Sri Lanka Institute of Information Technology

B.Sc. Honors Degree in Information Technology

Specialized in Software Engineering

Final Examination
Year 3, Semester 1 (2021)

SE3040 – Application Frameworks

Duration: 4 Hours

June 2021

Version – Student ID ends with 4 (Four)

Instructions to Candidates:

- This paper has 1 question.
- The total mark for the paper is 100.
- This paper contains 4 pages, including the cover page.
- Please see the instructions at the end of the paper

A Hardware store decides to create an online application to sell their hot selling items. They have selected a third-party delivery service to deliver their products. You have been recruited as their full stack developer.

Note the following points as well.

1. The system should be built around the hot selling items and the delivery. One delivery and the products share many to many relationships. For example
    a. Products: Sand, gravel, drill, nails, box bar, hammers,…
    b. Delivery: trucks, bikes, three wheelers, vans,…
2. Delivery method should list all the products they can deliver.
3. You can decide which type of vehicles can deliver which products. (No need to think of the quantity)
4. Following should be handled in terms of data.
    a. Get all the products.
    b. Get all delivery vehicles.
    c. Get product in each delivery type.
    d. Add new products.
    e. Add new vehicles. Assign products to vehicles.

    All the above services and not CPU heavy (please consider this when you are selecting the technology).

2. You need to develop another service where you should calculate the total charges per delivery. You can decide the priced for the delivery and the products. This service is expanding and will be a CPU heavy task, please consider this when you are selecting the technology.

1. RESTful services are required for the following.
    a. Get all the products.
    b. Get all delivery vehicles.
    c. Get product in each delivery type.

d. Add new products – pass all the vehicles that product belongs to as a list of IDs.
   Ex:
   {
   code: “P001”
   name: “Hammer”
   amount: 890
   inStock: 10
   vehicles:[Vehicle IDs …]

   }

2. Product and Vehicle data structures should be flexible. Please note the below example data structures.
   a. Food
      i. Code – P001
      ii. Name – Hammer
      iii. Amount – 890
      iv. inStock – 10
   b. Vehicle
      i. type - Three-wheeler
      ii. owner – “Nimal”
      iii. Description – available
3. There is no concern for transaction control and consistency in this system.

Architect suggests from following technologies, and you must select the best suitable technology depending on the requirements being provided **(You do not need to use all the technologies).**

1. ReactJS
2. Node JS/Koa JS/Express JS
3. MongoDB
4. Spring Boot

Deliverables

Please follow the steps to upload the deliverables to the submission link.

1. Create a folder named your student ID number.
2. Create two folders name Frontend and Backend inside the created folder above.
3. Start the frontend and backend development in the relevant folder.
4. Make sure there are no compilation or run time issues in your ReactJS project.
5. Remove the node_modules folder from the project frontend folder structure.
6. Make sure there are no compilation or runtime issues in the **backend** project.
7. Remove the node_modules folder from the project backend folder structure.
8. If you develop the backend application using Spring Boot, **keep only the relevant** files including the source folder and the pom file.
9. Create a PDF document including the following,
    a. Screenshots of all user interfaces of the frontend application.
    b. Screenshots of all MongoDB collections.
    (Please see the template report provided.)
10. Put the PDF file in the frontend project folder.
11. Zip the project folder with the student id number (including both frontend and backend) and upload using the provided submission link.
12. Note that the zip file must be less than 30 MB. If the size is more than that try removing assets in the frontend project if you used any.

Marking Rubric

1. Implementation of User Interfaces
   - 24 marks
2. Implementation of the service endpoints (get, add, …).
   - 30 marks
3. Implementation of the calculation service.
   - 24 marks
4. Suitable technology selection (Student will be eligible for this mark if only he/she select the correct technologies for the different parts of the application).
   - 10 marks
5. At least one unit test for either part of the application (UI / services).
   - 5 marks

6. Styling of the UI.
   - 2 marks.
7. Coding standards and quality
   - 5 marks

Note: Following are considered when awarding marks for Coding standards and quality.

1. Following the REST architecture (resource paths).
2. Variable, function, and class naming.
3. Clear code and directory structure.

--- End of the Paper ---