Sofware Engineering: System Requirements Specifications

Due on Monday, August 28, 2017

Daniel COMS3002

Group 4

Contents

Introduction

Purpose

The purpose of this document is to provide a detailed description of the final Fast Food Management System we intend to implement with SCRUM development life cycle.

Intended Audience and Reading Suggestions

Software development teams, our peers and superiors alike are who this document is addressed to. Now, this document is written in an effort to facilitiate cross-departmental contribution to this project. We open with an overall description to state our intended end product followed by external interface requirements to expand upon our development strategies. System features are included to futher modularize the necessary tasks our project must undertake. Nonfunctional requirements explain our intended performance security goals.

Product Scope

Fast Food Management System brings the people to food and bye-bye to queues in Fast food restaurants, fast food are suppose to be *fast*. Registered fast food restaurants will be alliviated of queues by receiving orders online and providing estimate preparation time. On accessing the site or the mobile app, our clients/customers are able to place orders at their selected establishments. Geo-location information is gathered and fast food restaurants servers registered in our database, will appear in order of proximity from our customers recorded geo-location. Menu presentation of the selected restaurant is followed by a place order interface the customer will use to select restaurent's items. Payment credentials will be secured and stored prior to issuing an order number to our customer which must be presented on collection of their order.

Definitions

Three-tier client-server architecture : System Architecture that will define three logically independent tiers, namely, Presentation, Data management and Domain logic.
Geo-location : An estimation of real world geographical location.
Service providers : Refers to restaurants and onther registered fast food outlets
PostgreSQL : An open source Database management system.
iOS(X) : Operating system used by mobile Apple devices.
Android : Google's Most common open-source operating system used by mobile devices.
Windows 10 mobile : Windows operating system ported for mobile device functionality.
AngularJS : A JavaScript framework that allows a dynamic page refresh.
CSS3 : A framework used to format webpage layout.
Ionic : A modern HTML5, CSS framework that supports mobile application development.
JSON : (JavaScript Object Notation) Is used to transmit data between server and web application.
JQuery : JavaScript framework that allows manipulation of a web page to enble user interaction.

References

- van Vliet, H, (2007), Software Engineering Principles and Practice,Wiley.

Overall Description

Product perspective

The product being represented in this document is really 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 as well as sending meals to friends. 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 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.

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 within a 20km radius
- Have a user friendly menu to avoid confusion when making orders
- When an order has been made, give a the user a period of 2 minutes to make changes to the order
- Make online payment possible via credit/debit card
- Make the 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 smartphone 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. Quick training is to be received upon installation of the software.

Operating Environment

This software is designed to run on the popular smartphones as well internet browser later on, as a result it will be accessible to almost all operating systems. It shall 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

High accuracy GPS is very crucial for the system to work at it's best. Also the app should not use more than 128MB of the RAM and take minimum CPU time as that may hamper smooth operation of the entire smartphone.

Overall Description continued on next page...

User Documentation

- Consumers will be given a 1-page comprehensive graphical 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. At any point all errors and technical failures must be handle for by the system and then only notify the user.
The first page is a Welcome screen then a Home screen. Options that stay available on each screen page 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 - To show pending orders and well as approved orders
- Home - To go to the Home screen
- (Idle) - This button on the footer doesn't have a function as of yet

Hardware Interfaces

Since by design the app is crossplatform 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 on 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 restuarants.

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

Pages and permissions

- Restaurant locator

- Identify restuarant name
- Calculate distance to restaurant
- Connect to restaurant server
- Load admin/customer interface

- Admin

- Can view and change the restaurant menu
- Order coordination to ensure effiecient meal preparation for shortest waiting times
- Can view user accounts and see accumulated reciepts and log

- Customer

- Places an order online
- Selects preferred restaurant and places order
- Payment credentials are stored and protected
- Upon successful placement of order, a unique reference number is given to the customer. The number shall be used to access the customers payment credentials
- Customer selects preparation method, eat-in/collect. Their order is then verified and at this point the customer may change their order
- Order is verified and stored in archive. Approximate preperation time and map route(address) is displayed

- Customer locator

- Identify customers location
- Retrieve proxity information from restaurant server and display restaurant location to customer and route.

Logistic management

- User accounts

- Users can register an account that will need their email and password in this way customers can store their receipts with us

- Menu display

- Restaurant menus need to be quite presantable
- The order placing interface will comprise of drop box item selections and view of accumulated cost

- Unique reference number

Non-functional Requirements

Performance Requirements

Search feature :

- The search feature should be easy for the user to find
- Results from any search should be easy to use

GPS :

- The results displayed in the map view should be user friendly and easy to understand
- Selecting a pin on the map should only take one click
- Retreving customer's location should'nt take too long
- Getting the nearest restuarant also shound'nt take too long
- Preferable timmings

others :

- Dropdown menus should be identified and used easily
- Submitting an order should take as lil time as possible
- Admin (restaurant) should recieve an order as soon as the delay time had elapsed
- If the system loses the connection to the Internet or to the GPS device or the system gets some strange input, the user should be informed

Safety and Security Requirements

- Communication between the system and server should be secured. Messages for log-in communication should be encrypted so that other would not get any infomation from them.
- Admin Account should be secured,the Resturant owner should not be allowed to log in (for a certain period of time) after three times of failed log-in attempts. And this must be reported to the developers to check if the system is being attacked or it was just a mistake by the restaurant owner.
- Since the system require users to pay before an order is made, Customers' sensative information should be heavily protected.
- Orders should'nt be mixed up,make sure that the order made goes to the correct restaurant, so to avoid this orders may be stored with respect to restuarants. If a user chooses to sign up, their email adress/cellphone no should be protected.

Software Quality Attributes

Maintainability : The code should be written in a way that allows extension of functions or implementation of new function, for upgrading to be easy.
Portability and flexibility : The system should be accessible from any well known browser (e.g. Google Chrome, Mozilla) other than the app.
Availability : The web should be reachable whenever it is needed, and anyone in South Africa should be able to access it (so long as they are connected).
Reliability : Users should get correct results from any search they make.

Non-functional Requirements continued on next page...

Business Rules

Admin :

- Admin can view orders made by customers
- Can change menu of their restaurant
- They can also give feed back

Customers :

- Customers can create an account
- Customers can make an order, they can cancel an order (before the delay time have not elapsed)
- They can give feedback
- They can also specify a restaurant they want to place on order to