TAMPERE UNIVERSITY OF TECHNOLOGY

Factory Information Systems

Course Assignment

Group member:

Farid Khosravi 267964

Mehdi Mahmoodpour 267958

Palash Halder 267962

ASE – 9476 Winter 2017

Contents

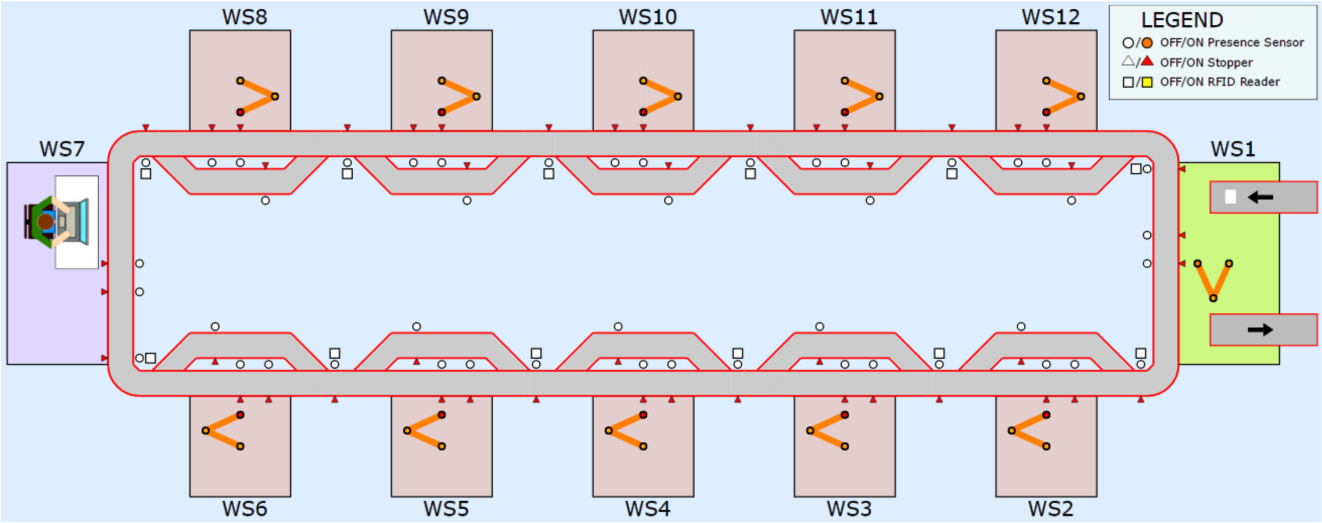
Topic: Information system for the Factory making mobile phones

Objective:

- Understanding of **Factory Information System** concepts.
- Ability to design and implement information systems
- Ability to implement monitoring the information of production process at the plant-floor level of system.

Introduction to system:

- The factory line was being used to assemble and produce cell phone in Nokia Company. Now it is utilized for education purposes in TUT university.
- It consists of 12 work stations. Pallet loads/unloads to/from the line from work station 7 and the paper loads/unloads on the pallet in work station 1.
- Each work station contains one robot and two conveyors.
- Each pallet has a unique Pallet-ID number.
- The system can produce 729 different types of mobile phones (3 frames * 3 colors * 3 screens * 3 colors * 3 keyboards * 3 colors). In all the cases colors, can be either red, green, or blue.
- The following picture shows the overview of the FASTory line :



Overview:

Factory Information Systems (FIS) is utilized to help manufactures to optimize the production process by analyzing the captured information from the entities within the system. This leads to decrease costs of production and increase the quality of products.

Tools and environment:

- Fastory line simulator.
- Visual Paradigm: In this assignment the “Visual paradigm “software is used to display the diagrams of components of the system based on the Unified Modeling Language (UML).
- MySQL: To make database to implement the project.
- Node.js: To implement monitoring the information of production line.
- Webstorm: to run and implement the JavaScript codes.
- Web browser: The UI for the system implemented.

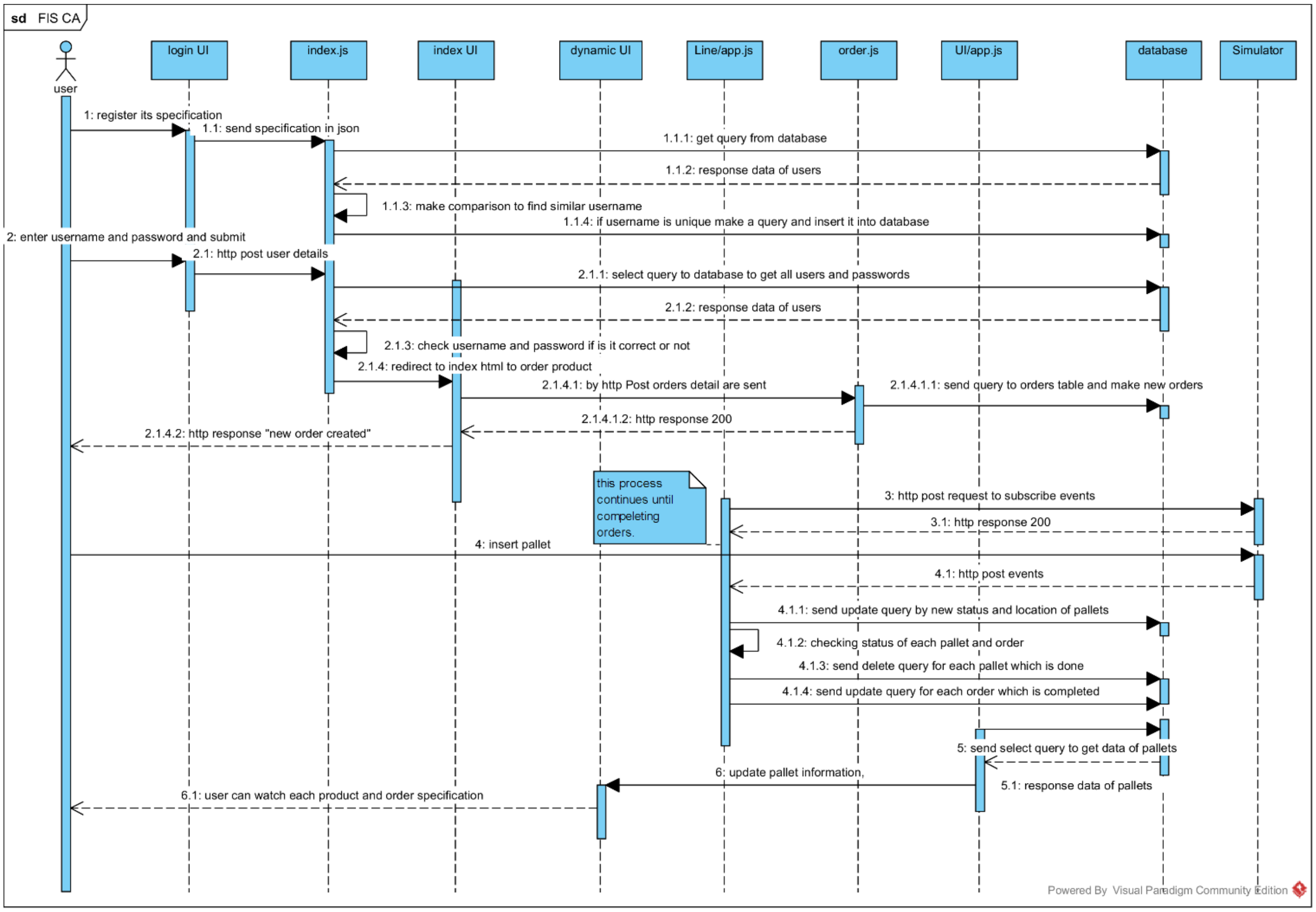
Design:

The design of application is based on the following:

- Fastory line simulator at **localhost:3000** is used to move pallets and make events.
- **Login.HTML** form is used to register new users and login existing users. New users are added by making a POST request to **localhost:3333/register** and via **Line/index.js** the user data is saved in **customers** table in MySQL database **fis_assignment**.
- After logging in, only the authenticated user is redirected to Index.HTML for order entry. Invalid users are not allowed and are requested to register first.
- **Index.HTML** form is utilized to get the details of needed product from the customer. Different type of products and the required quantity of each model can be selected by customer. The user inputs will be sent to **localhost:3333/orders** by making a POST request. Then data is stored in **orders** table in MySQL database **fis_assignment** by sending query from **Line/orders.js**.
- **Line/routes/orders.js** Receives POST data containing the order parameters and stores data in MySQL database. Finally, it responds to the client on completion of request.
- **Line/app.js** is the orchestrator for the entire program. We have used Express app to make our server. It is used to subscribe to the events of the Fastory line simulator. On receiving the events **Line/app.js** updates the information of pallet including its location and status in MySQL database **pallet_info** table due to the previous location and status of pallet.
- **UI/app.js** makes pallet_info database on start. It also queries the database **fis_assignment**, table **pallet_info** and table **orders** too. On querying the **orders** table the order requirements

are received periodically. Similarly, periodic query to **pallet_info** table is made to receive real time line data.

- The view is provided by **UI/views/index.ejs. UI/app.js** makes a **socket** connection with **index.ejs** to provide real time update of the view.


sd FIS CA
user
login UI
index.js
index UI
dynamic UI
Line/app.js
order.js
UI/app.js
database
Simulator
1: register its specification
1.1: send specification in json
1.1.1: get query from database
1.1.2: response data of users
1.1.3: make comparison to find similar username
1.1.4: if username is unique make a query and insert it into database
2: enter username and password and submit
2.1: http post user details
2.1.1: select query to database to get all users and passwords
2.1.2: response data of users
2.1.3: check username and password if is it correct or not
2.1.4: redirect to index html to order product
2.1.4.1: by http Post orders detail are sent
2.1.4.1.1: send query to orders table and make new orders
2.1.4.1.2: http response 200
2.1.4.2: http response "new order created"
this process continues until compeleting orders.
3: http post request to subscribe events
3.1: http response 200
4: insert pallet
4.1: http post events
4.1.1: send update query by new status and location of pallets
4.1.2: checking status of each pallet and order
4.1.3: send delete query for each pallet which is done
4.1.4: send update query for each order which is completed
5: send select query to get data of pallets
5.1: response data of pallets
6: update pallet information,
6.1: user can watch each product and order specification
Powered By Visual Paradigm Community Edition

REAL LINE IMPLEMENTATION:

- The code is written for implementation at the real line, we are creating **XML data** which can be sent to the **raspberry pi** orchestrator and run the real line autonomously.
- For receiving the events from the real line, we used different type of **body-parsers**. We observed that the events generated at the line were not **JSON**, and as such we were receiving error **415(Unsupported Media Type)**.
- Next we tried with **bodyParser.text({type: ‘text/plain’})** we received, successful POST from the line but the body of the events was **empty-{}**.
- As such we implemented this Course Assignment based on simulator. However, this code can be easily implemented with the real line if the events are coming properly. The codes related to **XML data** format is commented in our code.

SKILLS ACQUIRED:

- Acquaintance with Node.JS, MYSQL.
- Learning how to design and implement information system for a production line.
- Learning how the flow of information can be utilized to monitor the production process.

GROUP MEMBERS’ INVOLVEMENT:

- Palash Halder: Making HTML interface,writing Node.JS application and implementing database.
- Farid Khosravi: Making HTML interface, writing Node.JS application and implementing database.
- Mehdi Mahmoodpour: Making HTML interface, writing Node.JS application and implementing database.