**Software Requirements Specification**

February 28, 2020

Team 4 (Lina McDermott, Caroline Sigl, Joe Kane, Mary Lasater, Liam Carriker, Mariem Bchir)

**Introduction:**

This report describes Team 4’s Software Requirements for its project to develop a mobile application to connect teachers and schools in the MENA region.

1. Introduction

1.1. Purpose

A number of schools in the African Continent and MENA region suffer a low teacher retention rate and still use old non-digital recruitment channels to replace and recruit new educators. Our purpose is to digitize the recruitment and search process for both teachers and schools by providing an online platform (mobile application) hosting the whole process.

1.2. Scope

This document covers the requirements for the entirety of the first version (pilot) of a completely new app.

1.3. Definitions, Acronyms, and Abbreviations

SRS - Software Requirements Specification

TCP - Transmission Control Protocol

IP - Internet Protocol

MENA – Middle East and Northern Africa

1.4. References

JAVA community Process

Android standards

1.5. Overview

This application aims to connect schools to teachers seeking employment. Both schools and teachers would have their user accounts where they input their respective preferences. Our application will provide a search and filtering engine that will allow both parties to find their best fit and it will potentially provide a matching algorithm that automatically pairs teachers and schools and provides them with suggestions.

2. Overall Description

2.1. Product Perspective

This project will be compatible with android platforms primarily to be used on mobile phones. This will make the project accessible to the users. The product will allow teachers and schools to search for and find each other. This will help bridge communication gaps and allow educators to find available positions at schools that they may not have been aware of otherwise.

2.2. Product Functions

2.2.1. Creates two types of user account

2.2.2. Stores user accounts in database

2.2.3. Provides a search and customized filtering mechanisms for both users

2.2.4. Saves the user preferences to be accessed when requested from the user

2.2.5. Allows Teacher users to visit other School user’s profiles

2.2.6. Allows School users to visit other teachers’ users profiles

2.2.7. Does not allow School users to access other school users’ profiles

2.2.8. Does not allow Teacher users to access other school teachers’ profiles

2.2.9 Provides suggestions to both users based on a matching algorithm

2.3. User Characteristics

Our app will be catered towards both teachers and school hiring staff. For this reason, we want to create an intuitive and easy-to-use app. Also, since we are aware of the higher use of android phones in our targeted users area, we are making an android app.

2.4. General Constraints

The product will be constrained by network connectivity and internet access. This product will only be available on Android devices, and only verified schools will be allowed to use the app’s school user function.

2.5. Assumptions and Dependencies

We are assuming that teachers would trust/be open to an online matching algorithm (considering the lack of digitization within the African continent). There is a potential need for a cultural and behavioral shift. We assume that teachers and schools are familiar with the new online interfaces (generational gap). Both users will have different ways of experiencing the application (hence providing different functionalities). We also assume that the simplicity of a mobile application features are enough to provide the complexity of functions our audience needs.

3. Detailed Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

3.1.1.1. The system will have an Android user interface, using XML layouts for

each screen, and various Android components for all interfaces.

3.1.1.2. The interfaces include: a login screen, a search screen, a search

results screen, and a menu available from the top left corner of the interface.

3.1.2. Hardware Interfaces

3.1.2.1. The system assumes that the user has access to an internet connection and an Android smartphone to be able to download and use the application.

3.1.3. Software Interfaces

3.1.3.1. The application will interface with a SQL database.

3.1.4. Communication Interfaces

3.1.4.1. The application shall use a TCP protocol to ensure the delivery of packets to avoid any delays or losses.

3.2. Functional Requirements

3.2.1. Teacher User

3.2.1.1. The project shall allow users to create an account and fill in relevant user data, including credentials, school subject, etc.

3.2.1.1.1.  The system shall accept a username, email, and password of at least 12 characters to create a teacher user account.

3.2.1.1.2. The user shall be able to input their name, residence, certification type, years of experience, location preference, teaching subject(s), degrees obtained, gender, photo. These inputs shall constitute their user profile.

3.2.1.1.2. The application shall save their user profile to the user database.

3.2.1.2.The application shall allow users to access a homepage dashboard

3.2.1.2.1Dashboard should display their profile, preferences and a filtering mechanism to go through the database.

3.2.1.3.The application shall allow users to search and filter through the database of schools/available jobs

3.2.1.3.1 The application shall allow the users to search for jobs by key words.

3.2.1.3.2 The application shall allow the users to filter search results by location, size, available positions, subject, working hours and average salary.

3.2.1.3.3 A search that yields no results shall indicate this fact to the user

3.2.1.4. The application shall allow the user to edit the information on their profiles.

3.2.1.4.1. The user shall have the ability to edit all aspects of their account that they created during 3.2.1.1.

3.2.1.5. The application may allow users to save jobs/schools to a list that they will be able to access later.

3.2.2. School User

3.2.2.1.The project shall allow users to create an account.

3.2.2.1.1. The system shall accept a username, email, and password of at least 12 characters to create a school user account.

3.2.2.1.2. The system shall handle invalid input for the account creation by not creating an account

3.2.2.1.2. The system shall allow the schools to enter in their location, the level(s) of education they provide, size (number of students and number of teachers), and average salary to fill in their profiles.

3.2.2.2. This application shall allow the verification of the school users’ identity.

3.2.2.2.1. The system shall verify the schools by prompting the user to upload a photo of their government license documents, which will be verified manually.

3.2.2.2.2. Invalid documents shall result in the school being denied access to the application.

3.2.2.3. Search and filter through teacher users

3.2.2.4.This application shall allow the school user to create and edit job listings.

3.2.2.4.1. This job listing creation shall be composed of a job title, the school requesting, the school’s location, and the requirements for the job itself. These jobs can be tagged appropriately for search and filter purposes by the school user.

3.2.2.4.2. Editing job listing shall allow a school to change any of the above job listing attributes.

3.3. Performance Requirements

3.3.1 Device Performance

3.3.1.1. The application shall not take more than 1.5 seconds to launch.

3.3.1.2. The system shall use a maximum of 200.00 mAh of power

3.3.1.3. The system shall not use more than 100 MB after installation, and may use up to 200MB of RAM when active.

3.3.1.4. The system shall run smoothly on Android 10 or later.

3.3.1.5. The system shall run properly with no slowing down of features while other applications are running in parallel.

3.3.1.6. The system shall preserve the state of user input and usage state when running in the background to protect the data from getting lost.

3.3.2 Server/API Performance

3.3.2.1. The system shall efficiently handle the data transferred during its communication with the server through API.

3.3.2.2. The system shall have a small number of API calls by avoiding generating several calls for the same functionality.

3.3.2.3. The system shall return a server access error in the case of lack of network connectivity.

3.4. Design Constraints

The application has no design constraints.

3.5. Attributes

The application has no additional attributes.

3.6. Other Requirements

The application must respect and abide by each country’s laws. Additionally, the system must take into account user privacy and safety.