**EDM1142 – LAB 4**

**Name of participants Registration Number**

MBANGA THEODORE ESONG 20210276

ORUH NDJOMO BRYAN CLIFFORD 20210005

**Specialty : Software Engineering**

**Course : Introduction to Software Engineering**

**SRS SHEET FOR THE DEVELOPMENT OF A RESTAURANT FOOD DELIVERY APP**

**OUTLINE**

1. Introduction
2. Purpose
3. Intended Audience
4. Intended Use
5. Scope
6. Definitions
7. Overall Description
8. User Needs
9. Assumptions and Dependencies
10. System Features and Requirements

10.1 Functional Requirements

10.2 External Interface Requirements

10.3 System Features

10.4 Non-functional Requirements

This document details the project plan for the development of a Restaurant food delivery app

It is intended for developers, designers, and testers working on this project as well as project investors. This plan will include a summary of:

* how the system will function
* the scope of the project from the development viewpoint
* the technology used to develop the project, and
* the metrics used to determine the project’s progress
* Overall Description

Nowadays there are a lot of local restaurants producing good quality meals but lack the ability to make this food reach a larger public and also, people sometimes feel tired to leave their homes or work places to move to these restaurants.

So, this app will help address this problem

CUSTOMERS :

The customers will be the different restaurants and the public

FUNCTIONALITY

* Users should be able to search a restaurant of their choise and select a meal
* Users should be able to communicate with the individual restaurants
* Users should be able to purchase online
* Restaurants should be able to sell and deliver food using this app

PLATFORM

The application will be developed to run on both android and IOS devices, also on PCs

DEVELOPMENT RESPONSIBILITIES

The developers on the project team will be responsible for writing all the code for the application, developing the database, and managing releases.

USER CLASS AND CHARACTERISTICS

There will be a class of users called premium user that will have permissions to access all functionality of the app, including:

* Commanding more than 6 meals at a time
* Discounts
* Unlimited different restaurant availability

Standard users will have access to all functionality of the app except those listed above.

**SYSTEM FEATURES**

FUNCTIONAL REQUIREMENTS

* Users should be able to create an account on the app server
* Users should be able to pay their commands via : **Mastercard, Visacard etc**
* Users should be able to leave comments and feedbacks about the restaurants they’ve commanded from
* The application will be used online
* Users should be able to retract a command depending on the restaurants guidelines
* Users should be able to ask for help from customers service through text or calls
* Different restaurant should also have their accounts on the server

HARDWARE INTERFACES

* Both Mac and Windows operating systems through the app
* iPhone
* Android

SOFTWARE INTERFACES

* Front-end software: React Native
* Back-end software: .NET Core
* Database software: MySQL
* LDAP connection For Authentication

PERFOMANCE REQUIREMENTS

* The application should load and be usable within 3 seconds
* The application should update the interface on interaction within 2 seconds
* The database should be normalized to prevent redundant data and improve performance
* The database should be distributed to prevent outages

SAFETY REQUIREMENTS

* Databases should use sharding to be redundant to prevent loss of data.
* Backups of the databases should be done hourly and be kept for one week.

SECURITY REQUIREMENTS

* Informations about customers credit cards should not be shared publicly.
* Databases should be behind a firewall.

SOFTWARE QUALITY AND ATTRIBUTES

* Availability: This application will be available 99.99% of the time since customers may have different needs at different times
* Correctness: The application should never allow anyone’s personal details be read by any other person
* Maintainability: The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime.
* Usability: The interface should be easy to learn. A tutorial on how to use the application should be included and should allow users to accomplish their goals without errors.