Programming & Systems on the Web Lab *Academic Year 2022 - 2023*

Subject: Collaborative offer search platform for consumer products

Version 1.0 11/28/2022

Target

The aim of this work is to develop a collaborative system for registering, searching and evaluating super-market consumer goods offers between users. The system allows its registered users to inform others about the existence of products that are at a good (by their own criteria) price, working alongside existing tools (such as e-katanalotis), which only inform about price but not stock, but also not including all super-market stores (eg local chains, mini-markets).

The control of the validity of the offers is left to the other users, who have the possibility to validate the offer (like) or show that it is not valid (dislike) or that the product is sold out (out-of-stock). Users can also indicate that an offer has been terminated. To encourage participation, the system rewards users who offer information with "points".

Functional Specifications

There are two types of users in the system: Administrator and User.

User

The interaction with the user is through an adaptive website that allows access via desktop or mobile phone, and has the following features and specifications:

1) **Registration in the system.** The user registers and gains access to the system by choosing a username & password of his choice, and providing his email. The password must be at least 8 characters long and contain at least one capital letter, a number and a symbol (eg #$*&@).

2) **Show map.**

   a. The map initially focuses on the user's current location and displays stores with active offers.

   b. Search (filter) stores (POIs), based on their name: Markers appear on the map for stores with a specific name. Stores that have at least one offer for any product, regardless of category, are shown with a different marker than those that have no current offer.

   c. Search (filter) offers by general product category: Markers appear on the map only for stores that have at least one offer on a product belonging to the selected general category

d. Show offers: By clicking on the marker of a store that has one or more offers available, the name of the store and the offers it has with the following details will appear in a pop-up: product, price, criterion fulfillment indicator (icon) 5 .a.i or 5.a.ii, registration date, number of likes/dislikes, stock (yes/no).

e. Evaluation of offers: If the user is within 50 meters of a store where there is an offer, in the pop-up mentioned in point d above, a button appears that can take him to a separate page entitled "Evaluation". On this page, the offer list appears again with the same information as above. The user can select one of these offers and see more details about the offer, including the above in d, a photo of the product and information about the user who submitted it (username & total score). The user can interact with the offers by liking or thumbs-up, as well as the opposite (dislike or thumbs-down) (See *point 5. Evaluation Score).* Users can also report that the product is out of stock or (back) in stock. If a product is out of stock, the like/dislike features (grayed out) are not activated.

f. Ability to select a store to submit an offer. If the user is within 50 meters of a store, by clicking on the marker that shows it on the map (regardless of whether it has an offer), in the pop-up that appears there is an "Add Offer" button that enables a new offer to be submitted (see *point 3. Submission of an offer).*

3) Submit an offer: Once a store is selected to submit an offer, a hierarchical list of categories (accordion or dropdown), subcategories and products (depth 3, the last level is products) and a quick product search box to find the product are displayed. By selecting the product either from the list or from the search box, the user can fill in the price he considers an offer and submit it for publication, in which case he will be informed of the result of the evaluation check (See point 5. Evaluation Score) . An offer is automatically deleted after one week has passed since its submission. If a week has passed, but nevertheless criterion 5.a.i or 5.a.ii is still met. below, the offer is renewed for another week, but the user who submitted it does not receive points again. A user cannot submit the same offer, i.e. for the same product and in the same store, as long as an existing offer is active. Exceptionally, if a new price 20% lower than the corresponding active offer is found, it is allowed to be listed, and the user receives a score according to the criteria. An offer for the same product but in a different store is considered a different offer.

4) Tokens system: The system rewards its users with a token distribution system. For each registered user, a fixed number of tokens (100) are generated every month. At the end of the calendar month, the system distributes 80% of these tokens to users, in proportion to the evaluation score they have achieved within the month, as a reward. Therefore, if on 1/1/2023 there are 50 registered users in the system, a "reserve" of 5000 tokens is created. On 31/1/2023, the 4000 tokens (80%) are distributed to users according to the evaluation score they have accumulated (see point 5), rounded to the nearest whole number.

5) Evaluation score: offers recommended by users are checked in two ways:

a) based on the price history of the item.

i. If the price found by the user is 20% less than the latest available average price of the previous day, the user is rewarded with 50 points

ii. If the price found by the user is 20% less than the latest available average price from the previous week, the user is rewarded with 20 points

iii. In any other case the user does not receive points, but is allowed to publish his offer.

iv. Average price means the average of product prices across all stores on the previous day or the average of all product prices across all stores for the previous seven days, respectively.

b) based on user ratings.

i. For each user who finds the offer useful, which is indicated by a specific user action (eg pressing like or thumbs up) the recommending user receives 5 points.

ii. For each user who does NOT find the offer useful, which is indicated by a specific user action (eg pressing dislike or thumbs down) the recommending user loses 1 point.

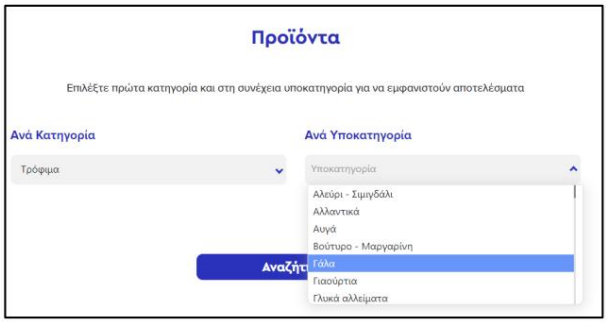
For each user, the system maintains his total score (from the date of his registration in the system), and the score for the current calendar month. The user's current score cannot have a negative value and is reset to zero every month.

6) **Edit profile.** The user can change the username/password and see the detailed history of the offers he may have submitted. He also sees the history of likes/dislikes he has made, his total score and the current month's score, the number of tokens he has received in the previous month and the total tokens he has received since the registration date.

Admin

The Administrator gains access to the system with a fixed computer, through an appropriate username / password mechanism. The administrator is not registered in the system but is created as an entity directly in the relevant database. When entering the system, it has the following capabilities:

1) **Upload, update and delete product data:** The administrator uploads to the system files containing the product data, possible codes, their categories and their prices. The general categories, subcategories and products are drawn from the e-consumer platform (https://e-katanalotis.gov.gr/products/navbar).



For the purposes of the paper, it is sufficient to include at least 4 general categories, 2 subcategories for each general category and 5 products for each subcategory.

The hierarchy of categories, subcategories and products can be given by a separate file in a machine-readable format (XML or JSON).

The prices for each product are given by a separate file in a machine-readable format (XML or JSON). Included in the file are the prices of the products which are considered to be the average prices

in all stores for a specific day. Reuploading a file that includes products that already exist in the NW just updates the prices. If products that already exist in the NW are omitted, this does not lead to their deletion. The administrator can choose to delete all data.

For your convenience, you are given 2 python scripts that upload a) category data, b) product price data, from the e-Consumer platform in JSON format (https://github.com/komis1/e-katanalotis-data). In the repository you will find a JSON file with all the categories and products contained in e-katanalotis, and another JSON file with the price variation of 3 products for a specific week.

2) Upload, update and delete shop data: POIs can be retrieved via nominatim API or Overpass API (shop=supermarket or convenience) based on OpenStreetMap system data. Indicative query for the Patras area: https://overpass turbo.eu/s/1o2L . POIs can be loaded into the system once, either indirectly with the 1st visit to the system and API call, or directly by uploading a file from the administrator. The same file as above or a different one can be used. In any case, the administrator can choose to delete all data.

3) Display Statistics:

a. The administrator can display the number of offers for each day in a graph by selecting a specific year and month. The horizontal axis of the graph includes the relevant dates, from the 1st to the last day of the selected month (be careful, not all months have the same number of days!).

b. The admin can display in a graph the average discount registered for all products, in %, by selecting category and subcategory. If he only selects a (general) category, the average discount for all products in that category is displayed. The "average discount" is calculated as the average price of the differences that all offers have for the products of the specific category or subcategory, from the average price of the product in the previous week (see User, point 5.a.iv) . The horizontal axis of the graph includes the days of the current week, and there is a possibility to navigate backwards (display previous weeks).

4) Leaderboard Display: A ranking list of users based on their total score is displayed. If there are more than 10 users, pagination is done per 10. Also displayed is the number of tokens a user received in the previous month, as well as the total number of tokens received.

5) Display a map with the additional possibility of deleting offers: The administrator is shown a map with all the possibilities that a user has. In addition, the administrator has the option to delete an offer.

Restrictions

1. Groups of 3 (three) people at most.
2. The core of your deliverable must be implemented with technologies they have taught in the course (PHP, JavaScript, Node.JS, HTML).
3. The appearance and functionality of the application is evaluated.

Deliverable

1. Summary report that will include:
   a. The design of the base (Entity-Relationship Diagram for relational bases, its description basic data model for non-relational databases).
   b. The settings you made on your server to properly utilize the capabilities of caches (caches, e.g. setting TTLs for the different types of files). Mention the relevant settings you made in the server configuration, or via .htaccess, or in line definition of response headers and provide some screenshots from the developer tools of the browser you use, to show that the settings you made had an effect. Support the relevant choices regarding the rules you implemented or the values you chose for various parameters, with relevant references to technical articles or scientific literature. 2. The source code and an export of the BD 3. All of the above (1) and (2) are submitted in a compressed file to eclass.

Use of technologies

- You will exclusively use open-source technologies for DB (MySQL, PostgreSQL, MongoDB), maps (the Leaflet library is recommended), charts (the chart.js library is recommended).

- The loading of data from the NW for the presentation of content on the web pages, should to be done using **exclusively** AJAX techniques.

- In any case avoid using **PHP to generate Javascript code.** The following examples are indicative of what NOT to do:

Example 1: using php to complete javascript code <script type="text/
javascript"> var my_var = <?php echo json_encode($my_var); ?>? </script>

Example 1: using php to generate entire blocks of javascript code <?php echo "var test"? echo
"function logvar(f){ console.log(f); "; echo "logvar(test);"

?>?

- Attention to the database: Use appropriate indexes on the tables to speed up the queries. Also make sure that data entry into the database is done with bulk inserts (eg 1000 records together) and not one-by-one records, to speed up the process.