Sofware Engineering: System Requirements Specifications

Due on Monday, August 28, 2017

Daniel COMS3002

Group 4

Contents

Introduction

Purpose

Overall Description

Product perspective

The product being represented in this document is a Fast Food Ordering System, a first of it's kind. The software will take the form of a three-tier client-server architecture, with it's primary aim being to give consumers the flexibility of ordering from any restaurant or fast-food outlet. The software will save consumers the time and effort of having to wait in long queues to order their food. The orders are to be made and paid for one online, with the option of choosing to eat at the restaurant or simply to collect your food to eat in the comfort of your own home.

Product functions

- Enable the user to log on to the system so that orders can be tracked
- Make use of geo-location services to give the user recommendations on places to eat based on a 20km radius
- Have a user friendly menu to avoid confusion when ordering food
- When an order has been made, give a the user a period of 10 minutes to make changes to the order
- Make online payment possible via credit/debit card
- Make online payment secure
- Make use of geo-location services to give user directions to the restaurant
- Keep track of all the orders coming through using numbers

User Classes and Characteristics

From the consumer side, anyone with access to a smart-phone and a working bank account is a potential user of the software. To narrow it down a bit, it will be well suited for the working class, as they have long working hours and they are often stuck in traffic. Thus this software will be perfect for them, i.e they can order whilst they are stuck in traffic, and by the time they get to the food outlet their order will be ready to take home.

Looking at the service providers (restaurants and fast food outlets), the system will be used by trained staff members. Training which is to be received upon installation of the software.

Operating Environment

This software will be designed to run in an internet browser, as a result it will be accessible to most operating systems. It will be developed in conjunction with the Google API to provide most of it's geo-location services. PostgreSQL will be the open source database management system that will be in place to implement all relationships that exist between datasets.

Design and Implementation Constraints

Ehhh Leaders, can I please be given a hand here!!!!!!

Overall Description continued on next page. . .

User Documentation

- Consumers will be given a comprehensive user-manual
- Services providers will also be given their own user-manual which will also double as training manual.

External Interface Requirements

User Interfaces

The system may later be adapted to work as a desktop website but currently we are only focusing on the mobile versions, iOS, Android as well as Windows mobile. That being said, the main feel of the app will be a modern crossplatform design so tiles and buttons will be of AngularJS, CSS3 as well as Ionic.
The first page is a Welcome screen then a Home screen. Options that stay available on each screen is an option icon with:

- Preferences
- Help
- Feedback
- About
- Logout

The Footer of the App will have 3 fixed buttons (more like Instagram). The buttons shall respecitively be:

- Orders
- Home
- (Idle)

Hardware Interfaces

Since by design the App is crossplaform the constraints here needed to be a bit more flexible, however GPS location as well as mobile identity permissions will be required. The app should run in any Android device of version 6.0 or above, iOSX and Windows 10 mobile.

Software Interfaces

An SQL database management system, preferably PostgreSQL shall be used for this three-tier system. The Google Maps API should be used to make it easier for users to visualise their locations as well as locations of the local restuarents.

Communication Interfaces

The main protocal of communication will be HTTP. So we expect encrypted data to shared as JSON arrays or anything better. Libraries like that of JQuery should be considered when handling form inputs that have been sanitized. Minimize response time at all times so persistant open threads may need to be open on the server-side.

Functional Requirements: System Features

System Feature 1

Non-functional Requirements

Performance Requirements