School of Electronics and Computer Science

INFO6005 - Application Development Report

OptiCaff

By Adam Costello, Michael Elkins, Jonathan Harrison, Samantha Kanza, Pratik Patel & Craig Saunders

Abstract

Contents

1 Introduction

OptiCaff is the new innovative way to maximise your productivity during university hours! By analysing your caffeine content in addition to your weekly timetable, OptiCaff can work out the optimum time for your next caffeine consumption. However, this is not all it can do. Not only will it notify you when its time for you to consume your next caffeinated beverage, it will tell you what type will be most effective (tea, coffee, energy drinks) and then point you to the nearest place on campus that sells that product.

1.1 Project Aims / Goals

2 Background Research

2.1 Mobile Platform Research

2.2 Coffee / Caffeine Research

Given the vast range of different cafffeinated products and the limited time to produce a prototype application, it was decided that the products displayed by OptiCaff would be grouped into four different types of drink, and each type would be allocated an average caffeine content. Below is a table showing these totals, which were got from these sources [1] [2] [3].

Drink Category	**Average Caffeine Content (mg)**
Tea	40
Coffee	54
Energy Drinks	80
Soft Drinks	34.5

2.3 Application Research

2.4 Monetisation Research

2.5 Idea Conception

3 Analysis and Specification

3.1 Background Research Analysis

3.2 Application Research Analysis

3.3 SWOT Analysis

3.4 Specification

3.5 Scope

4 Design

4.1 Interface Design

4.1.1 Home Screen

4.1.2 Details Screen

4.1.3 HCI Justification

4.2 Application Interaction & Customisation

4.2.1 Application Walkthrough

4.2.2 Application Customisation

5 Implementation

5.1 Project Management

5.1.1 Version Control

There were several version control mechanisms we could have used. Git and SVN were both considered, with potential to store the code on UGForge or Googlecode or Github etc. In the end the group chose to use SVN Googlecode. SVN was chosen primarily due to the fact that Eclipse was being used as the main development environment, and there is an SVN plugin for Eclipse that makes the subversioning process easier to integrate with the development process. Googlecode was chosen because the application that is being developed is for Android, and therefore it was felt that Android's creators Google would be the most sensible place to store the application.

5.1.2 Group Methodologies

The group decided to implement several agile-based technologies to aid with the application development process. The development areas have been broken down into individual tasks (stories) to simplify the process. An iterative development process was also adopted, with the base of the application built first, followed by a gradual development of its features.

Pair Programming was also an agile technique that the group used. The application development was split into three areas: Database development and SPARQL querying, the map interface and GPS positioning / directions, and the overall user interface. Each section had two group members assigned to it, and each pair worked together to share their skillset and therefore produce a superior result.

5.2 Backend implementation

5.3 UI implementation

5.4 Maps implementation

5.5 Tools & Techniques

5.5.1 Data Source

OptiCaff used the Open Data Service from the University of Southampton [4] to retrieve the relevant information for the application. This service provides open linked data about some of the administrative information regarding the university. It also provides a SPARQL Endpoint [5] (a service which facilitates users querying a knowledge base using the SPARQL query language) [6]. OptiCaff utilises this with a few specialist queries, and combined with user preferences can provide the user with a wide selection of caffeine choices around campus.

5.5.2 Platform

OptiCaff 's purpose lent itself heavily to being a mobile application, and as such that sparked the debate of what platform it should be developed for. It was decided that for the prototype application, Android [7] would be the simpler option for the following reasons: Android can be developed on any platform [8], which is useful as the group uses a combination of OSX, Linux and Windows. Android uses Java, which all members of the group have had significant experience with, and given the timescale in which to complete the application, in addition to other commitments it seemed the most viable option.

5.5.3 Development Tools

Given that OptiCaff was to be an Android application. The development tools needed to facilitate its production were the Android SDK tools [9]. The recommended development environment suggested by Android was to use Eclipse [10] with the ADT (Android Developer Tools) plugin [8]. Given the groups overall familiarity with Eclipse and it's additional useful plugins for version control (see section 5.1.1) this was used as our IDE of choice.

5.6 Implementation Problems

6 Evaluation

6.1 Future Work

6.2 Conclusions

References

[1] http://www.energyfiend.com/caffeine-content/coca-cola-classic.

[2] http://www.food.gov.uk/science/surveillance/fsis2004branch/fsis5304.

[3] http://www.livestrong.com/article/296806-energy-drink-nutritional-values/.

[4] http://data.southampton.ac.uk/.

[5] http://sparql.data.southampton.ac.uk/.

[6] http://semanticweb.org/wiki/sparql_endpoint.

[7] http://developer.android.com/index.html.

[8] http://developer.android.com/sdk/installing.html.

[9] http://developer.android.com/sdk/index.html.

[10] http://www.eclipse.org/.

Appendices