uses and per warehouse only 1 type of raw material can be stored. When a warehouse is empty any raw material can start accumulating in the warehouse, but as soon as a type of material is inside the building, the building can only contain that type of material. The amount of material is always calculated in tons. The most popular truck is a 25-tonner and as such can hold a payload up to 25t, but smaller trucks containing 250 kg (0,25T) can pass the weighing bridge and are all welcomed by KdG.
3. The truck drives to the allocated warehouse and dumps its load on a conveyor belt, the **conveyor belt** makes sure that all materials are stored securely in the warehouse. As soon as the truck payload hits the conveyor belt, a pressure sensor will start the conveyor belt, and a **payload delivery ticket (PDT)** will be generated. The PDT registers the type of material and the time of delivery. This is needed for invoicing as KdG can charge per ton per day in the warehouse. (See the pricing and storage strategy for more details on this topic)
4. The truck passes the weighing bridge again, where we can finally deduct the weight of the payload delivered, the truck driver will get a **(Weigh Bridge Transaction) WBT** document.
5. After the truck passes the weighing bridge, we enrich the payload delivery ticket with the weight from the WBT document.
6. The truck passes the exit, where its license plate is scanned again and we consolidate the appointment made by the seller.

we need an
info about the
warehouse
number from
the warehouse
context



Ships (Water side)

1. A ship can arrive anytime and dock at the loading quay at any time. For the sake of simplicity, we assume that we have unlimited space for ships to arrive. At arrival the captain will input his Shipping Order (SO) and his vessel number. For this vessel an inspection operation (IO) will be planned and a bunkering operation (BO) will be issued (refueling) of the ship.
The shipping order contains the purchase order, the vessel number and the customer enterprise number buying the goods.
2. The shipping order is matched with the purchase order and a loading slip (out of scope) is issued. For the sake of simplicity we only implement the happy path meaning that for each shipping order there will always be a purchase order sent upfront by the selling company.

See the example of how the PO looks like. Please remark that some ships can carry multiple types of materials at once. As per the selection of the raw materials to be loaded, we always pick the warehouse with the “oldest” material. As it is KdG’s customer service to charge the selling company as little as possible. For simplicity’s sake, one SO matches exactly with one PO.

3. As soon as mandatory tasks are done (BO ended) and (IO successfully ended), the ship is loaded (loading slip issued) the ship will leave. The amount of raw materials is deducted from the respective warehouses. As soon as the ship leaves, KdG can charge an additional commission fee on the total purchase order that was fulfilled.

Pricing and Storage strategy

Storage

- KdG has multiple customers, and each customer has 5 warehouses (one per type of material), if in the future we offer to handle more types of materials, we will configure our warehouses to do this and just assign an additional warehouse to each customer.
- If the warehouse is full (> 80%) we will not accept any new appointments anymore. Deliveries that were already planned via an appointment can be delivered without a problem as we have overflow capacity techniques. So in our system, it is acceptable that a warehouse is for example 107% full.
- When a shipping order (SO) can be fulfilled, we will first load the oldest material stock (‘oldest’ is determined by the PDT, we take abstraction of internal management of warehouses in segments), then the second oldest stock, then the third and so on until the complete SO is fulfilled.
- A ship can leave when the IO, BO and SO are all fulfilled successfully.
- A warehouse can carry 500 kt of raw material.
- A shipping order can take up to 150 kt of raw material, we assume that each ship can carry its own shipping order.

we need to know
the capacity of the
respected
warehouse from
warehouse
context

Pricing:

- KdG can send an invoice to its customers at any time. An invoice is calculated based on storage fees and commission on the sold goods.
- Calculating commission is straightforward. We just take the total amount of the PO times our commission percentage.
For instance: if we have a PO of 100 kT of iron ore, with 50 kT of gypsum. We just calculate the price per ton per material and take 1% (our commission) of the total sum.
Iron core is currently 110\$ per ton and gypsum at 13\$ per ton. So a total of $(110 * 100.000 + 13 * 50.000) * 0,01 = 116.500\$$ commission for this shipment.
- Storage is easy as well. Per ton there is a flat fee per 24 hours depending on the type of material per ton. Gypsum is the cheapest and costs 1\$ per day storage ton.
For instance: Let’s say that we have 100 ton that has arrived in 4 truck loads:

Arrival	Tons	Storage cost on	Number of days	Cost in dollar
27/07/2024 09:00	25	1\$	3	75
27/07/2024 21:00	25	1\$	2	50
28/07/2024 09:00	25	1\$	2	50
29/07/2024 09:00	25	1\$	1	25
Calculated on		30/07/2024 09:01		200

- Let's say we have a SO that needs 30 tons of this material. We will first load 25 tons of the oldest material (3 days) and an additional 5 tons of the payload delivered on 27/07 21:00. If no material gets added the next days we will have the following calculation:

Arrival	Tons	Storage cost on	Number of days	Cost in dollar
27/07/2024 09:00	0	1\$	4	0
27/07/2024 21:00	20	1\$	3	60
28/07/2024 09:00	25	1\$	3	75
29/07/2024 09:00	25	1\$	2	50
Calculated on		31/07/2024 09:01		185

- Storage fees get calculated each day at 9:00 am.

Bounded Contexts

Warehousing Context

- Manage warehouse information, including storage capacity and raw material stock levels.
- Maintain records of mineral types and quantities stored.

Landside Context

- Schedule truck arrivals to ensure efficient processing and avoid congestion.
- Maintain a log of scheduled and actual arrival times. **activity window? record**
- Maintain a log of departures.
- Record the weight of trucks arriving and leaving the warehouse.
- Calculate the net weight of minerals delivered.

Waterside Context

- Track outgoing shipments.
- Track Inspection Operations, Bunkering Operation
- Match Shipping Orders with Purchase Orders

Invoicing Context

- Calculate storage fee each day per customer
- Calculate commission fee each day per customer
- At request invoicing to customers (invoice outstanding credit)

Some concepts in detail

- **Weighing Bridge:** Full automatic system, that scans the license plate, and registers the weight of the truck, point in time of the weighing and the license plate.
- **Shipment Order:** A shipment order contains a purchase order reference, a vessel number and estimated arrival date and departure date. When the ships arrive and depart the actual dates are filled in. When all dates are filled in the SO is considered fulfilled. Of course, a ship can only leave when it is loaded, inspected and refueled. The SO contains a signature and date of inspection by the safety inspector.
- **Purchase Order:** A purchase order contains a date, purchase order number, customer number, name and a number of purchase order lines. For each type of material, we get the type of material, the amount in tons and the agreed pricing per ton of raw material.
- **Inspection Operation:** This entire operation was performed by the safety inspector. To limit the scope, we reduce the operation to a date on the SO with a signature. Yet we want to see the concept of an IO in the code.
- **Bunkering Operation:** This entire operation is performed at the port and to limit scope it is just a date on the SO to be completed. Yet the BO exists as a concept in the code.
- **Warehouse:** KdG's warehouses are located on different sites around the port. Each warehouse has a unique number. So each payload can be delivered at the following unique place: W01 - Warehouse 01. We allocated 1 warehouse per customer, per material type. A warehouse can contain up to 500 kt (=100%) but as we have overflow techniques the warehouse can be overloaded. In this way, it can still accept loads that have already been planned. This shows in the system that the warehouse is for instance at 115% capacity.
- **Truck:** The concept should be made as simple as possible, the truck should at least contain a license plate.
- **Material, Mineral, Raw Material:** Per ton
- **Units of measure:** Raw material is measured in tons, money in dollars, we use 24-hour time notation and UTC timezone for all dates. Dates are always presented in ISO format.
- **Delivery appointment:** There can be up to 40 appointments each hour. We work 24/7 all year long. A whole-hour slot is booked.

- **PDT**: Official payload delivery ticket: type of material, date of delivery, warehouse number, dock number.
- **WBT**: Official weighbridge ticket: weights, times of weighing and truck identification

User Stories (import these in Gitlab)

(Most stories are only implemented in the backend and demonstrated using http files in IntelliJ, only some of them will be enriched with a UI (*Includes UI*), more info below)

- LANDSIDE**
- As a seller, I want to make an appointment for a truck loaded with a specific raw material so that the truck can enter the facility during a specified arrival window. *Includes UI*
 - As a **truck driver**, I want to be recognized by my license plate so that the gate opens for the truck to enter the facility.
 - As a **truck driver**, I want to receive the weighing bridge number when accessing the site so I know where to go.
 - As a **truck driver**, I want to receive a warehouse number when I pass the weighing bridge I was assigned to, so I can dump my payload on the assigned conveyor belt.
 - As a **truck driver**, I want to dock to the correct conveyor belt and receive my copy of the **PDT** and new weighing bridge number.
 - As a **truck driver**, I want to pass the weighing bridge and get a Weighbridge Ticket (**WBT**) that includes: The gross weight upon arrival, tare weight, net weight, timestamp of weighing, truck license plate number.
 - As a **warehouse manager**, I want to check if trucks arrive within the scheduled arrival windows. *Includes UI*
 - As a **warehouse manager**, I want to know how many trucks are on site so that in case of emergency I know if there is anyone on site or not. *Includes UI*
 - As a **warehouse manager**, I want to have an overview and know what the total raw material is in my warehouse as well as the details of each warehouse. *Includes UI*
 - As a buyer, I want to send a PO to KDG to make sure they can expect a shipment soon.
 - As a **warehouse manager**, I want to know what PO's are fulfilled and which ones are still outstanding. *Includes UI*
 - As a ship captain, I want to input the Shipping Order and vessel information upon arrival at the loading quay, so that the necessary operations, inspection, bunkering and loading can be planned and initiated.
 - As an inspector, I want to see all outstanding IO's and complete inspections so that vessels can be inspected
 - As a bunkering officer, I want to plan all BO's with a maximum of 6 a day so that vessels can be refueled.
 - As a foreman, I want SO's to get matched with PO's so that we know the vessel is at the right port and is an accepted buyer's vessel.
 - As a foreman, I want to see all shipment arrivals containing relevant information such as vessel number, inspection status and so on.

- As a **warehouse manager**, I want the oldest stock of raw materials for loading allocated automatically so that the seller is charged as little as possible for storage.
- **As a ship captain I can enter my shipment order (with PO reference).**
- As a **warehouse manager**, I can set the IO and BO as completed.
- **As a ship captain, I want an overview of all operations (IO, BO) to know if I can leave the port.**
- As a **warehouse manager**, I automatically want each warehouse volume adjusted accordingly and inventory levels accurately maintained.
- **As an accountant**, I want the system to automatically calculate the commission fee on each fulfilled PO as soon as the ship leaves.
- **As an accountant**, I want the storage costs to be calculated each day at 9 am precisely.

Stories marked with *Includes UI* not only involve backend implementation but also extend to the development of a **React**-based frontend client designed primarily for the *warehouse manager's* use. The core feature is a comprehensive **dashboard** providing insights into real-time site operations. From this dashboard, users must be able to **drill down** into detailed views, including lists and item specifics.

for the exam, you will need some type of filtering, you can do client side filtering, or server side.

The following information is presented:

- **The total number of trucks on-site.** Optional: categorised by status (e.g. "bridge," "belt," "docked," "bunkering").
- **Lists of trucks for the current date/time** with drill down to **details like arrival window, seller, license plate, mineral type, WBT,...**
Trucks that are delayed are visually highlighted.
Another date/time can be picked.
- Overview of **warehouses with seller, type/amount of material** and drill-down to (incoming) payload records. Provide a floorplan view (each warehouse as a rectangle, this may be done using AP-positioned <div/>'s in px units). **Use visual elements like colour to denote fullness/overflow. A seller can be selected to highlight its warehouses.**
- **Amount and list of PO's (all, filled, outstanding) by seller.**

Besides this, a seller can

- **Make an appointment for a truck by supplying the necessary information. If no appointment is possible at the requested time, an error message is presented.** Optional: pick an available timeslot to book an appointment.

The frontend assignment is open for (creative) interpretation, which can be discussed with a teacher when in doubt.

Technologies and techniques

All technologies and concepts handled in this course.

Backend:

- Concepts of Domain Driven Design with a rich domain, entities, aggregates, value objects, domain events
- A clear context mapping technique between bounded contexts
- Hexagonal Architecture with clear separation, boundaries, infrastructure agnostic uses cases
- Commands and events
- At least 1 event sourced aggregate.
- At least 1 projection of data (derived state)
- At least 1 snapshotted event sourced aggregate
- Apply resilience patterns where needed
- Apply security to REST endpoints
- Write at least 1 architecture test, 1 integration test, 1 unit test using mocking, 1 unit test using stubbing, 1 test using testcontainers

Frontend:

- Use React with Functional Components (hooks) and Typescript
- Use an asynchronous state library like React Query
- Use routing for separate pages (like a list of items)
- Use a component framework (like Material UI, etc.)
- Organize your code using a clean folder structure, components/pages, hooks and (API) services
- Secure the application with OAuth. Authentication suffices. Optional: authorisation with roles (warehouse manager, seller)
- Keep the dashboard up-to-date using polling (optional: web sockets).
- Apply UX tips given during class and test the UX (minimal 3 key task tests on one person, sensible feedback applied in the UI, feedback report provided).

Approach and planning

During this course, you'll work out this project. Your progress will be monitored, as we are building an architecture that is designed for evolution, change requests will be asked along the way. A low-stake evaluation moment will be planned in the 2nd half of the teaching period.

Your Git repos will be used as a single source of truth and should contain:

- A clean commit history (very important!)
- User stories and clean progress
- Wireframes if applicable
- Lightweight, to the point, documentation
 - Domain models
 - Command and event catalog
 - Make sure it is easy to read and understand

Project setup

A gitlab group will be created for you with two empty repos (frontend, backend). You are required to work in this group.

Backend repo

There will be a **documentation** folder containing your documentation (event and command catalog) A docker-compose file will be provided as well in the folder **infrastructure**.

The docker file consists of the following services:

- A database for your application (MySQL):

This is a MySQL database. Use this database for your application. The MySQL database port will be mapped to 3308, check the service `mysql_kdg_db` for more information on how to connect with what credentials. Volumes are mapped under the infrastructure folder, make sure to not commit them by setting up a gitignore file.

This service connects via a custom docker network called `backend`.

- A message oriented middleware (RabbitMQ):

This is RabbitMQ, the management port is mapped to port 15672, check the service `service_kdg_rabbitmq`.

Volumes are mapped under the infrastructure folder, make sure to ignore them in your gitignore file. This service connects via a custom docker network called `backend`.

- An identity provider (Keycloak):

This is a keycloak identity provider. Detailed documentation can be found here:

<https://www.keycloak.org>

This service is called `idp_keycloak` and it will be clear during lectures how we are going to integrate with this service. A special docker network is set up called `kc`.

The management console is exposed at <http://localhost:8180/auth/> - a nice starting point for documentation as well.

- A database for all things related to Keycloak:

This is a mysql database as well, but cannot be used for direct interactions. It links to another docker network especially for keycloak called `kc` network.

This service is called `idp_mysql`.

Frontend repo

- Create a new React project using 'vite' (see instructions during class)
- Init git on the React project, make the initial commit, set the remote and push (*see instructions on the empty gitlab frontend repo page*)
- Start working: commit and push frequently (at least after each work session)