

SOFTWARE REQUIREMENT SPECIFICATION

iCAME STUDENT ATTENDANCE CHECKING SYSTEM

AUTHOR: JEFFREY ASANTE WIREDU (SYSTEM ANALYST)

ORGANIZATION: APPLEPI INC.

PREPARED FOR: COE 356- SOFTWARE ENGINEERING

INSTRUCTOR: MR JEPHTAH YANKEY

KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **DESCRIPTION** | **AUTHOR** | **COMMENTS** |
| 13th March, 2019 | Version 1.0 | Jeffrey K.O Asante Wiredu | Basic Version of the application |
| 27th May, 2019 | Version 2.0 | Jeffrey K.O Asante Wiredu | Updated version of the application |
|  |  |  |  |

**DOCUMENT APPROVAL**

This document has been approved by:

|  |  |  |
| --- | --- | --- |
| **NAME** | **TITLE** | **DATE** |
| Jeffrey K.O Asante Wiredu | System Analyst | 30th May, 2019 |
| Lois Frempomaa Frempong | Project Manager | 30th May, 2019 |
| Steve Ababio Ansah | Chief Backend Programmer | 30th May, 2019 |
| Richard Quaicoe | Chief Frontend Programmer | 30th May, 2019 |

Table of Contents

[1. INTRODUCTION 4](#_Toc10550936)

[1.2 SCOPE 4](#_Toc10550937)

[1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 4](#_Toc10550938)

[1.4 REFERENCES 4](#_Toc10550939)

[1.5 OVERVIEW 4](#_Toc10550940)

[2.0 GENERAL DESCRIPTION 5](#_Toc10550941)

[2.1 PRODUCT DESCRIPTION 5](#_Toc10550942)

[2.2 PRODUCT FUNCTIONS 5](#_Toc10550943)

[2.2.1 ADMINISTRATOR 5](#_Toc10550944)

[2.2.2 USER 5](#_Toc10550945)

[2.3 USER CHARACTARISTICS 5](#_Toc10550946)

[2.4 ASSUMPTIONS AND DEPENDENCIES 6](#_Toc10550947)

[3.0 SPECIFIC REQUIREMENT 6](#_Toc10550948)

[3.1 EXTERNAL INTERFACE REQUIREMENT 6](#_Toc10550949)

[3.1.1 USER INTERFACES 6](#_Toc10550950)

[3.1.2 HARDWARE INTERFACES 6](#_Toc10550951)

[3.1.3 SOFTWARE INTERFACES 6](#_Toc10550952)

[3.1.4 COMMUNICATION INTERFACES 7](#_Toc10550953)

[3.2 FUNCTIONAL REQUIREMENT 7](#_Toc10550954)

[3.3 USE CASES 7](#_Toc10550955)

[3.4 NON-FUNCTIONAL REQUIREMENT 9](#_Toc10550956)

[3.4.1 PERFORMANCE 9](#_Toc10550957)

[3.5.2 SECURITY 9](#_Toc10550958)

[3.5.3 MAINTAINABILITY 10](#_Toc10550959)

[3.6 DESIGN CONSTRAINTS 10](#_Toc10550960)

[3.7 OTHER REQUIREMENT 10](#_Toc10550961)

[4 .0 ANALYSIS MODELS 10](#_Toc10550962)

[4.1 DATA FLOW DIAGRAM 10](#_Toc10550963)

[APPENDICES: USER INTERFACE 11](#_Toc10550964)

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK

# INTRODUCTION

Attendance to classes has become an important factor in the modern education field. This system is to help any institution to monitor and provide real-time access to data of how regularly students attend classes. The study findings enable the definition of the project problem statement, its objectives, scope, and advantages of this specific attendance management system.

## SCOPE

This document covers the requirements of the iCAME Attendance Checking System. This software will provide a graphical environment in which the users of the system will be able to register, generate graphical images of a code and retrieve information. The purpose of this is to guide developers and users in selecting a design that will be able to accommodate the full-scale application. The system will capture information of students, lecturers and their respective courses.

# 1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Attendance Based System – ACD

The QABD has to handle records of a number of students and lecturers hence, maintenance was a bit difficult. Though this is an information-based system, some events are to be handled manually and therefore, the is the need to upgrade the system with a computer-based information system.

# 1.4 REFERENCES

Software Engineering A Practitioner’s Approach – Roger S Pressman

# 1.5 OVERVIEW

The purpose of this document is to present a detailed description iCAME ACD. It explains the purposes and features of the software, the interface of the software, how the software operates and the constraints under which it must operate. This document is intended for both end users and developers of the software.

# 2.0 GENERAL DESCRIPTION

# 2.1 PRODUCT DESCRIPTION

The product is a dependent product. It depends on APIs provided by Google in order to generate QR codes. The product will automate various tasks associated with handling student and lecturer details, attendance and better organization of the stored information and thus, helping lecturers track their students and the school as a whole.

# 2.2 PRODUCT FUNCTIONS

This system has two types of accessing modes:

1. Administrator
2. User- 2.1 Lecturer

2.2 User

# 2.2.1 administrator

The database of this software is managed by the administrator. The administrator has the monitor the updating of student details, assigning courses to lecturers and students, ensuring that attendance is taken properly etc.

# user

There are two types of users:

1. **Lecturer:** This user can be added unto the portals to view their schedules, project a code for students to take attendance, view student details etc.
2. **Students:** This user only view and edit their personal details, check courses assigned to them and view their attendance.

# user charactaristics

The software gives access to:

1. **Administrator**: The personnel and college administrator will have administrator rights. They will have the ability to add, delete and modify information stored in the database.
2. **Authorized User**: Teaching staff will have access to only view the data stored in the database and can update a student’s attendance in form of formatted reports.

# assumptions and dependencies

* We assume that all the computers that will use this software will be part of a Land Area Network.
* End users of this software are assumed to have basic computer knowledge i.e. point and click.

# 3.0 specific requirement

# 3.1 external interface requirement

# 3.1.1 user interfaces

* GUI along with mobile application
* Reports are generated as per requirement by lecturers

# 3.1.2 hardware interfaces

|  |  |
| --- | --- |
| Hardware Environment | Intel Core i3 3rd Generation/ Snapdragon 600 |
| System Configuration | RAM-512 MB HDD 80GB(PC)/ RAM-2GB HDD 8GB(Mobile) |
| Operating System | Windows 7/8/8.1 or Android Jellybean |

# 3.1.3 SOFTWARE INTERFACES

|  |  |
| --- | --- |
| FRONT END | JavaScript(React, ReactNative) |
| BACK END | JavaScript(NodeJS), MySQL |

# 3.1.4 communication interfaces

The systems to be used will be part of the college Local Area Network to access the central database.

# 3.2 functional requirement

The QABD involves the following functions:

1. **Code Generation**: The lecturer types in a special code for a QR picture of the special code to appear on a screen for students to scan.
2. **Attendance Management**: This is to quickly produce a single or multiple day attendance bulletin upon request.
3. **QR Scanner**: This is for students. They can be able to use this to scan a QR picture for their attendance to be taken.
4. **Student/Lecturer Registration**: This provides an online registration platform for lecturers and students.

# 3.3 use cases

|  |  |
| --- | --- |
| Use case name | Sign-up |
| Actors | Student |
| Summary description | Student enters details to be stored in the system |
| Pre-condition | Students can access the system via a web browser or an android app |
| Basic path | 1. Students provides their details 2. System saves the students details into the database 3. System registers the new student 4. System then logs the customer into the web app or android app |
| Alternative path | N/A |
| Post condition | System creates an account for the student |

|  |  |
| --- | --- |
| Use case name | Sign up |
| Actors | Lecturer |
| Summary description | Lecturer enters their details to be stored in the system |
| Pre-condition | Lecturer can access the system through a web browser or the android app |
| Basic path | 1. Lecturer provides details 2. System saves the lecturer’s details into the database 3. System registers the lecturer 4. System then logs the customer into the web app or android app |
| Alternative path | N/A |
| Post condition | System creates an account for the lecturer |

|  |  |
| --- | --- |
| Use case name | Lecturer creates a class on the app |
| Actor | Lecturer |
| Summary description | The lecturer creates a class |
| Pre-condition | The lecturer must be registered |
| Basic path | 1. Lecturer signs up 2. System registers lecturer 3. Lecturer has the option of creating classes |
| Alternative path | N/A |
| Post condition | System stores the classes |

|  |  |
| --- | --- |
| Use case name | Display of code |
| Actors | Lecturer |
| Summary description | The lecturer generates the QR code to be displayed |
| Pre-condition | A class must have been created |
| Basic path | 1. Lecturer selects course 2. Lecturer generates code 3. Code is displayed |
| Alternative path | N/A |
| Post condition | System stores the code used |

|  |  |
| --- | --- |
| Use case name | Scanning |
| Actors | Students |
| Summary description | The students scan the code to confirm their attendance |
| Pre-condition | 1. Students must be registered 2. The course must have been registered by the lecturer 3. The lecturer must have generated the code to be scanned |
| Basic path | 1. The students sign in once into the app 2. The courses are displayed or the student can search for a course 3. The lecture generates the QR display 4. Students scan 5. System stores the student’s details in the database |
| Alternative path | N/A |
| Post condition | System stores the record to be retrieved by the lecturer |

# 3.4 non-functional requirement

# 3.4.1 performance

Easy tracking of attendance records and updating can be done. All the requirement relating performance characteristics of the system are specified in the section below. There are two types of requirement.

1. **Static Requirement**: These requirements do not impose any constraints on the execution characteristics of the system. They are:

* **Number of Terminals:**  The software makes use of an underlying database that resides in the server while the front end will be available online and on the administrative and departmental computers as well as lecturers. A mobile application will be made for students.
* **Number of Users:** The number of users may vary based on the department of the class in question.

1. **Dynamic Requirement**: These specify constraints on the execution of the system. They include time response of the system. Since these factors are not applicable to the proposed software, it will be sufficient for if the response time is high and the transactions in the database are carried out quickly and precisely.

# 3.5.2 security

The security requirements deal with primary security. Hence, the database of this software should be handled exclusively by the administrators and authorized users. Only the administrator has the right to assign permissions like creating new accounts and generating password. Only authorized users can access the system with a username and password.

# 3.5.3 maintainability

Backups for the database are available in order to check attendance records.

# 3.6 design constraints

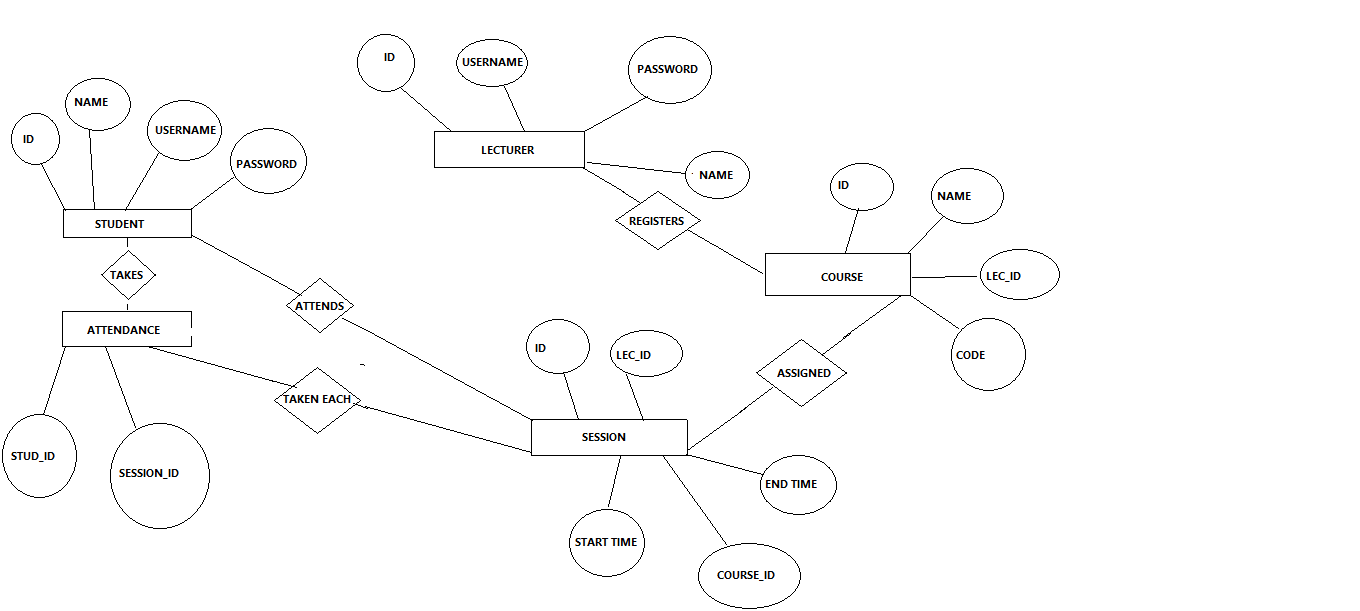
This software provides records and thus, the records need security. The login is a one-login per device system model. Therefore, students will not be able to use their mobile devices to take the attendance for absentees.

# 3.7 other requirement

All data from students and lecturers will be stored in a relational database alongside the attendance records.

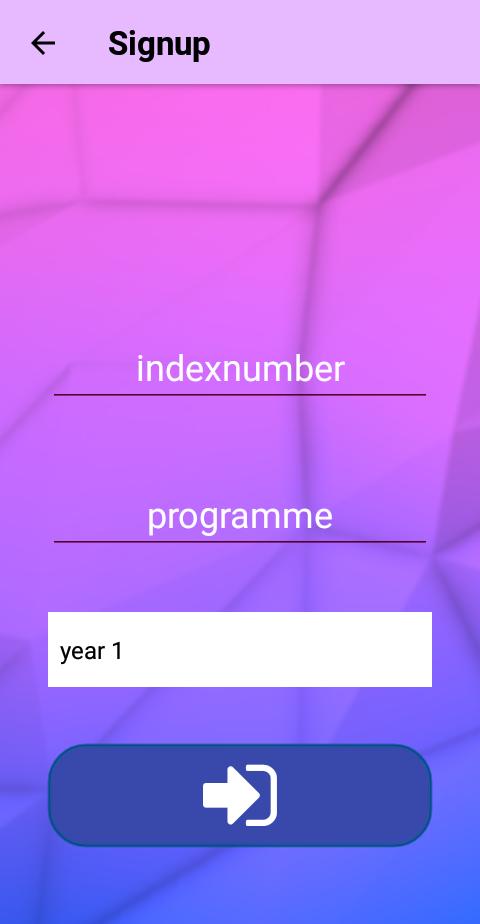
# 4 analysis models

# data flow diagram

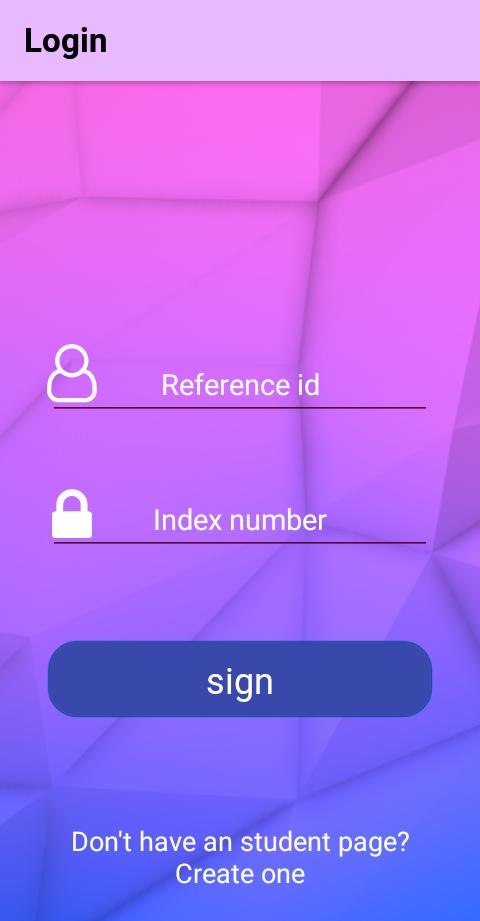


# APPENDICES: USER INTERFACE

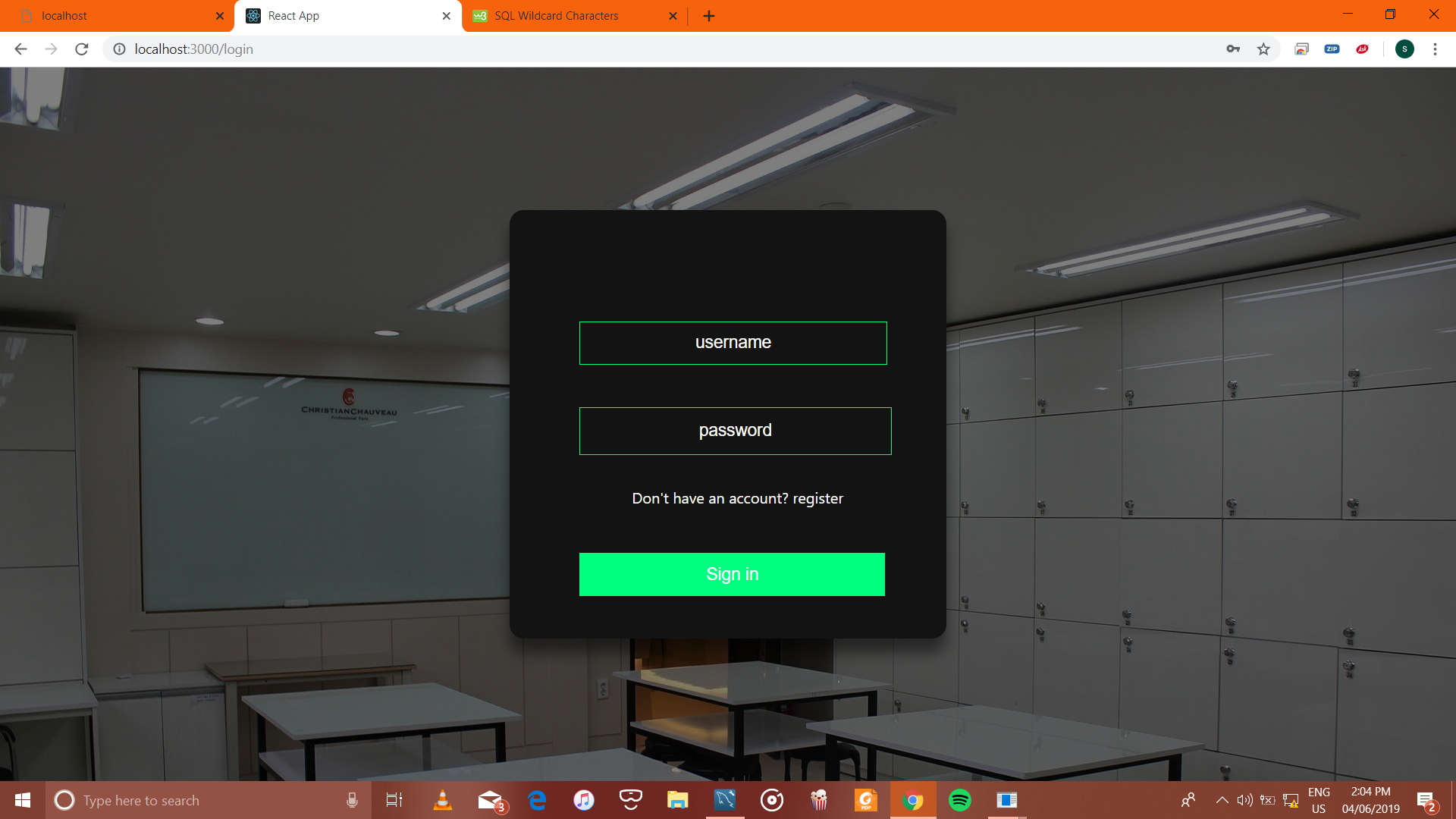
1. **SIGN-UP(MOBILE)**



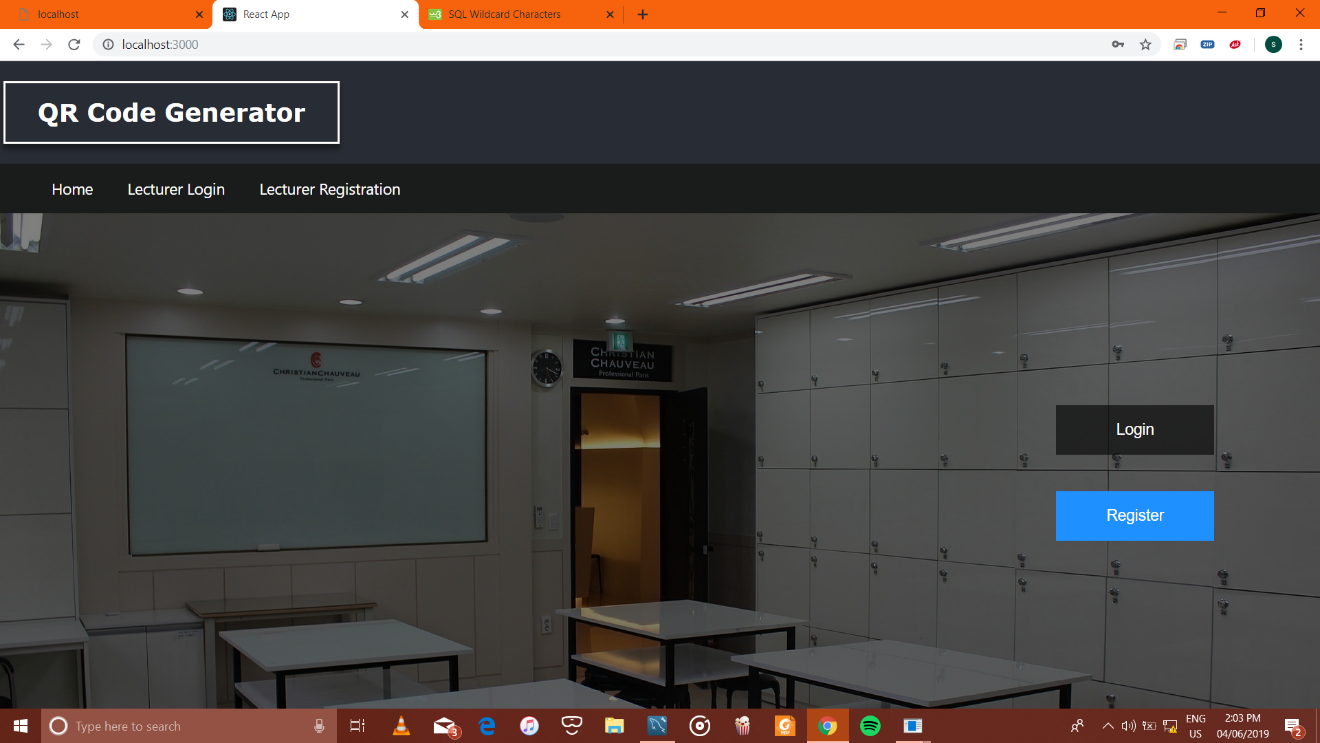
1. **LOGIN(MOBILE)**

****

1. **LOGIN(WEBPAGE)**

****

1. **HOMEPAGE(WEBPAGE)**

****