4DV609 – Project Development
Microservices-based SmartShopping

In this assignment you should work *in group*.

If you have questions, post them on the forum. It is ok to help each other in public forums. For any question about the Assignment use **"Assignments Forum"** to get in contact with us.

Please note that all tasks should be documented, that is, a model without text that explains it has little or no value.

Time log

Track the time you spend on this assignment. Create a table where you log time spent on every activity for each task. Include the log in your submission.

Grading

Your submission will receive a grade from A to F where F is Failed. You are allowed to improve your work after the initial submission before the deadline. **The grade is final**, *i.e., you will not get an opportunity to correct/improve after grading*.

Your answers should be your own! You are not allowed to copy code, models, or texts (books articles, blogs, wikis) in your answers! Each submission will pass through a plagiarism/clone detection system before correction. If plagiarism is detected, the assignment is failed and a formal investigation will be initiated.

Submissions that arrive after the due date will be downgraded by 25 %-units.

Evaluation Process

Submit to MyMoodle by **November 01st (11.00AM)**

1. A .pdf file containing the Team Report
2. A .pdf file containing the Individual Report

Final Presentation and Demo on November 02nd (10.15-12.00)

- Prepare a short PowerPoint presentation summarizing
    - Team Report
    - Individual Repor

Mauro Caporuscio mauro.caporuscio@lnu.se

Department of Computer Science Lnu.se

Project Description

Nowadays, there are many stores which offer the same or similar products, but the prices often vary from store to store. In today's times of the recession, people are trying to buy products where they are cheaper and save money. This turns out to be a challenging task because of the sheer number of stores and products that they offer.

SmartShopping is envisioned as a mobile application which should ease up the daily process of buying groceries for the end users. SmartShopping will enable users to enter groceries they want to buy to a list, and then find out in which store they can buy the cheapest groceries and save money, or find the closest store with the requested products. Although, there are some existing web sites and applications with purpose of finding cheapest items across multiple stores, they do it on the basis of a single item.

SmartShopping on the other hand, will do it for the whole list of groceries. For example, given the following input:

- **Stores**: City Gross, ICA, Netto, Willys, Lidl, Coop
- **Groceries**: Carrots (1kg), Spinach (0.5kg), Pumpkin (0.2kg), Potatoes (2kg), Tomatoes (2 x 0.250kg)

SmartShopping should provide the user with the following output

1. Ordered by *stores distance*:
   - ICA Quantum (0.5km): Carrots (10kr/kg), Spinach (15kr/kg), Pumpkin (100kr/kg), Potatoes (5kr/kg), Tomatoes 250g (25kr/st) -> Total = 10 + 7.5 + 20 + 10 + 50 = 97.5kr
   - Willys (1.4km): Carrots (12kr/kg), Spinach (20kr/kg), Pumpkin (90kr/kg), Potatoes (6kr/kg), Tomatoes 250g (15kr/st) -> Total = 12 + 10 + 18 + 12 + 30 = 82kr
   - Netto (1.9km): …
   - CityGross (5.3 km): …
2. Ordered by *products price*
   - Carrots: ICA Quantum (10kr/kg), Willys (12kr/kg), …
   - Spinach: Netto (10kr/kg), ICA Quantum (15kr/kg), …
   - Pumpkin: Willys (90kg), CityGross (92kr/kg), …
   - Potatoes: …
   - Tomatoes: …

Mauro Caporuscio mauro.caporuscio@lnu.se
Department of Computer Science Lnu.se

Requirements

SmartShopping must make the grocery shopping easier for the user and provide the functionality of finding the cheapest / closest store which has the products the user wants. System must be constantly updated with up-to-date prices either by gathering information from stores' web pages or by manual input from users.

Functional requirements

User shall be able to:

- browse all products in the system,
- add, modify quantity and remove products from the list (shopping list),
- add / update information about a product, including its price and the store where the product is located,
- get suggestions of stores according to his shopping list and selected criteria (*products price* or *stores distance*)
- browse stores nearby on map

Application shall be able to:

- update the products and stores nearby on user demand,
- store user shopping list with quantity of products,
- display a single product's details (price in different stores),
- display the *cheapest stores* or the *closest stores* containing products in the list.
- automatically collect products and prices from available store's web site (web scraper),
- communicate with Client applications across different platforms,
- automatically insert the stores and their locations into the repository (store loader).

Mauro Caporuscio mauro.caporuscio@lnu.se
Department of Computer Science Lnu.se

Task 1 – Team Report
Provide a Design Document reporting how you exploited the Domain-Driven Design, and Message/Event-Driven Architecture, to develop the MS SmartShopping.

A. Outline of the design:

- Domain-Driven Design
- Identified Services
- Service APIs and Collaborations

B. Design details[1]:

- Describe the Microservice composition designed (i.e., used MS Patterns)
- Describe the Message/Event-Driven Architecture designed

C. End-to-End Testing

- Acceptance test for the SmartShopping application

Task 2 – Individual Report

A. Microservice design and implementation

- Detailed design of each implemented Microservice, including UML diagrams (e.g. Class diagrams)
- Source code for the Microservice

B. Testing

- Unit test
- Integration test
- Component test

[1] **NB:** Motivate all Design decisions