In this project, we will design and implement a collection of inter-connected backend components and front-end applications to showcase the principles of resource-oriented architectures. The domain is that of plant hire (also known as equipment rental). A description of a scenario is given below. The description may be ambiguous or incomplete at some points and it is your task to resolve any of those ambiguities and to complete the missing information. Do not hesitate to ask questions to the lecturer or lab assistants about ambiguities or design choices you make.

For your project, we will take as the starting point the scenario that we used during the course (i.e. lectures, practicals and assignments). For completeness, you are required to deliver also the full implementation of the procurement business process (Order-to-cash from RentIt perspective and Procure-to-pay from BuildIt perspective), as described in this document.

**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)

**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 one plant, he/she consults the catalogue of a plant supplier in order to identify the plant that fulfils his/her requirements. Once the site engineer has identified the required plant, he/she checks its 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 (PHR), which includes the
identifier of the site engineer making the request, the identifier of the construction site where the plants are 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.

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. 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 (or in about 6 working hours if the request is urgent). The site engineer inspects the plant and if everything is in order, he/she accepts the delivery. In some cases, the plant is sent back because the plant does not comply with the original specifications of the site engineer. In this project, we will not consider what happens when a plant is rejected by the site engineer, except for the fact that RentIT will not send an invoice for a rejected plant, while BuildIT will not pay an invoice for a rejected plant.

Sometimes, the site engineer requests an extension of the period of engagement (e.g. to keep the plant for one additional week). To this end, the site engineer requests the extension via BuildIt’s information system, which forwards it to RentIt’s. If RentIt’s information system finds that the plant is available for the requested period of time, it will automatically adjust the purchase order (i.e. create a new reservation and update the overall rental period and cost) and notify the customer that the extension has been accepted. If the period of the extension overlaps with the engagement with another customer and a compatible plant is found by the information system (e.g. a 3 tons excavator can be a replacement for one with capacity of 2 tons), RentIt’s information system will notify the customer that the extension has been processed. In this case, an employee at RentIt will verify the loss implied by the changing the plant and, if the loss is less than 30% of the overall income (total amount for rentals –using the price of the original plant—minus the cost implied by the replacement), the employee will organize the replacement (i.e. the replacement plant will be sent to BuildIt and the original plant will be picked-up as planned). If the cost of the loss exceeds the 30% of the overall income, RentIt’s employee will reject the extension request. As part of the notification of the rejection, RentIt’s information system will include a comment describing the possibility of replacing the plant. In the latter case, the replacement would be done via a new Plant Hire Request/Purchase Order. If the requested plant is not available and no replacement is found, RentIt’s information system will reject the extension request and notify BuildIt about the decision. In any case, BuildIt’s information system should update the state of the plant hire request accordingly.

When the period of rental is concluded, the supplier comes to pick up the plant. A few days after the plant is picked up, the plant’s supplier sends an invoice to Buildit. The invoice is automatically matched with a PO by BuildIt’s information system based on the PO identifier included in the invoice. 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.

Requirements (5 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 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 delivered. 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, reject or modify a plant hire request.

CC6. The system should produce a PO for every approved plant hire request and forward it to the corresponding supplier. The supplier may respond that the plant being requested is no longer available (which means the PO is rejected), or it may respond with a confirmation of the PO.

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

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 an invoice for a given PO.

CC10. When an invoice is received, the system must check that the PO identifier in the invoice corresponds to an existing and unpaid PO. If the PO does not exist, an error message is returned to the supplier.

CC11. The system must allow site engineers to retrieve the PO associated to an invoice and to approve an invoice.

CC12. 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 from the perspective of the construction company’s system (BuildIt). The plant supplier (Rentit) also needs a system. 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. The PO may be accepted or rejected depending on the plant’s availability.

PS5. The system should allow employees at Rentit to determine which plants need to be delivered on a given date

PS6. The system should allow a customer to view the status of a PO (PO accepted, PO rejected, plant dispatched, plant delivered, plant rejected by customer, plant returned, invoiced)

PS7. The system should allow a customer to submit an extension request for a given PO. If the plant is available for the requested extension period, the system should accept the request. If the plant is not available for the requested period, but the system finds a replacement such that the loss does not exceed 30% of the income, the system should accept the request and adjust the PO accordingly. If the plant is not available or the loss implied by a replacement exceeds 30% of the income, the system should reject the extension. If the extension is rejected but a replacement exists, the system should propose the customer the replacement.

PS8. The system should allow a customer to submit a request to cancel a PO. A cancellation request is normally accepted if the request arrives prior to the plant being dispatched. If the plant has already been dispatched, the cancellation request is rejected.

PS9. The system should allow employees at the plant depot to mark the plant as “dispatched”. This happens when the plant leaves the depot.

PS10. The system should allow employees at the plant depot to mark the plant as “delivered” or “rejected by customer”. This latter option happens if the customer did not accept the plant because the plant did not meet the specifications in the catalogue.

PS11. 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 period has expired.

PS12. The system should submit invoices for “returned” plants.

PS13. The system should submit payment reminders for unpaid invoices.

PS14. The system should allow customer to submit remittance advices and 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

Both systems, i.e. RentIt and BuildIt, should implement role-based authentication. BuildIt’s system have at least two roles: site engineer and works engineer.

- User interfaces can be minimalistic and are only meant to demonstrate fulfilment of the functional requirements and the integration requirements.
• The implementation should normally be done in Java/Spring. However, exceptions to this rule can be made if you explicitly ask for an exception in advance. Please ask authorization from the lecturer if you prefer to implement your system (BuildIT and/or RentIT) using another technology (e.g. Ruby).

**Integration requirements (10 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 three implementations of Rentit (the team’s own implementation of Rentit plus the implementations of two other teams). 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 fulfill its functional requirements (placing an order, checking order status, etc.) with each of the three Rentit systems.

**Testing (6 points)**
You must implement test cases covering all the requirements. You can use both integration (i.e. tests on rest controllers) and acceptance test (i.e. cucumber scenarios). For each one of the requirements, in a document, indicate where to find the corresponding test and the type of test.

**Report and APIs (6 points)**
You should submit a report with the following structure by 28 May at 20:00 EET.

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 resource model and a state model for each resource type with non-trivial lifecycle.
3. A URL to your Apiary blueprint for BuildIT and your Apiary blueprint for RentIT, as well as root URLs of your BuildIT and RentIT systems.
4. Any other documentation that you feel would support your submission, but please keep it short.

**Oral presentation (3 points)**
• The oral project presentations will take place on Monday May 28th during the lecture/practice session time-slots (12:15-14:00 and 16:00-18:00).

• Each team will have 8 minutes maximum to present their architecture and their system. You should cover the following points in your presentation (in any order):
  o Brief overview of your domain model and API for BuildIT and RentIT highlighting your main design choices.
  o With which teams did you integrate. How did you achieve the integration (e.g. did you use adapters or set up a facade?)
  o Short demo showing how one rental request flows all the way from its creation to the corresponding invoice being paid. During the demo, you should show at least one usage scenario where the site engineer requests a deadline extension for keeping the plant longer.