In this project, we will extend the application we developed in Homework #3 in order to implement the Plant Hire process at RentIT and the corresponding order-to-cash process at BuildIT. Below is a description of these processes. Do not hesitate to ask questions to the lecturer or lab assistants about ambiguities in this description, or about design choices you make.

**Context: Plant Hire**

*Buildit* is a construction company specialized in public works (roads, bridges, pipelines, tunnels, railroads, etc.). Within Buildit, it often occurs that engineers working at a construction site (called “site engineers”) need a special type of equipment, such as a truck, an excavator, a bulldozer, a water pump, etc. A piece of heavy equipment is called a “plant” in the construction jargon.

Buildit owns very little equipment and instead it hires most of its equipment from specialized heavy equipment suppliers. One of Buildit’s preferred plant hire supplier is Rentit.

![Plant for hire (image taken from Holden Plant Hire Ltd., UK)](image)

**Buildit’s Plant Hire Process**

The business process for hiring a plant (seen from Buildit’s perspective) is described below.

When a *site engineer* needs to hire a plant, he/she consults the catalogue of a plant supplier in order to identify a plant that fulfils the requirements. Once the site engineer has identified the required plant, he/she checks the plant’s availability during the required period of time as well as the price. If the plant is available, the site engineer creates a *Plant Hire Request*, which includes the identifier of the construction site where the plant is needed, the supplier of the plant, the plant’s identifier, the expected start and end date of the hire period and the cost of hiring the plant for this period of time. This cost is calculated based on the plant’s price per day of the selected plant and the number of days the plant is hired. Naturally, the plant hire request also includes the name of the site engineer making the request.

Every plant hire request has to be approved by a *works engineer* at BuildIT. The purpose of this approval is to avoid excessive or unnecessary plant hiring and, more generally, to ensure that plant
hiring costs are minimized and that they are within budget. In some rare cases, the works engineer rejects the plant hire request or makes modifications to the plant hire request before approving it. If the works engineer rejects or changes the plant hire request, they normally write a short explanation in a “comments” field in the plant hire request. In such cases, it is usually the case that the works engineer talks with the site engineer prior to rejecting or modifying their request in order to avoid misunderstandings.

Once the works engineer has approved the plant hire request, Buildit’s information system automatically generates a Purchase Order (PO) for hiring the plant and sends this PO to the plant supplier. The supplier may accept or reject the PO. One reason why a PO might be rejected is that the plant that is being requested is no longer available during the requested period of engagement (e.g. it has been hired by someone else).

When the plant is hired, the supplier delivers it to the construction site at the required date. The site engineer inspects the plant and if everything is in order, he/she accepts the engagement. In some cases, the plant is sent back because the plant does not comply with the original specifications of the site engineer.

When the period of engagement is concluded, the supplier comes to pick up the plant. Sometimes, the site engineer requests an extension of the period of engagement (e.g. in order to keep the plant for an additional week). In order to request an engagement, the site engineer should first check the availability of the plant with the supplier for the period of the extension. If the plant is available, the site engineer can modify the plant hire request. Buildit’s information system then sends a modified PO to the supplier.

A few days after the plant is picked up, the plant’s supplier sends an invoice to Buildit by e-mail. The invoice is automatically matched with a PO by Buildit’s information system. The invoice should then be approved by the site engineer who hired the plant. The reason why this check is needed is to ensure that the supplier is not invoicing for a plant that was rejected or for an incorrect time period. If the invoice is accepted, payment is scheduled and a remittance advice is sent to the supplier. If the invoice is rejected, the payment is not scheduled and the site engineer is responsible for communicating with the plant supplier in order to resolve the issue.

Plant suppliers may send reminders of unpaid invoices (every invoice has a due date). When BuildIT’s system receives a payment reminder, it checks if the corresponding payment has been scheduled or not. If it has been made, BuildIT re-sends the corresponding remittance advice. If on the other hand the payment has not been made yet because the site engineer still needs to approve the corresponding invoice, the site engineer is notified (in one way or another) that a payment reminder has been received for an unapproved invoice. This is achieved for example by marking the unapproved invoice with a “late payment” note.

**Rentit’s Order-to-Cash Process**

The order-to-cash process at Rentit starts when a new purchase order (PO) for a plant hire is received. The PO refers to a specific plant (also called a Plant Inventory Item, e.g. a specific jackhammer with serial number 234136) of a specific type (also called a Plant Inventory Entry, e.g. Hilti Jackhammer model TE 905). Prior to placing an order, potential customers can query Rentit’s catalogue to check which plant inventory items are available during any period of time in the future.

When a PO is received, Rentit’s information system automatically checks if the requested plant is available during the requested period of time. If it is not, the PO is rejected. If the request plan is not available, an employee at Rentit needs to check the PO and either accept it or reject it (there may be other reasons why a PO is rejected even though the item is available). In any case, Rentit’s information system will send a notification to the customer whenever their PO is accepted or rejected.
Once a PO is accepted and the shipment date arrives, a clerk at Rentit’s depot will dispatch the plant to the customer’s site. They will then mark the PO as “dispatched”. The customer can cancel the PO anytime before it is dispatched, but not after. The customer may also change the rental period until the plant is dispatched.

Once a plant is dispatched, the only change that the customer is allowed to request is to postpone the date when the plant will be returned (i.e. to extend the period of the plant hire). Every change request must be approved by an employee at Rentit and it can only be approved if the plant is available during the extended period.

When the date of return arrives, the clerk asks the transportation provider to pick up the plant. When the plant is arrives back at Rentit’s depot, the clerk marks the corresponding PO as “returned”.

Rentit’s system automatically generates invoices for returned plants. The system monitors when a payment (remittance advice) is received and it then marks the invoice as “paid”. The system also sends reminders to the customer when the payment date of an invoice arrives and the invoice has not yet been paid.

**Requirements (12 points)**

The goal of this project is to implement a service-oriented system in order to support the above business processes (both on Buildit’s side and on Rentit’s side).

The functional requirements (features) for Buildit’s system are:

CC1. The system should allow site engineers to create a plant hire request.

CC2. The system should allow site engineers to modify a plant hire request prior to its approval by the works engineer.

CC3. The system should allow site engineers to cancel a plant hire request. Cancellations are allowed until the day before the plant is due to be dispatched. If a cancellation is requested after the PO has been sent, a request for cancellation should be sent to the supplier.

CC4. The system should allow site engineers to check the status of a plant hire request.

CC5. The system should allow works engineers to approve or reject a plant hire request. A works engineer may also modify the plant hire request and approve the modified request. The allowed modifications include changing the delivery site, or reducing the period of engagement (e.g. postponing the start date of the hire or bringing forward the end date).

CC6. The system should produce a PO for every approved plant hire request and forward it to the corresponding supplier. The supplier may respond with an acceptance or a rejection message, asynchronously.

CC7. The system should allow Buildit employees to view all submitted POs and their statuses.

CC8. The system should allow site engineers to request an extension in order to keep a plant longer than its initial period of engagement. When an extension is requested, the system should produce a modified PO and forward it to the supplier. The supplier may accept/reject the modified PO.

CC9. The system should allow a supplier to submit invoices for a given plant hire.

CC10. When an invoice is received, the system must check that the PO number in the invoice corresponds to an existing and unpaid Purchase Order. If the PO does not exist or has already been paid, an error message is returned to the supplier.

CC11. The system must allow site engineers to approve an invoice and to retrieve the PO associated to an invoice.
The system must submit a remittance advice to the supplier after the invoice is approved (normally the remittance advice should only be sent after the payment has been triggered, but in this project we do not deal with sending payment orders to the bank).

NOTE: One should keep in mind that Buildit’s accounts payable department handles hundreds of invoices per month, not only for plant hiring but for many other types of expenses such as construction material, office supplies, sub-contracting services, etc. In this project we will focus on building a sub-system for handling invoices for plant hiring, but we should keep in mind that the system should be extensible to handle many other types of invoices.

The above features are described mainly from the perspective of the construction company’s system (Buildit). The plant supplier (Rentit) also needs a system to handle its order-to-cash process (from the moment a purchase order arrives until the payment of the invoice is received). RentIT’s system should also allow BuildIT to query the catalogue. One can derive the features of the plant supplier’s system from the above ones. In particular:

PS1. The system should allow a customer to list the available plants and their prices.

PS2. The system should allow a customer to check the price for a given plant (given the plant identifier)

PS3. The system should allow a customer to check the availability of a given plant during a given time period

PS4. The system should allow a customer to submit a PO for hiring a plant. If the plant item is not available during the period indicated in the PO, the system should automatically reject it. If it is available, the system should show the PO as “pending” to the workers at Rentit. A worker at Rentit can accept or reject a PO (but the system should always prevent accepting a PO when the plant is not available).

PS5. The system should allow a customer to view the status of any PO they have submitted. Possible statuses are: pending, accepted, rejected, plant dispatched, plant returned, invoiced.

PS6. The system should allow a customer to submit modified POs requests and accept/reject a modified PO depending on the plant’s availability. PO modifications can only be used for modifying the period of plant rental. They cannot be used to change the plant inventory item that has been ordered (i.e. they cannot be used to order a different plant). If the plant item is not available during the whole (modified) rental period, the PO modification request should be automatically rejected.

PS7. The system should allow a customer to submit a request to cancel a PO. A cancellation request is accepted if the request arrives prior to the plant being dispatched. If the plant has already been dispatched, the cancellation request is rejected.

PS8. The system should allow employees at the plant depot to mark a plant as “dispatched”. This happens when the plant leaves the depot.

PS9. The system should allow employees at the plant depot to mark a plant as “returned”, meaning that the plant has been returned in due form and the rental is completed.

PS10. The system should submit invoices for “returned” plants.

PS11. The system should submit payment reminders for unpaid invoices, when the due date arrives.

PS12. The system should allow customer to submit remittance advices and it shall mark the corresponding invoice as "paid" once the corresponding remittance advice is received.

NOTE: The plant supplier has two separate sub-systems: one for handling plant hire requests, and the other for invoicing (i.e. accounts receivable).

Non-functional requirements
The Buildit system should implement role-based authentication. There are two roles: site engineer and works engineer.

There should be user interfaces both on the Renit and the Buildit side. These user interfaces can be very minimalistic and are only meant to demonstrate fulfilment of the functional requirements and the integration requirements.

The backend implementation must be done in Java Spring Boot and the frontend using Vue.js. You may use React for the frontend, but we will not answer technical questions about React.

**Integration requirements (2 points)**

Each team must complete one implementation of Buildit and one implementation of Rentit. Upon completion of the project, every implementation of Buildit should be integrated with two implementations of Rentit (the team’s own implementation of Rentit plus the implementation of another team). In this context, integration means the following:

- The Buildit system provides the ability to search the list of available plants and prices of three Rentit systems, either together (site engineer can do this in a single query) or separately (site engineer has to run separate queries to consult each Rentit system).
- The Buildit system can fulfil its functional requirements (placing an order, checking order status, etc.) with each of the two Rentit systems.

**Deployment requirements (2 points + up to 2 bonus points)**

Your Rentit and Buildit systems should be deployed on a server. We recommend you to use Heroku, but you can deploy on any (cloud) service. You can get bonus points by creating Docker containers with your RentIT and Buildit applications and deploying them on a cloud platform such as DigitalOcean.

If you cannot or do not wish to use public cloud services, we can provide you a Linux server in the University’s cloud to deploy your applications (ask the lecturer or lab assistants).

You will get points for deployment as follows:

- One point for deploying your Rentit system on a server (e.g., on Heroku). You will get one additional (bonus) point for creating a Docker container of your Rentit system, pushing the image to Docker Hub, and deploying it on a server (e.g., using DigitalOcean).
- One point for deploying your Buildit system on a server (e.g., on Heroku). You will get one additional (bonus) point for creating a Docker container of your Buildit system, pushing the container image to Docker Hub, and deploying it on a server (e.g., using DigitalOcean).

**Checkpoints (5 points)**

In order to encourage you to start early and to be iterative, there will be two checkpoint meetings with the lab assistants, on 7 May and 14 May during the practice sessions. At each checkpoint, you or one of your team members will explain and show your team’s progress to the lab assistant. The lab assistant will give 2.5 points for very good/excellent progress, 2 points for good progress, and 1.5 points for minimally satisfactory progress, and 1 point or less for unsatisfactory progress. For the first checkpoint, you should have a first draft of your domain model(s), state model(s), and API specification(s), and you should have covered CC1-CC7 and PS1-PS6, including the corresponding tests. During the first checkpoint meeting, you will agree with the assistant what will be your target for the second checkpoint.
Report (3 points)

You should submit a report with the following structure by 21 May at 14:00.

1. A title page including the names and student numbers of all team members. Only include team members who contributed to the project in a non-negligible way.

2. Any models to document your system design (both for BuildIT and RentIT). As a minimum you should provide, for both RentIT and BuildIT, a domain model, a state model of the main resources, and a description of the API in a table form (as shown in Lecture 6).

3. A list of the RentIT systems with which you integrated your BuildIT system, and brief explanation of how did you achieve this integration. For example, did you introduce adapters, or do you have multiple separate RentIT clients, or did you simply hack it all by mixing business logic code with integration code in your BuildIT system.

4. Links to the deployed RentIT and BuildIT systems (and the Docker Hub username and repository(ies) if you wish to claim points for Dockerization).

5. Links to the code repository(ies) containing all the code to build, deploy, and run your BuildIT and RentIT systems. Please note that we will check the commit history of your project to assess the contributions of each team member.

6. Any other documentation you feel would support your submission, but please keep it short.

Oral presentation (2 points)

- The oral project presentations will take place on Tuesday 21 May during the practice session (14:15-16:00).

- Each team will have 8 minutes to present. You should cover the following points in your presentation (in any order):
  - Very brief overview of the architecture of your RentIT system and your BuildIT system. Indicate if there are parts of either system that you did not manage to implement.
  - With which team did you integrate? How did you implement the integration (e.g. did you implement a Façade, an adapter)?
  - Short demo showing how one rental request flows all the way from its creation to the corresponding invoice being paid. During the demo, the site engineer should request a deadline extension for keeping the plant longer.

Testing (4 points)

You must implement automated acceptance tests (cucumber scenarios) covering at least the requirements. In the report, please indicate where to find the corresponding test for each requirement.