**CS551 Advanced Software Engineering**

**Second Increment Report**

**Title: UMKC Student E-ID Card**

By: Ampabathina Abhiram (Class ID: 2)

Gadepalli Anusha (Class ID: 19)

Komalla Harsha (Class ID: 38)

Yarragonda Harshi Priya (Class ID: 69)

**I. Introduction**

UMKC Student e-ID card app is an app which allows students to view their ID card online in mobile. Our app can be used by UMKC students to view or verify others ID card. Each student is given a unique QR code to eradicate duplicity. One cannot duplicate the QR code as the app will be verifying the QR code with the UMKC database. The ID card given to the students can be easily duplicated. As technology is rapidly evolving we need to keep the data more secure. By this app we can verify a student identity, whether they belong to a college or not. This is inexpensive and very fast. QR code makes it possible to generate a unique hashed image.

**II. Objectives:**

The main objective of the project is to view and access the student ID card from the mobile phones with the student credentials. Another goal of our application is to verify the student details to check whether he /she belongs to the same university.

In this increment the main features involve the generation of QR Code, view the student details and read the QR Code. Includes generating a database using mongo DB to retrieve data of students.

**III. Import Existing Services/API:**

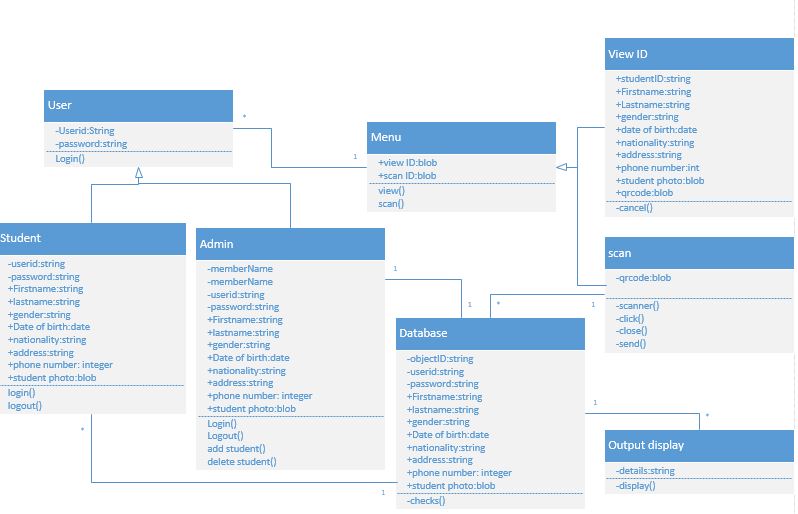
API to create a QR code image : <https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=Example>

API to read the QR code image :<http://api.qrserver.com/v1/read-qr-code/?fileurl=http%3A%2F%2Fapi.qrserver.com%2Fv1%2Fcreate-qr-code%2F%3Fdata%3DHelloWorld>

API to scan QR code using phone gap API: <https://build.phonegap.com/plugins/261>

**IV. Detail design of services:**

Class diagram mainly defines the structures of the classes defined in the system. The system is developed for Student e-ID card. The important terms are User, menu, database and output display. In the User class we will take the student ID and password and perform the login operation and the properties are inherited to student and Admin to add and drop students. This user is associated with Menu where there will be options like view ID, scan ID. This properties are inherited to view the ID and scan the ID. This QR code is scanned and the details are verified with the database and the output is to be displayed whether the student is actually registered or not.



**Sequence Diagram:**

Sequence diagram is a kind of interaction diagram that shows how processes interact with each other in a particular order. In our application initially user logins and his/her account gets validated. Then the user chooses from two options, either to view his/her details or scan another user’s details. After scanning the database is connected through mongo DB and details are displayed.



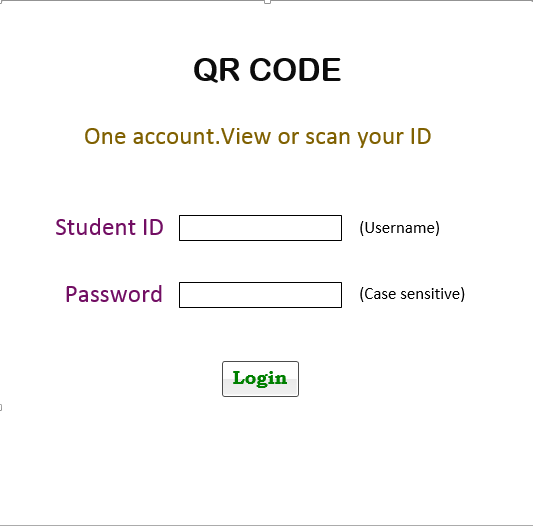
**State Diagram:**

A state diagram defines behavior of a system. The current behavior of the system is dependent on the preceding state of the system. In our application, if the login is successful a menu is displayed else the system is stopped. After login if the user selects to view the details then the details are viewed else a scanner is appeared. The scanner is connected with the database which retrieves the data whether the student is actually registered in the particular college or not.

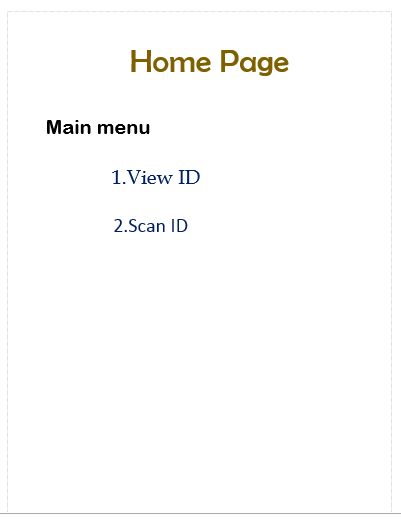


**IV. WireFrames and Mockups:**

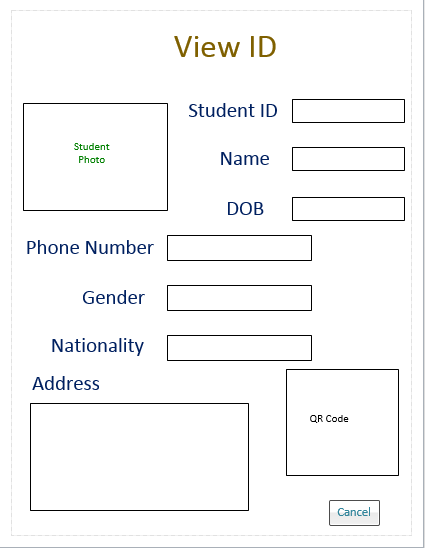
Login page: The unique student ID and password are to be entered to login to the account. On click of the login button, it opens to home page



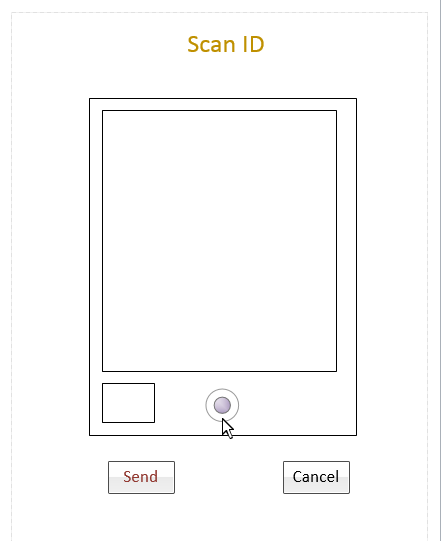
Home page: The home page shows two menu items called view the ID and scan the ID. With view ID option, all the student information can be displayed along with the unique and dynamic QR code whereas with scan ID the QR code can be scanned and verified with the student database information.



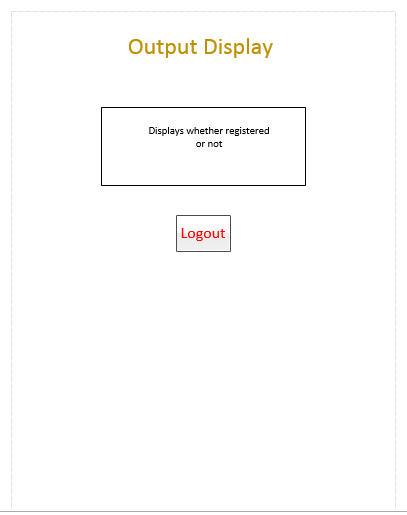
View ID: In the view ID page the student details such as student ID, name, DOB, phone number, gender, nationality, address, student photo and QR code are displayed.



Scan ID: In the scan ID page, the QR code can be scanned and read from the database to verify whether the student is enrolled in the particular college.



Output Display: In this page it displays whether the student is actually enrolled in the college and it also contains a logout option to logout from the account.



**User stories:**

In this increment,

As a system, I need to be able to get the QR code along with the student details so that every student can have a unique code.

As a user, I want to able to view the details of the student like QR-code, name, ID, date of birth and so on so that I can use the QR-code to verify my details in college database.

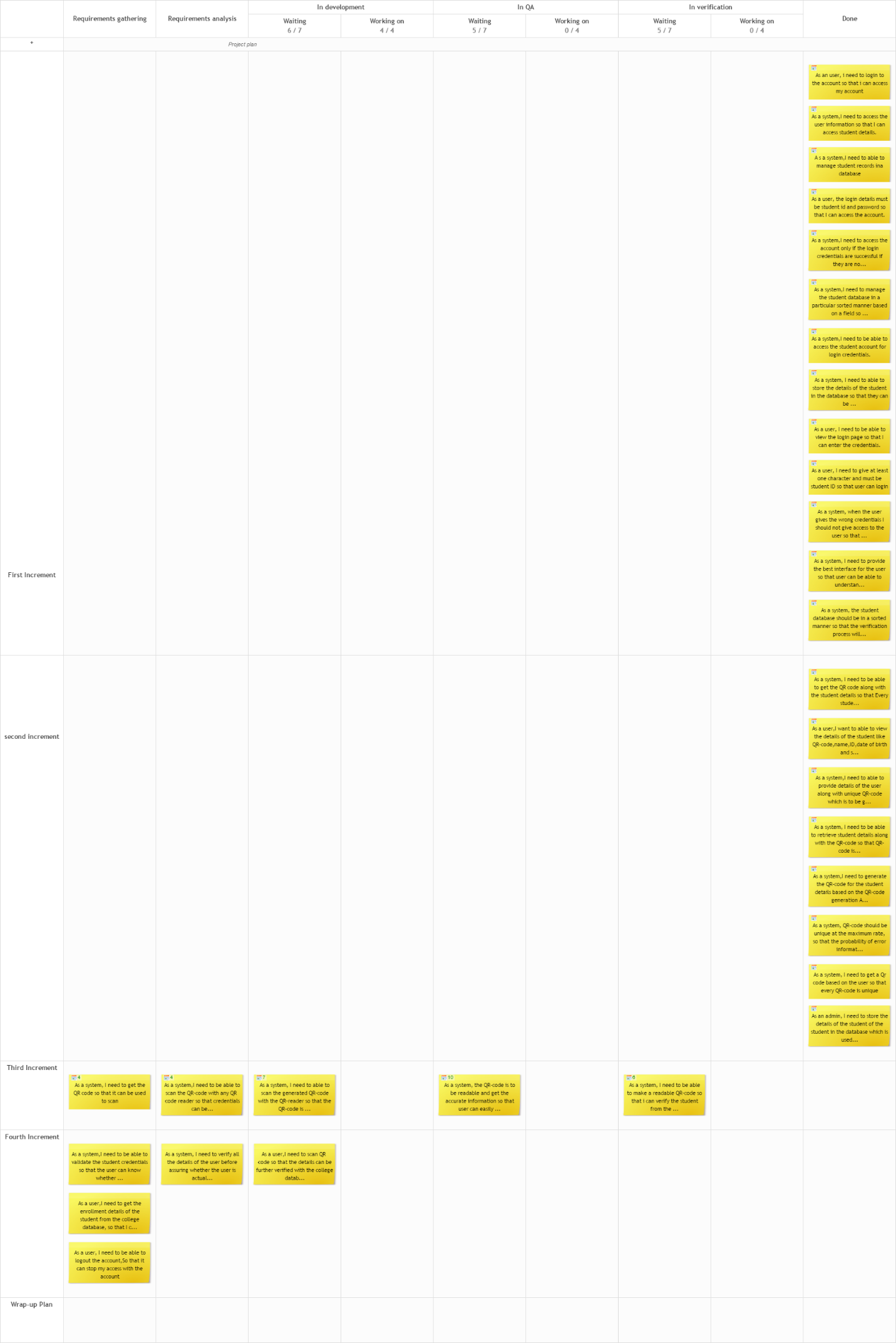
As a system, I need to able to provide details of the user along with unique QR-code which is to be generated based on the user details so that the QR-code will be unique for every user.

As a system, I need to be able to retrieve student details from the database along with the QR-code so that QR-code is scanned.

As a system, I need to generate the QR-code for the student details based on the QR-code generation API so that the QR-code is unique.

As a system, QR-code should be unique at the maximum rate, so that the probability of error information is minimum.

As an admin, I need to store the details of the student of the student in the database which is used to retrieve and verify the student details.

****

**User Interface scenarios:**

* The login button should be placed in the main page.
* The username and password should be placed in the login page.
* The password field should be placed after the username field.
* The main page should be opened after clicking login
* The id should be opened after clicking the “View ID” button in main page.
* The QR code should be scanned after going for the scan ID button in the main page.

**Screenshots:**

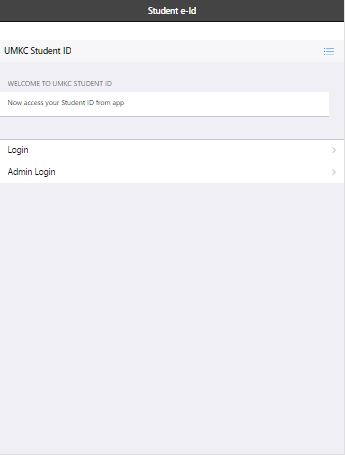
****

Figure Home Page

This page is the first screen that the user sees when using the application.

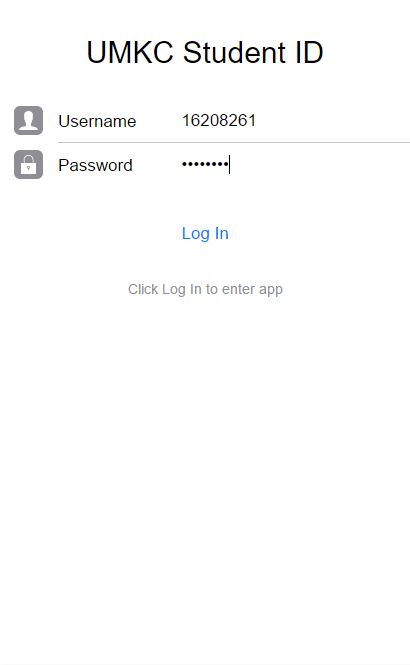
****

Figure Login Page

This page appears when student clicks login button.

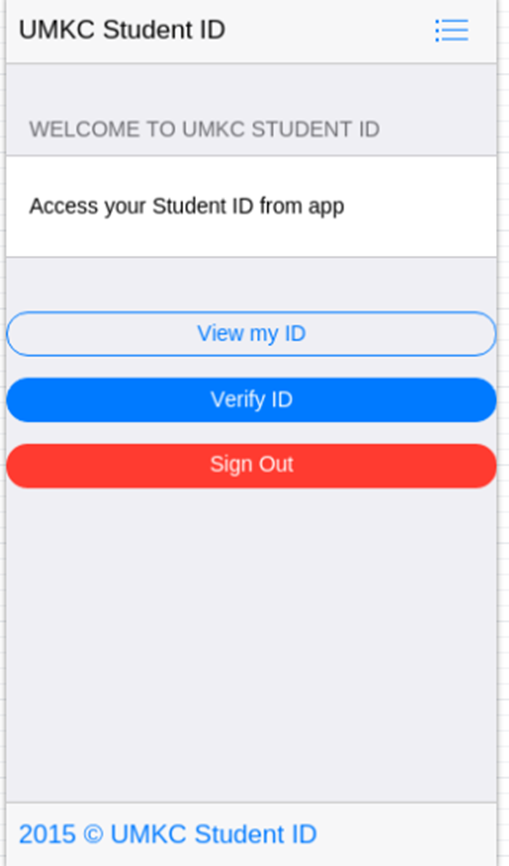


Figure Home Page

This is the screen that appears when student logs in. It has a menu containing options such as viewing the ID and verifying the ID.

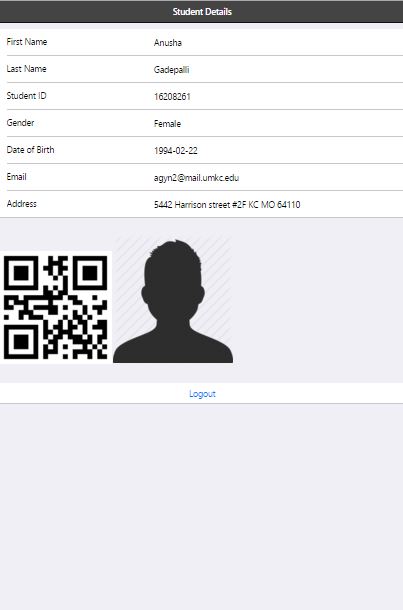
****

Figure 3 View ID page

On clicking view my ID button, the above page appears. It contains all the details of the student including the unique QR- code based on the username.

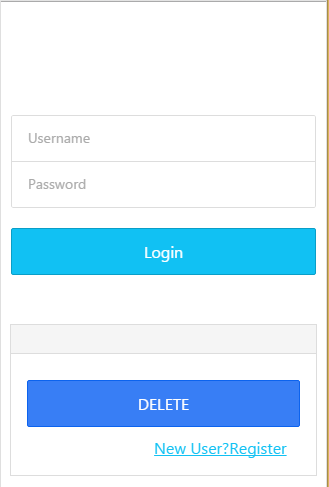
****

Figure 4 Admin Login page

The above page appears when the user clicks on admin login button.

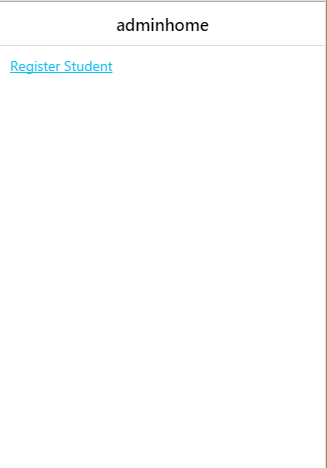


Figure 5 Admin Home page

This screen appears when the admin logs in.

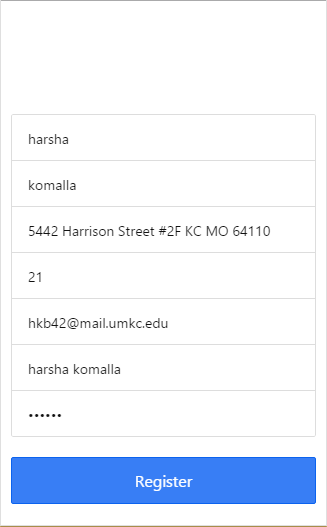
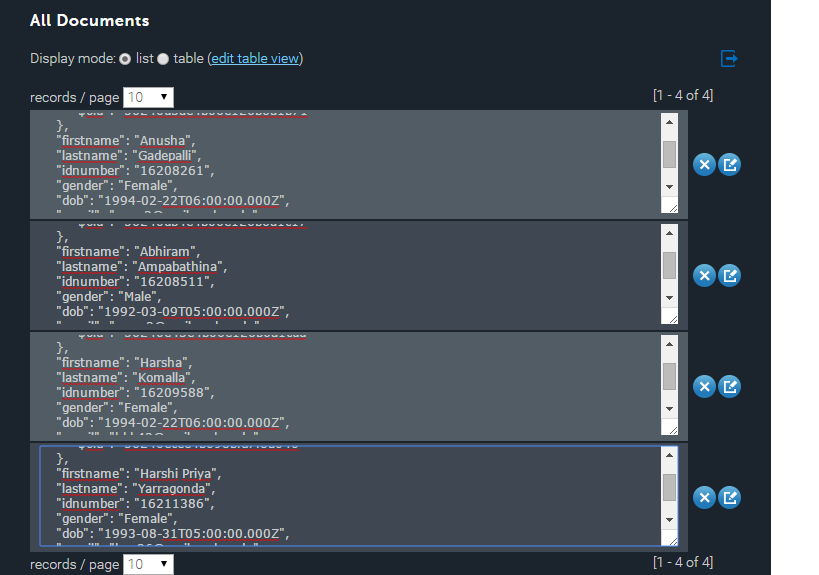
****

Figure 6 Register page

When admin wants to add a new student details then above screen appears.

**Database:**

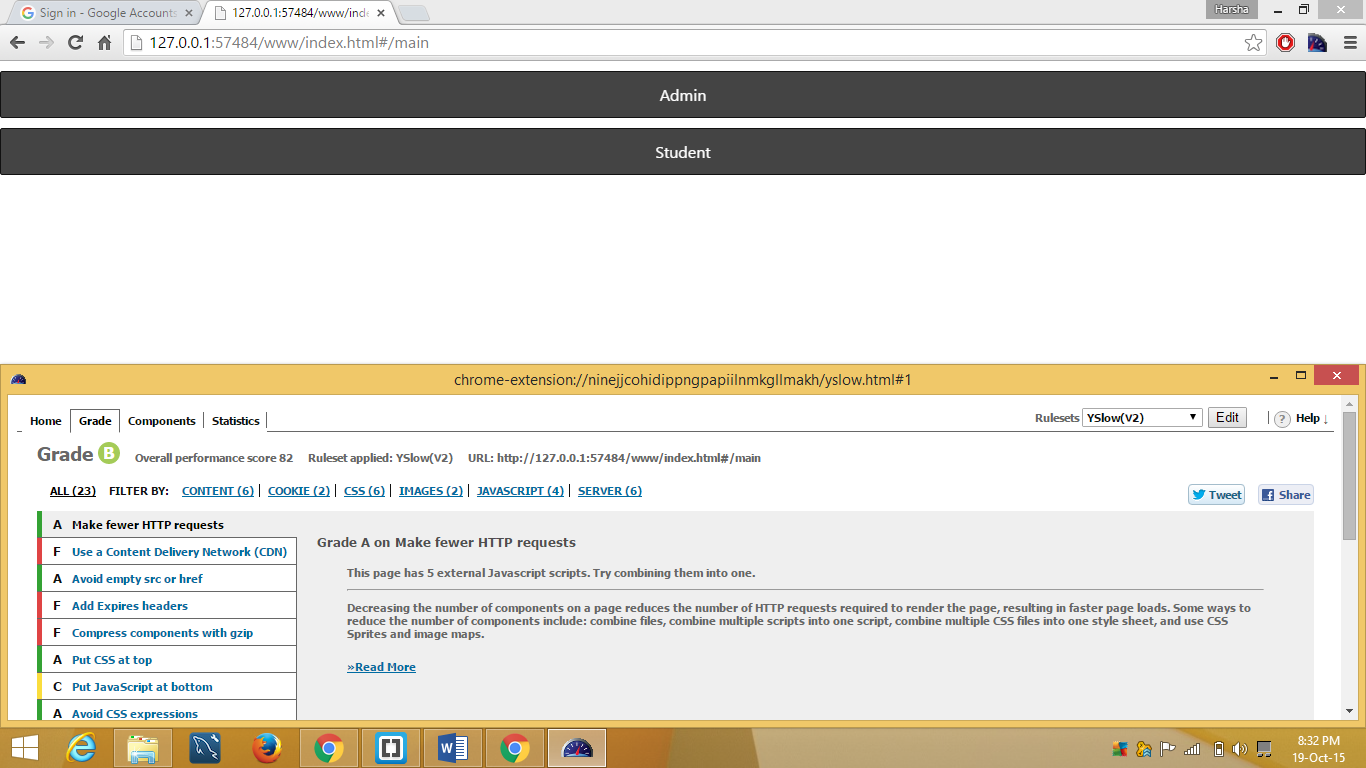
Details of the students has been added to the database.

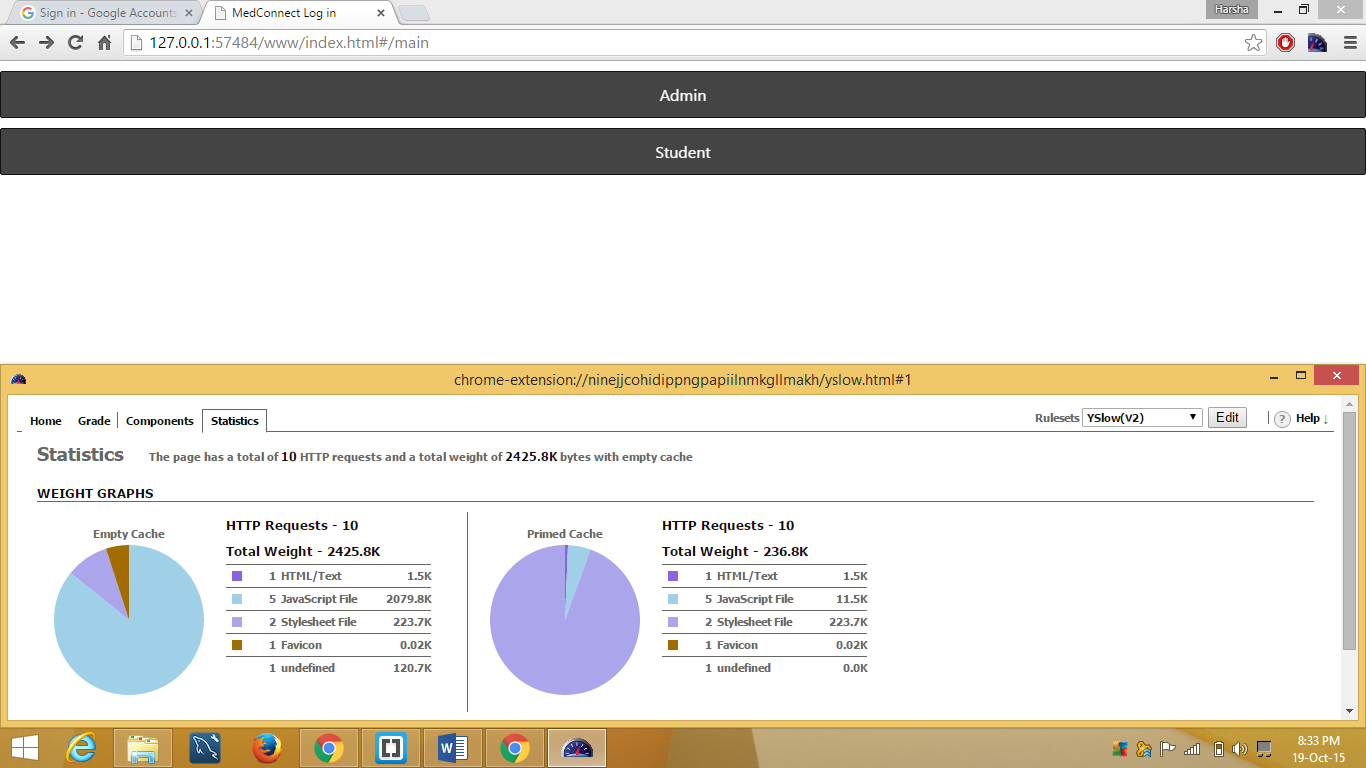
****

**V.Testing :**

Performance testing:

Using Yslow the testing has been made. The grade and statistics and shown in the below pictures.





**VI. Implementation:**

Server Side Implementation:

This implementation deals with the functionality of the application on the home, login, main, view and scan page.

The structural framework used to design this application is Angular JS.

The QR code API is able to generate the unique QR code based on the student ID and password. The scanner API is able to scan the QR code generated. The mongo DB databases hosted by the cloud database services include all the student information and this student information is retrieved and verified for the user information.

The API used to generate the unique QR code based on the user name and the password is

<https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=username+%27/%27+password>

The API used to read the QR code is

<http://api.qrserver.com/v1/read-qr-code/?fileurl=http%3A%2F%2Fapi.qrserver.com%2Fv1%2Fcreate-qr-code%2F%3Fdata%3DHelloWorld>

The plugin used to scan the QR code is

<https://build.phonegap.com/plugins/261>

The Mongo DB databases that is hosted by cloud database services is

<https://mongolab.com/databases/students>

Client Side Implementation:

**a) User Interface**

First phase implementation deals with designing the user interface and test cases for our home page, login and main page.

After installing the app, the user logs in to the app through student login. Once the clicks to login, the login screen opens. Student can log in to the app by entering student SSO id. The login details are validated properly by connecting to university database. If the login fails the user is not allowed to log in to the app.

This login is very important as only students/professors with valid SSO id are allowed to use this app. There will not be any registration for the users as they are already registered or identified by university.

If the student logs in they are given access to further features of application.

Validation is done for the login fields.

**b) Test Cases**

1. **Login Page:**

To test whether the login screen is opening or not by clicking “Login” in the menu or not. This test case is used to validate the login button.

2. **Valid** **Student ID:**

To test whether the student id is valid or not after entering a proper student id in the login screen. This test case is used to validate the student ID.

3. **Invalid** **Student ID:**

To test whether the student id is invalid or not after entering a wrong student id in the login screen. This test case is used to invalidate the student ID.

4. **Valid Password:**

To test whether the password is valid or not after entering a proper password in the login screen. This test case is used to validate the password.

5. **Invalid Password:**

To test whether the password is invalid or not after entering a wrong password in the login screen. This test is used to invalidate the password.

**6. Main Page:**

To test whether main page is opening after successful.

This test case is used to validate the main page.

**7. View page:**

To test whether the student details are in the view page.

This test cases are used to validate student details.

**8. Scan page:**

To test whether the QR code generated is able to scan.

This test case is used to validate scan page

**Test Requirements**

Student Id

* Should contain alphabets and numbers
* Should not be more than 10 characters
* Should not contain special characters
* Should not be blank

**Test Scenarios**

* To verify whether the login page is opening or not
* To verify whether the username is a proper student sso id or not.
* To verify whether the password is a proper password or not.
* To verify whether the main page is not opened if the login credentials are wrong.
* To verify whether the main page is opened after successful login of credentials.
* To verify the page is opening to view ID page and scan ID page.
* To verify the view ID page contains the student details along with QR code and scan ID page is opening to the qr-code which is to be able to scan.

**VI. Project Management**

**Second Increment Work done**

* Created designs, User interface for home page, login page and main page of the mobile application.
* Validated all the screens in the mobile application.
* Responsibility
  + Database connection – Anusha , Abhiram
  + Student Login and View ID – Harsha Komalla, Harshi Priya
  + Admin Login page - Anusha , Abhiram
  + Validations - Harsha Komalla, Harshi Priya

**Work to be completed**

* Complete rest of the screens for the mobile app.
* Create Scanner to scan the QR code id.
* Use REST API and camera scanner for the mobile app.

Verifying the details of the user with the database after scanning the QR code.