|  |
| --- |
| **Project: Admission Office Software** |
| **Technical requirements** |
|  |
| This project aims at providing PKFokam Institute of Excellence with a Software to support the management of the academic maters of the University |
|  |
| **Author: Mr. Herman Mekontso** |
| **Lecturer CS/IT – PKFokam Institute of Excellence, CS Researcher** |
|  |



Table of contents

[1 – Project overview 3](#_Toc92193161)

[2. Current status of the project 3](#_Toc92193162)

[3. Future work 3](#_Toc92193163)

[4. Technical requirements 3](#_Toc92193164)

[4.1. Architecture 3](#_Toc92193165)

[4.2. Operating System 4](#_Toc92193166)

[4.3. Hardware specifications 4](#_Toc92193167)

# 1 – Project overview

The software we intent to design is an admission office manager that will assist admission officers in their daily tasks. In effect, having noticed the number and the diversity of responsibilities entrusted to these people, they have to be assisted by automated tools in order to be efficient. Daily activities of admission officers can be grouped in 6 modules plus a settings module:

* The management of students;
* The management of the staff (lecturers and administration staff);
* The management of grades, results, and transcript generation;
* The management of attendances;
* The management of courses;
* The Management of programs.

# 2. Current status of the project

* The software has been developed and installed. The first version was installed in august 2018, the admission officer has been trained to use the software and a user manual has been provided;
* The exploitation is ongoing and bugs fixing and enhancements are done gradually.

# 3. Future work

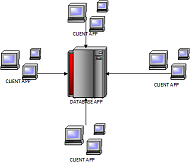
Some operations are still to be implemented to improve the current version. These operations include:

* The implementation of a caching system. The outcome will be a proxy which reduces the number of access to the database, enhancing the execution time of the application;
* The implementation of a user management module, to enable adding new user profiles, editing user profiles and deleting some existing user profiles.
* The implementation of a small data source configuration module which will enable the current version of the software to be fully 2-tier.

# 4. Technical requirements

## 4.1. Architecture

The software is a 2-tier application with a RIA (Rich Internet Application) developed with JavaFX on the client side and a MySQL 5.5 database on the server side.



The database can be deployed on a remote server (machine) and the client installed on any computer with a Java Virtual Machine, version 8 or later.

## 4.2. Operating System

There is no restriction on the OS on which the application can be deployed. The application works on any device with a Runtime compatible with java 8.

## 4.3. Hardware specifications

RAM:

* 2Gb for the client App is recommended;
* 4Gb for the database server.

# 5. Deployment procedure

As previously said, the application is a multi-users 2-tier with a database deployed on a tier and the application itself on another tier.

## 5.1. On server side

The actual version only need a database server. In this project, we DBMS used is MySQL 5.5. So make sure the server has this version or later installed. Below is the configuration to be set to prepare the importation of the data:

**User**: root

**Password**: herman24

**Database name**: schoolmanagement2

**Import the database data**

* This software will be delivered with data stored in a database file named **schoolmanagement.sql** (another name might be used) to be imported.
* Make sure mysql path is added to the path environment variable and then:
  + From the user **root/herman24**, create a database named **schoolmanagement2** if it does not exist;
  + Import base data with the following command from the command line:

**mysql –uroot –p schoolmanagement2<schoolmanagement.sql**

## 5.2. On client side

The client needs jre8 or latter for its execution. The client is a compress file to be unzipped in the location where you wish to deploy your application. Unzip it, create a shortcut to the executable file **“School Management”.jar,** and attach to it the icon of the University which can also be found from the unzip file.

The package contains a configuration file named **config.properties** containing the configuration of the remote database. There are two ways to edit the file and modify the configuration:

1. Edit it with a plain file editor and change the properties accordingly;
2. Launch the application and connect using the following user: **admin/admin2022** then the following window will be displayed and you can modify the configuration.

