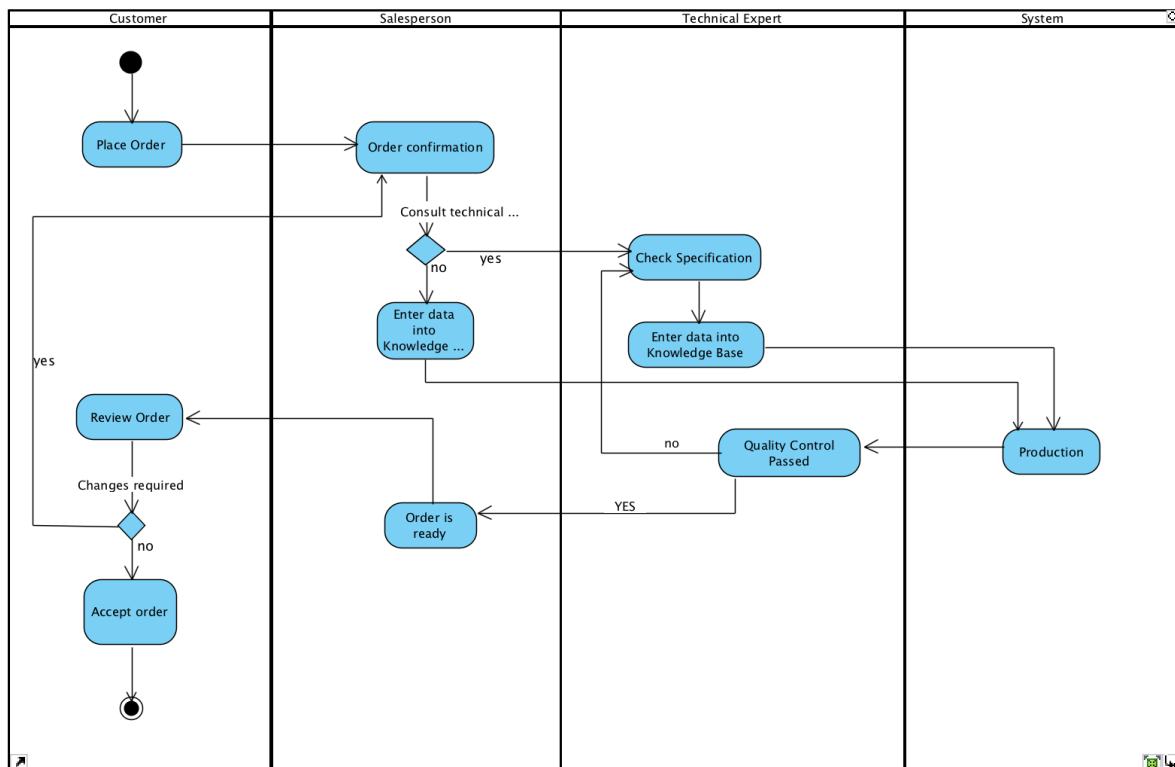


## Fifth Iteration Report

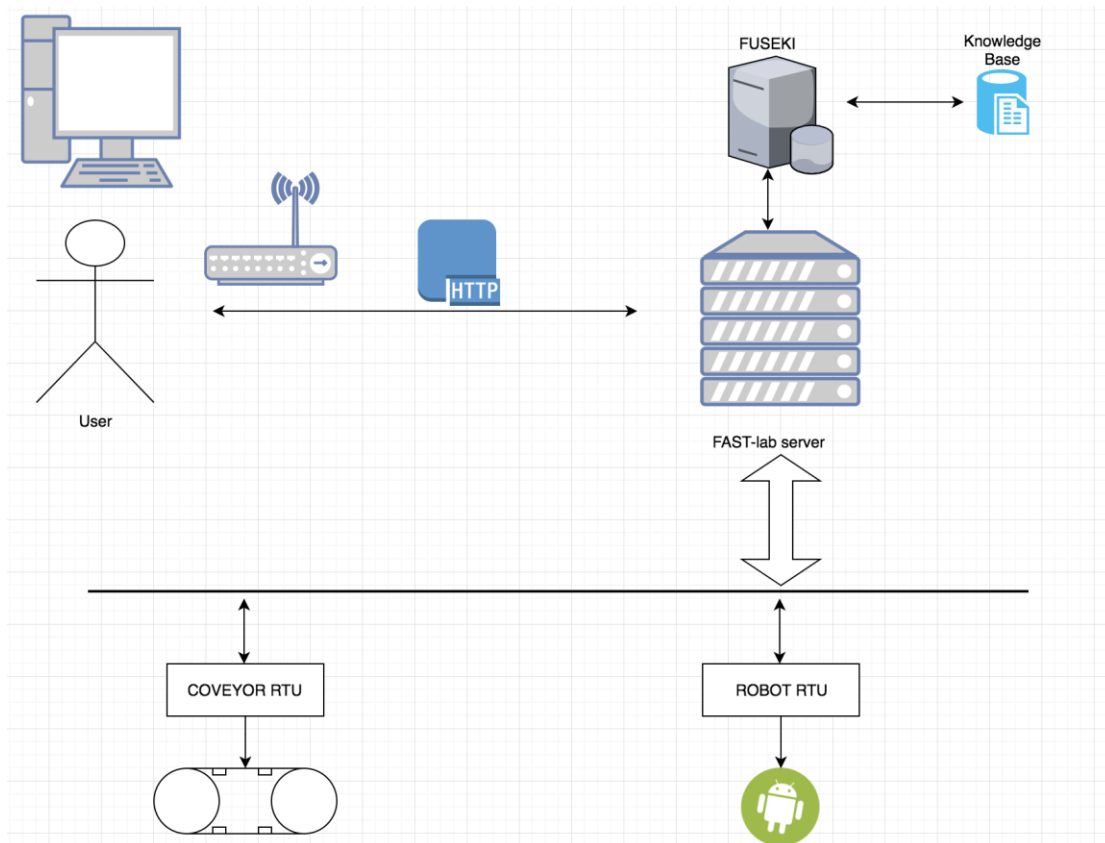
This is group seven's fifth iteration report for the Introduction to Industrial Informatics course work. In this report, went through the diagrams from last iterations and decided to create a new diagram to illustrate our project's processes. We made majority of coding which works with the simulator properly and the knowledge base which contains the components of the FASTory manufacturing line, order information and its properties as well.

### Activity Diagram

For this iteration, we refine the activity diagram to demonstrate the workflow of our project. This diagram is illustrated in the picture 1 below.



Picture 1. Activity Diagram



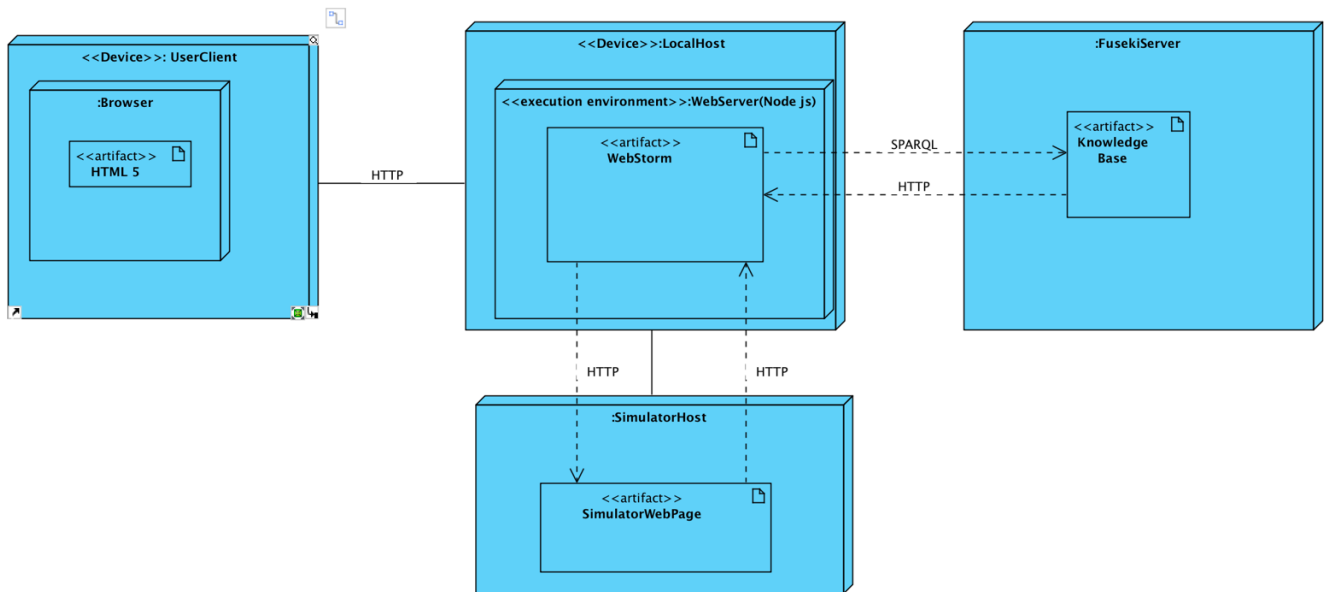
**Picture 2.** System Architecture

In these two diagrams, we have represented the workflow of the whole process of our project going on. Initially, the customers place the order on our UI, then it goes to our database. Our salesperson checks the order and confirms it then. If the specification of products has already existed in the knowledge base, we proceed it to the production line. If not, our technical experts need to do some changes in our production line. When it goes into the production line, the factory is rather automated and could finish the task properly. After production is done, it proceeds to quality control progress. If the products are qualified, our sales department would inform the customers to review the order again. If the product does not pass quality control, we must check the problem and put into production line once more.

If customers are satisfied with products, we will do the last step-delivery, otherwise, we might negotiate till both are satisfied.

## Deployment Diagram

The deployment diagram we developed for our project is illustrated in the picture 2 below. It has the information that shows how different mechanism interacts.



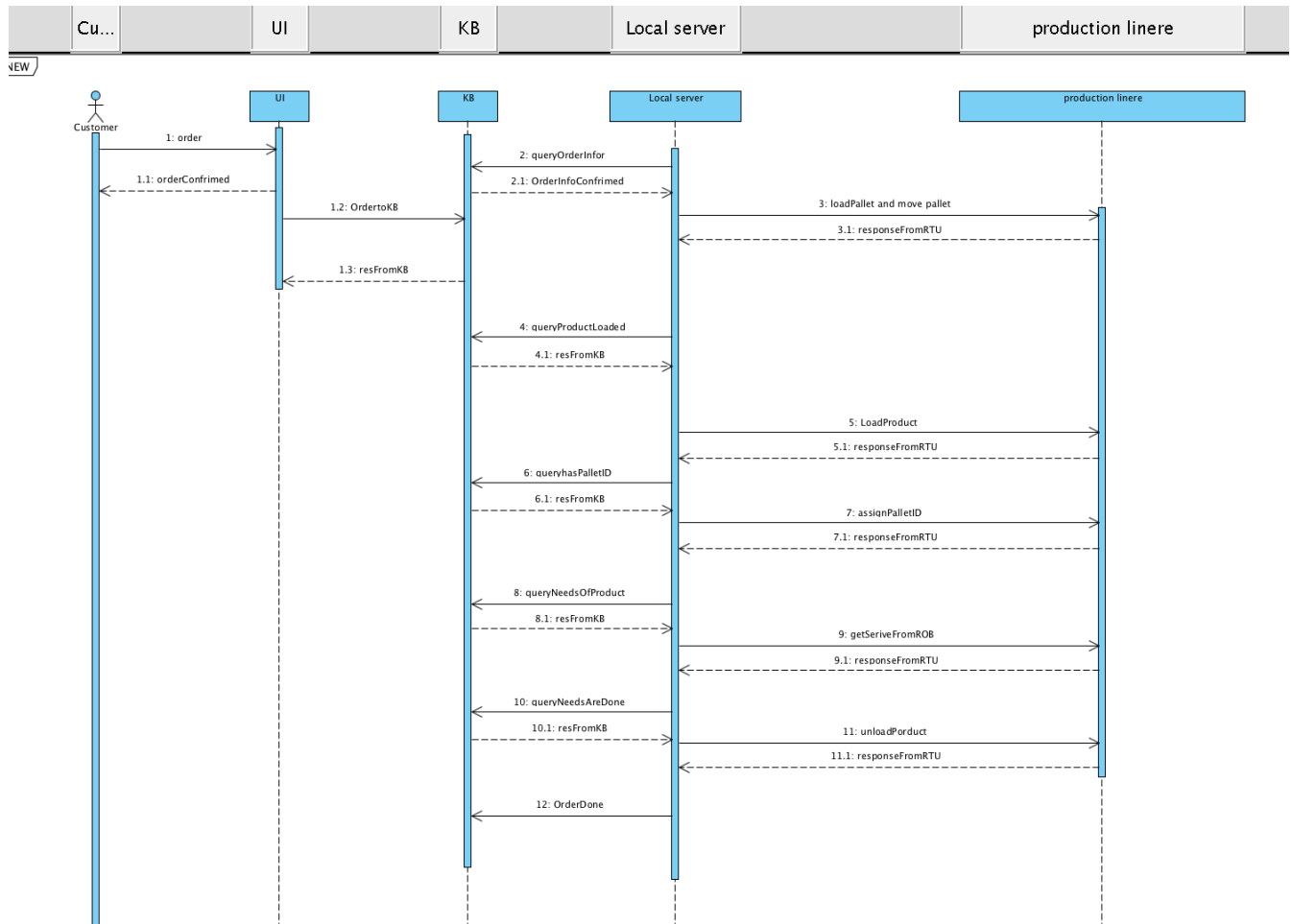
**Picture 3.** Deployment Diagram

On the customer side, the server runs on its device and they use a browser like Firefox or Google Chrome visiting our website. The page will be presented in HTML. The client-server communicates with our server by HTTP protocol. When the server receives the order, it will parse the important information and send it to the knowledge base. The localhost interacts with Fuseki server with SPARQL query via HTTP protocol. Also, the localhost connects with each RTU from production line by HTTP.

Each time, when the conveyor or robot moves, it communicates with the knowledge base to call the next action. Those three servers (Fuseki server, localhost, RTUs) constantly interact to ensure best options for pallet movements.

## Sequence Diagram

The sequence diagram we refined more focus on the detailed workflow of our whole project. It is illustrated below:



**Picture 4.** Sequence Diagram

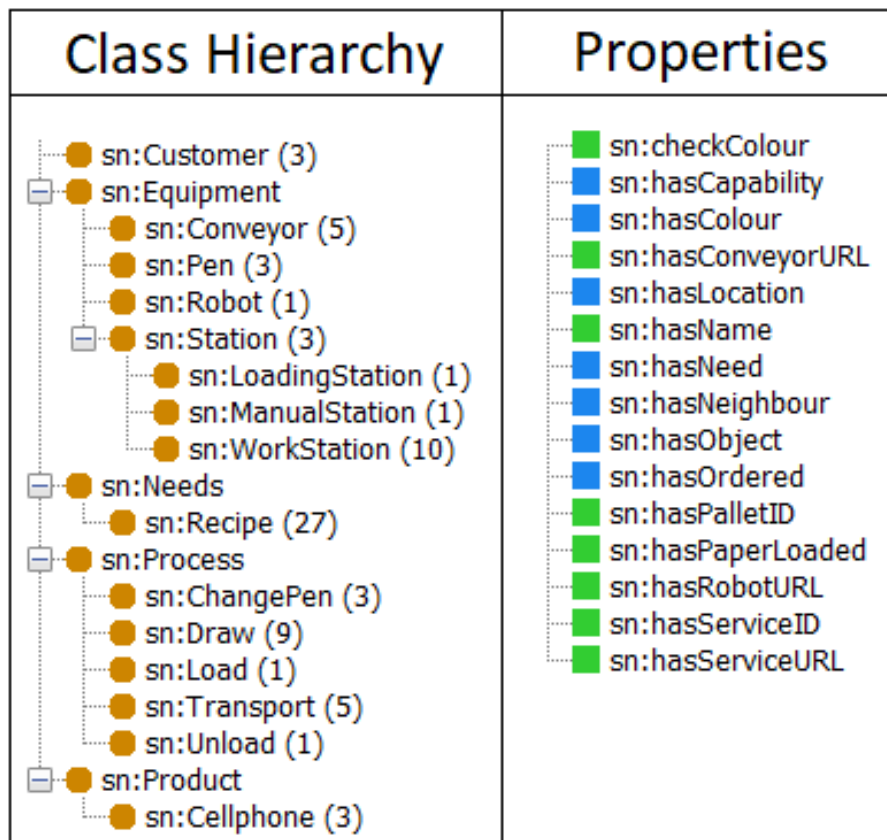
In this diagram, it shows the sequence of each step about when it should be happening. The customer places an order via our UI. Once the order is confirmed, the specification of the products will be inserted into the knowledge base. From this step, the Fuseki server starts to interact with localhost all the time. Localhost makes a query and gets the response from the knowledge base. The commands for RTUs is encompassed in each response.

When the conveyor moves, the localhost knows the corresponding events. According to the events, the localhost will communicate with Fuseki and then command the RTUs.

The whole process on the production line could be illustrated like this: first of all, the loading robot loads a pallet and keep the pallet moving on. If the pallet has no paper, it goes to all bypass conveyors to load a paper as soon as possible. Once it gets a paper, it moves to get service from a robot. At zone 1 of each workstation, it checks the needs of paper to ensure if the pallet should go into the workstation or bypass conveyor. If the need matches the service, it should move into zone 3. Before it proceeds to do a movement, the pallet checks the status of neighbors. A pallet will not move until one of the neighbors is free. After all the needs are done for one pallet, it goes to workstation 1, zone 3 where it gets unloading service.

## Knowledge Base

For this Iteration, we created refined the first version of our knowledge base using the tool Olingvo. The updated collection of the class hierarchy and the properties used in our knowledge base are represented in picture 3 below. The biggest change was that we incorporated the customer to our knowledge so that we can track its order requirements and when they are met. This way we get the information of completed orders in real time while the manufacturing line is running.



**Picture 5.** Representation of the Knowledge Base

The customer class was added to the class hierarchy to keep track of the different orders. Our knowledge base has three customers by default at this moment for testing purposes. The equipment class consists of two levels. The first level can be thought to contain different pieces of equipment that show up in more than one station which brings us to the second level which consists of the mentioned stations. The connection between these two levels is formed with the property hasObject. For instance, each type of station has a certain number of conveyors but only workstations have pens. The needs class holds within all the possible recipes that can be assigned to a product. The quantity of different recipes in this project is 27 because of the options to have three different parts, models and colors. The process class consists of all the different processes that can be achieved by the robots and the conveyors and the product class holds within all the ordered cellphones which have some needs assigned to them.

Each class can have various individuals and these individuals can have various properties and with these class-property relations, different queries can be formed in our program to extract specific information that is needed to run the manufacturing line. The two different colors (green and blue) in

the properties section of the picture 3 describe different types of class-property relations. The properties that are marked with the blue color have a range in which they operate in the knowledge base. The ones marked with the green color on the other hand don't have a range in the knowledge base, but a value assigned to them. Depending on the nature of the property we have used values of Boolean, String or Integer type.