CO3015 Computer Science Project

**Dissertation**

An Android Application of a Canteen Ordering System

Despina Dorotheou (dd186)

Contents

[Abstract 2](#_Toc6415890)

[Introduction 2](#_Toc6415891)

[Survey of Literature/Information Sources 2](#_Toc6415892)

[Requirements 2](#_Toc6415893)

[**User/Android app** 2](#_Toc6415894)

[Login/Signup 2](#_Toc6415895)

[Main 2](#_Toc6415896)

[**Admin/Web app** 3](#_Toc6415897)

[Login 3](#_Toc6415898)

[Main/Orders 3](#_Toc6415899)

[Products 3](#_Toc6415900)

[Offers 4](#_Toc6415901)

[Deals 4](#_Toc6415902)

[Specification/Design/Data-Structures 5](#_Toc6415903)

[Implementation and Testing 5](#_Toc6415904)

[Critical Appraisal 5](#_Toc6415905)

[Conclusion 5](#_Toc6415906)

[Bibliography and Citations 5](#_Toc6415907)

# Abstract

# Introduction

# Survey of Literature/Information Sources

# Requirements

The following are the project’s requirements separated into each application.

## **User/Android app**

### Login/Signup

* The user will be able to login into his/her account.
* The user will be able to create an account through the sign-up page.
* The user will be able to click on “Remember me” and save its email.
* The application will send HTTP request to the web application with the user’s email and password for validation.
* The application will send HTTP request with the information that the user entered in the sign-up page.

### Main

* The application will save any products that are left in basket before the user signs out.
* After login, the application will put in the basket products that the user left from a previous session if there are still in stock.
* The application will send HTTP request to take all the products from the database.
* The application will send HTTP request to take all the deals from the database.
* The application will send HTTP request to take all the offers from the database.
* The main activity will be divided into many fragments.
* The user will be able to click on a category and see all the relevant products.
* The user will be able to see the active categorical deals.
* The user will be able to see all the active offer and add the whole offer in the basket with one click.
* The user will be able to see all the products in a scroll down list.
* The user will be able to see if the products are out of stock.
* The user will be able to click on a product and see more information about it, like description, ingredients etc.
* The user will be able to search through the products by their name.
* The user will be able to search through the products by ingredients.
* The user will be able to set dietary constraints like vegan, vegetarian etc.
* The user will be able to set cost constraints.
* The user will be able to add a product in the basket.
* The user will be able to add a quantity of a product in the basket by increasing and decreasing the quantity of the product.
* The user will be able to remove products for the basket.
* The user will be able to put a product into his/her favourite list.
* The user will be able to remove a product from his/her favourite list.
* The user will be able to see all his/her favourite products in a section in the account page.
* Every time the user adds or removes a product, the application sends a HTTP request to update the user’s data in the database.
* The user will be able to see his/her previous orders in the order history section.
* The user will be able to see all the products in basket, their quantity, a total of their prices, the applied discount and the final total by clicking on the basket icon.
* The user will be able to pay for his/her order with a debit/credit card.
* The user will be able to select the pay by cash option and pay at the collection point.
* After checkout, the user will be able to add/remove items during the first 5 minutes.
* After checkout, the user will be able to delete the order during the first 5 minutes.
* If the user is editing his/her order after checkout, the admin will be notified by a HTTP request.
* After checkout, the user will be able to see his order status.
* The application will send a HTTP request every 10 seconds to check the status of the order until the order is collected.
* After checkout, if the order is ready for collection, the user will be notified by a notification with his/her order number.
* After checkout, if the order is collected, the user will be notified by a notification.
* The user will not be able to modify his/her order if the order is ready, even if the order is ready before the first 5 minutes.

## **Admin/Web app**

### Login

* The admin won’t have to sign up, his/her account will be given.
* Only admins will be able to login into the web application.

### Main/Orders

* The admin will be able to see a table of all the orders in the main page that are not collected.
* If an order’s status is pending, the admin will be able to click on a “ready” button and notified the user that his/her order is ready for collection.
* If an order’s status is ready for collection, the admin will be able to click on a “collected” button and notified the user.
* From the main page the admin can go and see the products, the offers and the deals.
* The main page will reload every 5 seconds, so the table with the orders is up to date.

### Products

* The admin will be able to see all the products in a table.
* The admin will be able to add new products.
* The admin will be able to delete a product.
* The admin will be able to edit the product’s information.

### Offers

* The admin will be able to see all the offers in a table.
* The admin will be able to add new offer and add the products that are in the offer.
* The admin will be able to delete an offer.
* The admin will be able to edit the offer’s information.

### Deals

* The admin will be able to see all the deals in a table.
* The admin will be able to add new deal and add the categories that are in the deal.
* The admin will be able to delete a deal.
* The admin will be able to edit the deal’s information.

### RESTful

* The application will respond to HTTP request for login with a json object of the user if the user exists and it will respond with “Invalid” if the user does not exist in the database.
* The application will respond to HTTP request for sign up with “ok” if the user account was created successfully and it will respond with “Already existing user” if a user with that email exists in the database.
* The application will take the necessary information from the HTTP request for adding or deleting a product of the favourites list of a user and make the necessary changes in the database.
* The application will respond to the HTTP requests for the products/deals/offers with json arrays.
* The application will take the necessary information from the HTTP request for the checkout, create a new order and send it back to the android application as a json object. (same procedure for pay by card or cash)
* The application will take the necessary information from the HTTP request for deleting an order and make the necessary changes in the database.

# Specification/Design/Data-Structures

# Implementation and Testing

# Critical Appraisal

# Conclusion

# Bibliography and Citations