**CS551 Advanced Software Engineering**

**Third 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>

The MongoDB databases that is hosted by cloud database services is

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

Services used in mongo DB are

**PUT:** Put service is used to update the student details in the database. In this app both the admin and student uses this to update the student profile

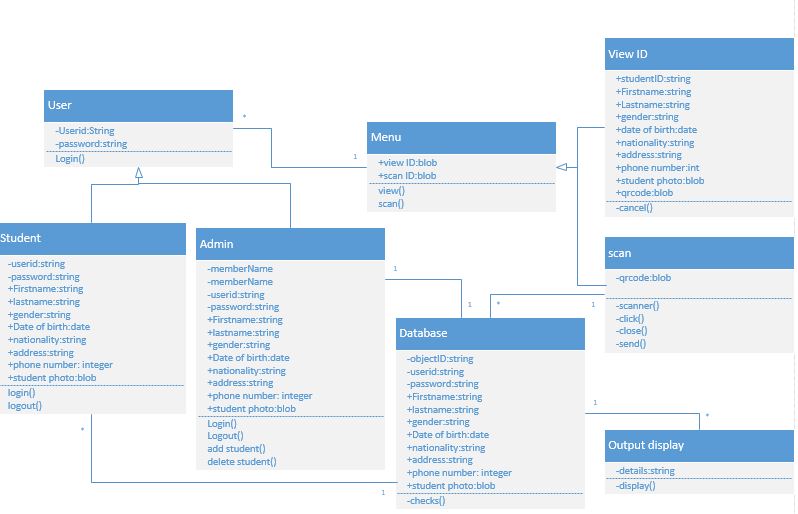
**GET:** Get service is used to retrieve student details and admin details from the database. It is also used to display student details in View ID page.

**POST:** Post service is used to post the details of the student in the database. This service is mainly used to register student

**Delete:** Delete service is used to delete student from the database. This service is used by the admin to delete the student records

**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.

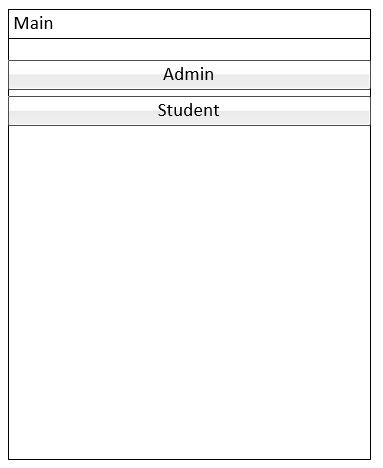


**State Diagram:**

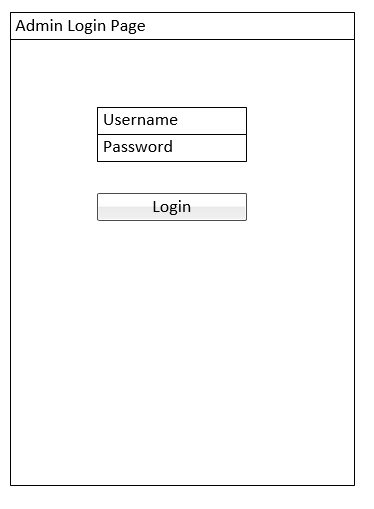
A state diagram defines behavior of a system. The current behavior of the system is dependent on the preceding state of the system. In our application, if the login is successful a menu is displayed else the system is stopped. After login if the user selects to view the details then the details are viewed 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:**



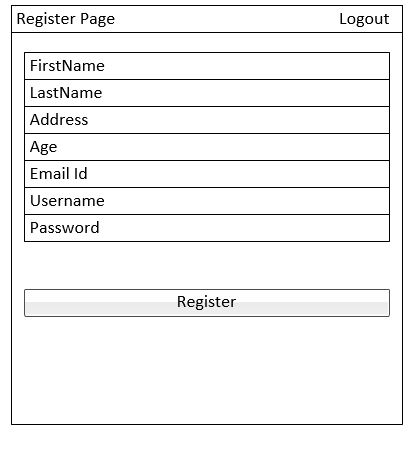
**Main page**: It has two buttons and the user has the option to choose based on whether the person is the student or the admin. On click of the Admin button it opens to Admin Login page and on click of student it opens to Student Login page.



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

****

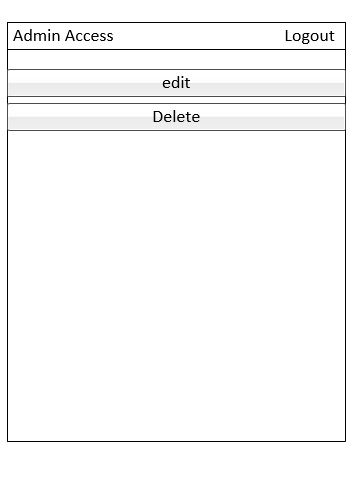
**Admin Home:** The admin can select one of the options Register or Access the student Account. If Register button is clicked, then it opens to Student register page where the admin can register the students and Access button opens to access page

****

**Register Page:** Admin can register student from the register page. It contains various details of the student along with unique username and password. On click of register button, the student can be registered

****

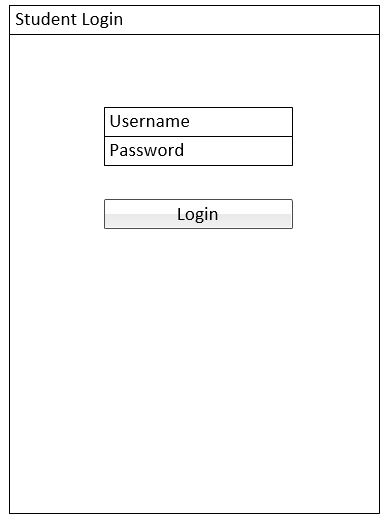
**Admin Home:** Admin can enter the username of any registered student and on click of Authorize it opens to admin access page.

****

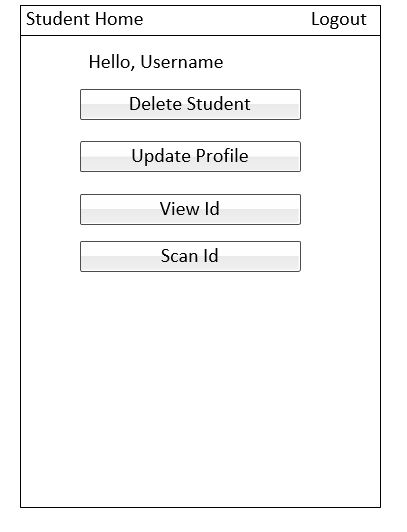
**Admin Access Page:** Admin has the access and can edit and delete the student profile on click of edit and delete options respectively.

****

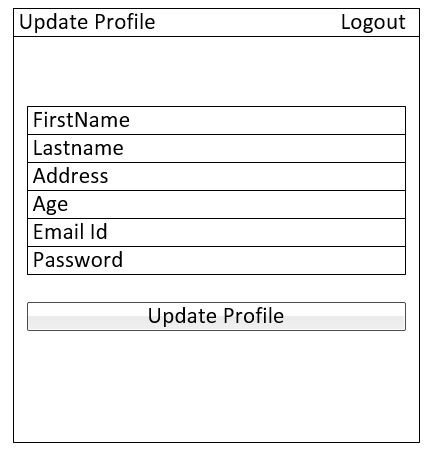
**Admin Edit Page:** Admin can edit the student information and can update students profile on click of admin edit.



**Student Login Page:** Student can login with unique username and password using the login button which opens to Student Home page.

****

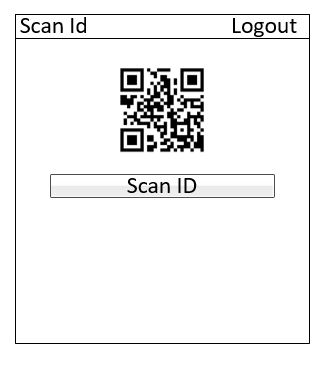
**Student Home:** In the student home page, after the student login it shows the username and has four buttons to edit or delete his/her profile, view his/her id and scan id. On click of delete Student the student can delete his profile and on click of update it opens to update page, view id button opens to view id page and scan id button opens to scan id page.

****

**Update Profile page:** Student can edit the information and can update student profile on click of update profile button.

****

**Student id:** The student id along with unique QR code can be displayed in the Student id page

****

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

**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.

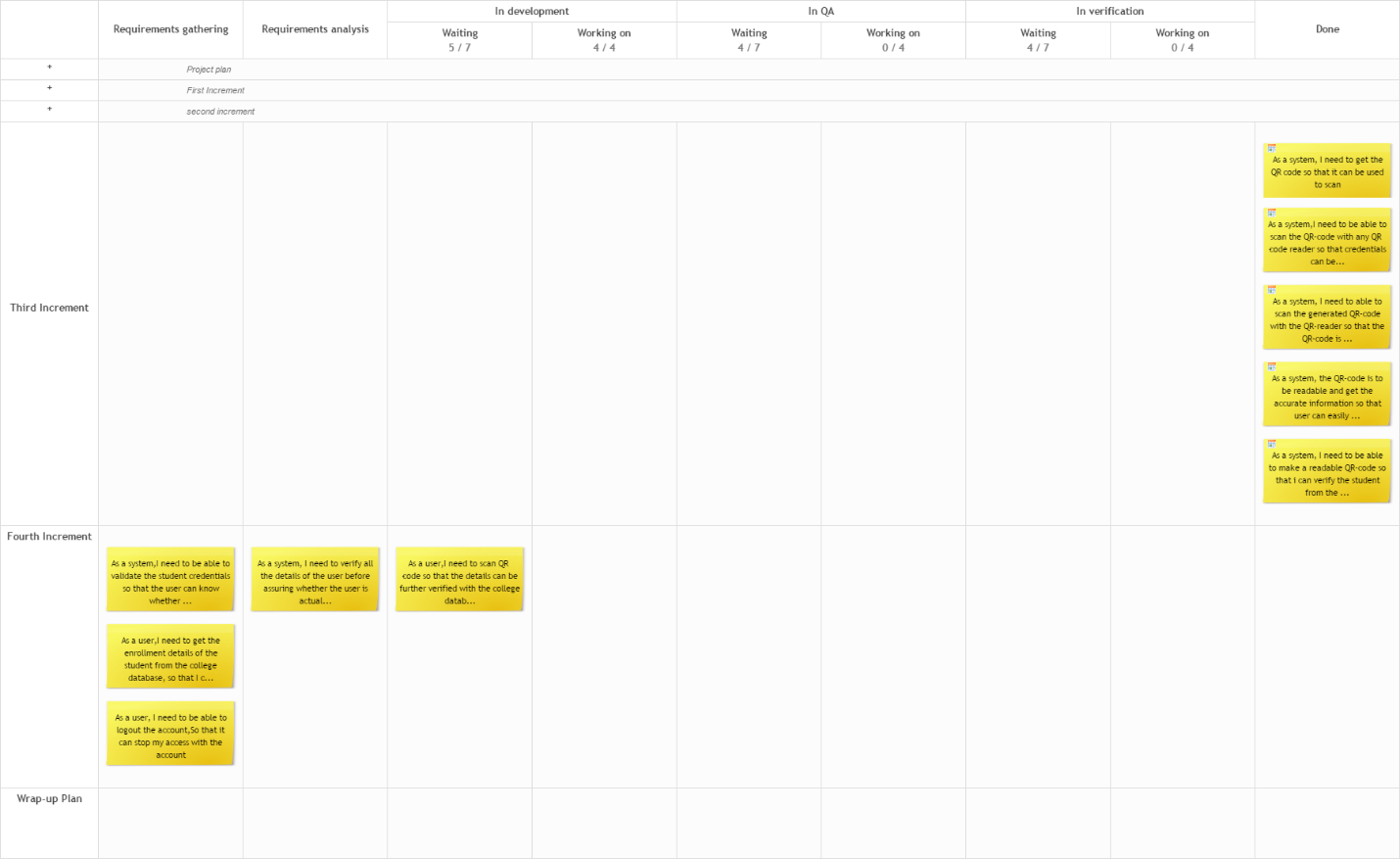
As a system, I need to be able to retrieve student details from the database along with the QR-code so that QR-code is scanned.

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 a system a scanner plugin is used to scan the QR code.

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.

**Mockups:**

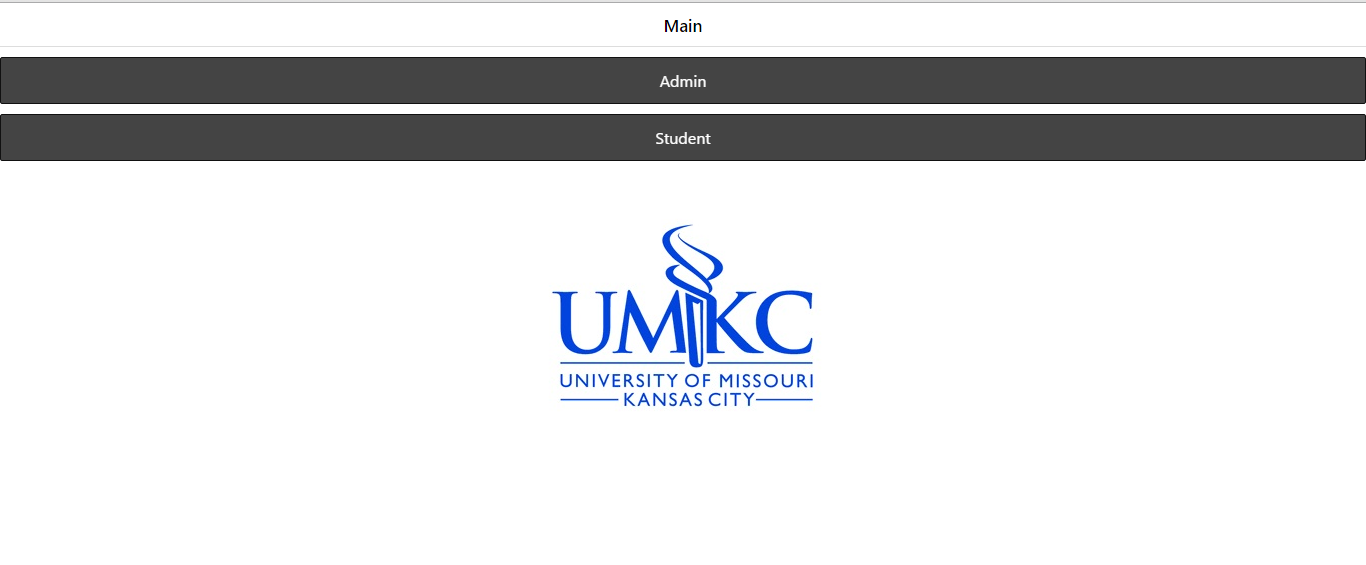


Figure 1 Main Page

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

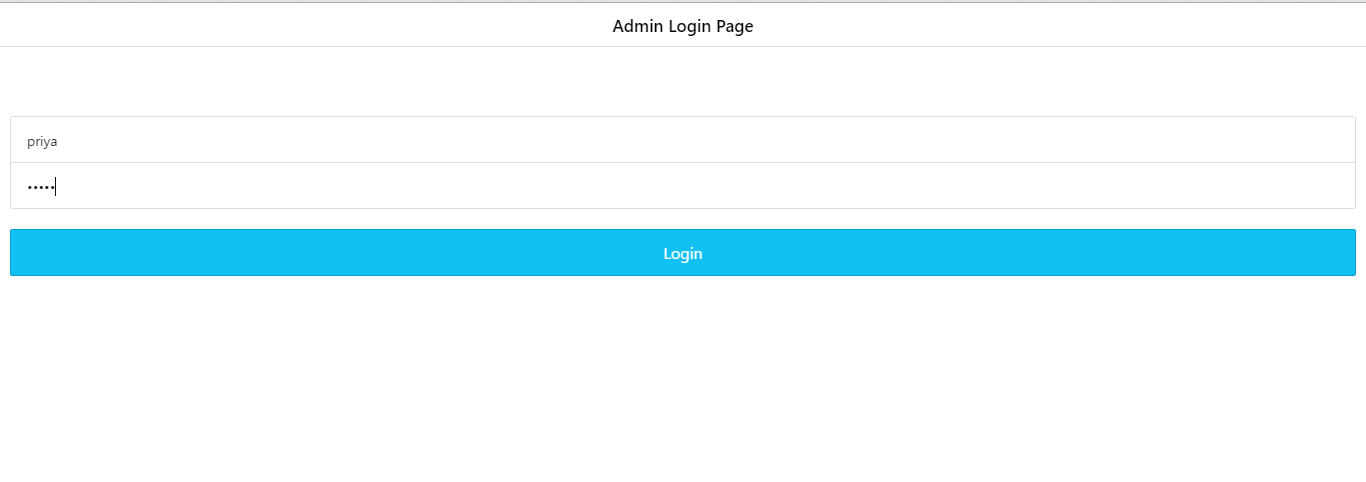


Figure 2 Admin Login Page

This is the page where the admin can login

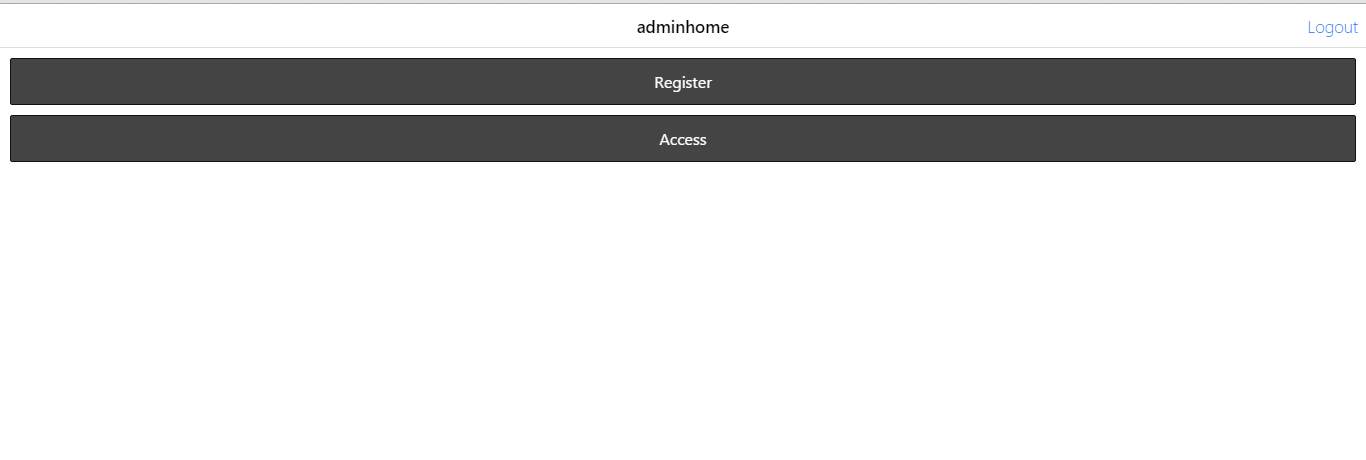


Figure 3 Admin Home Page

The admin can register a new student or can access the student account using the two buttons register and access.

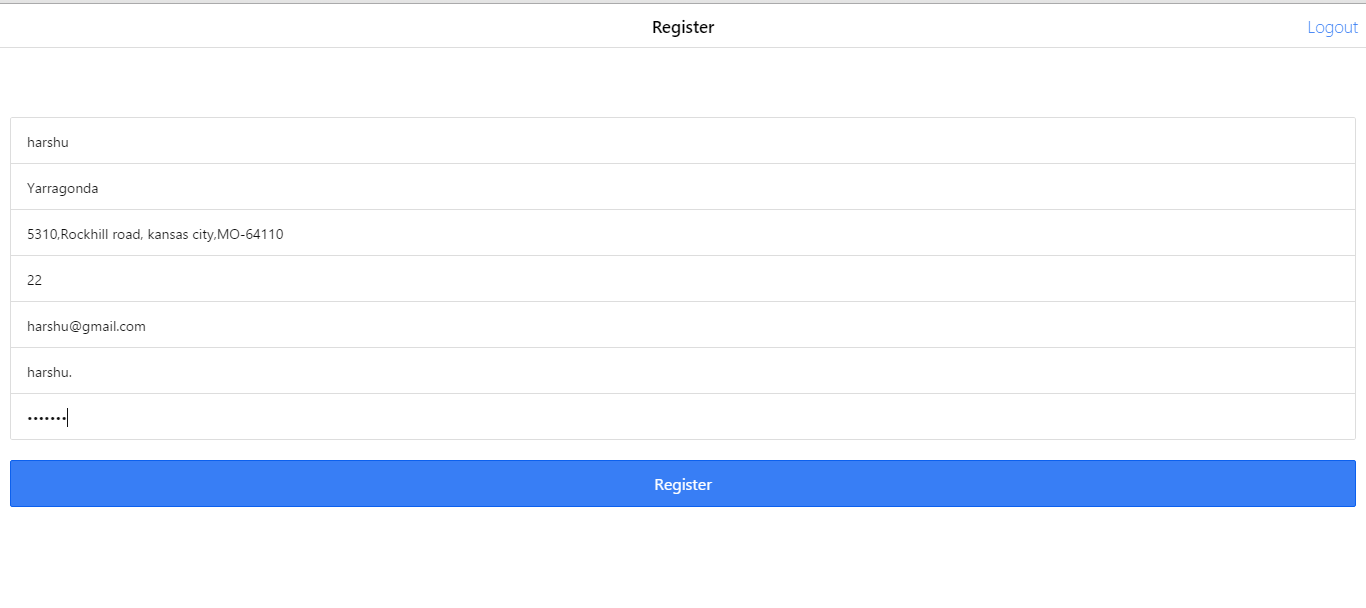


Figure 4 Register Page

Admin can register the student in this page

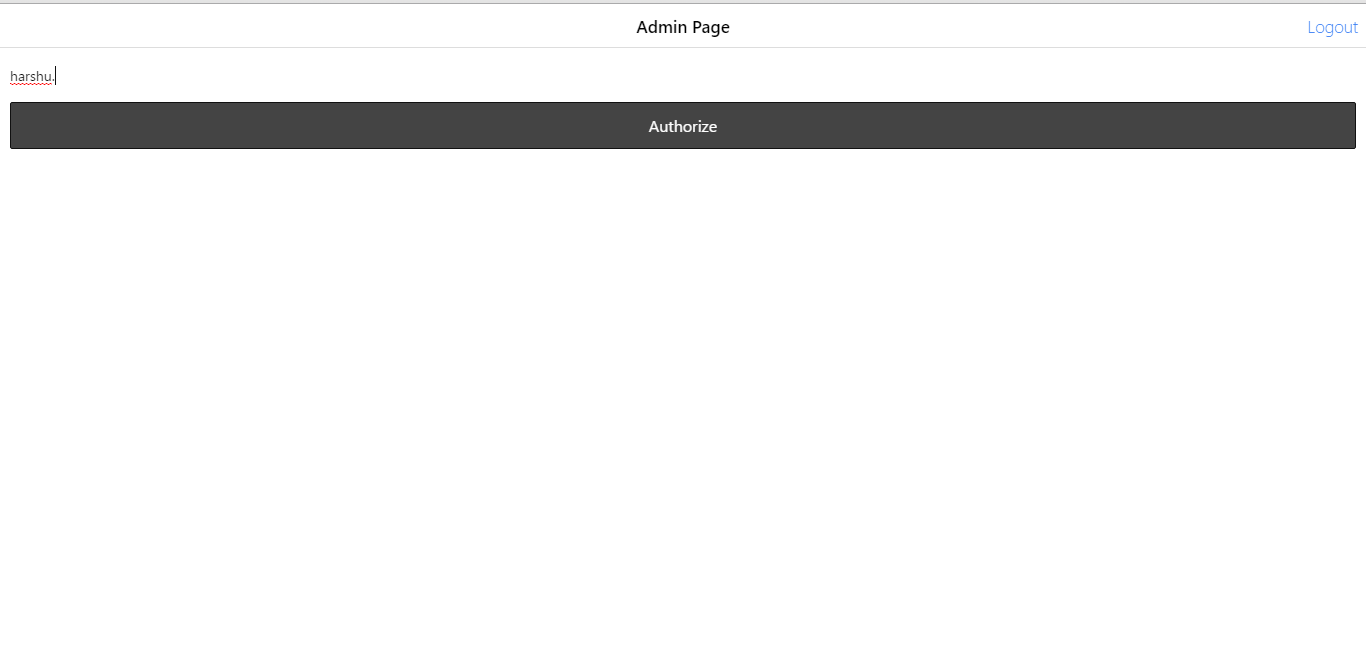


Figure 5 Admin Page

Admin can give the student name and access the student account using authorize button

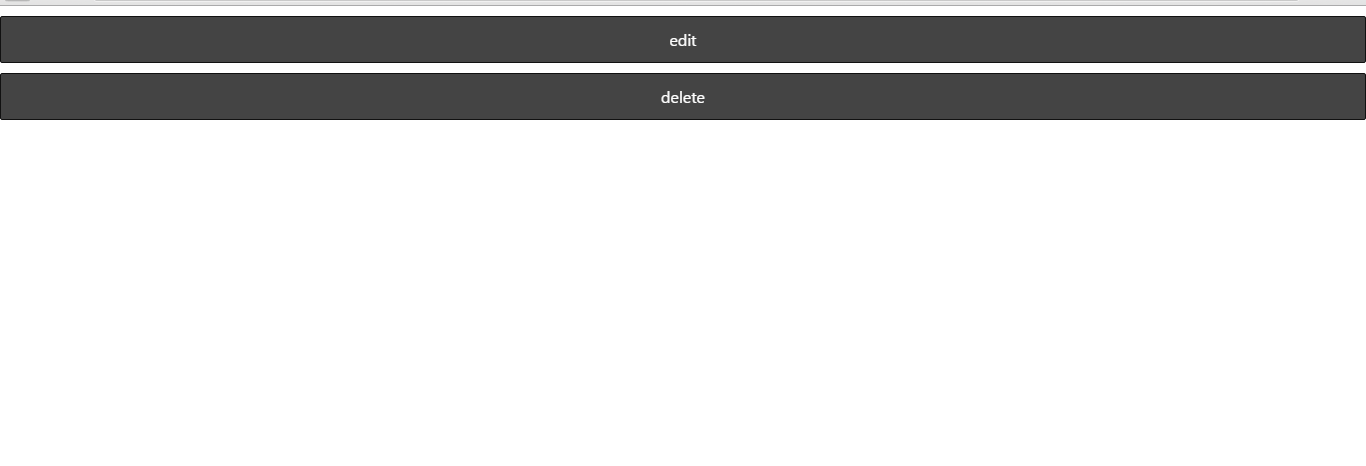


Figure 6 Access Page

Admin can edit and delete the student account in this page

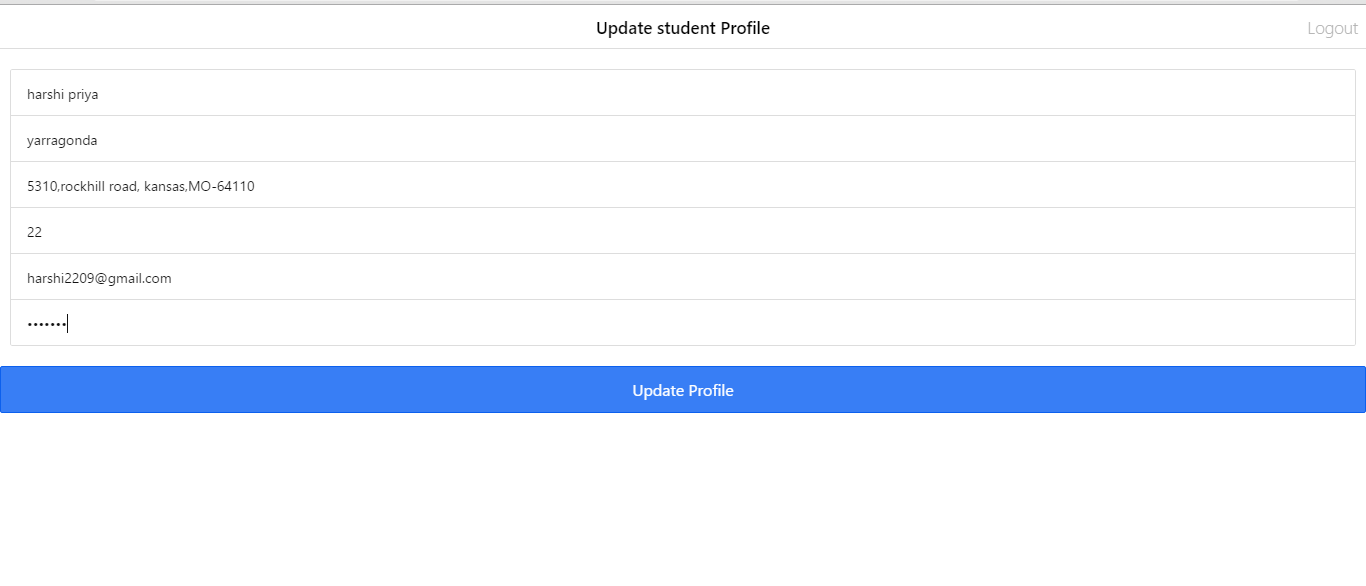


Figure 7Update student Profile

Admin can admin student information in this page

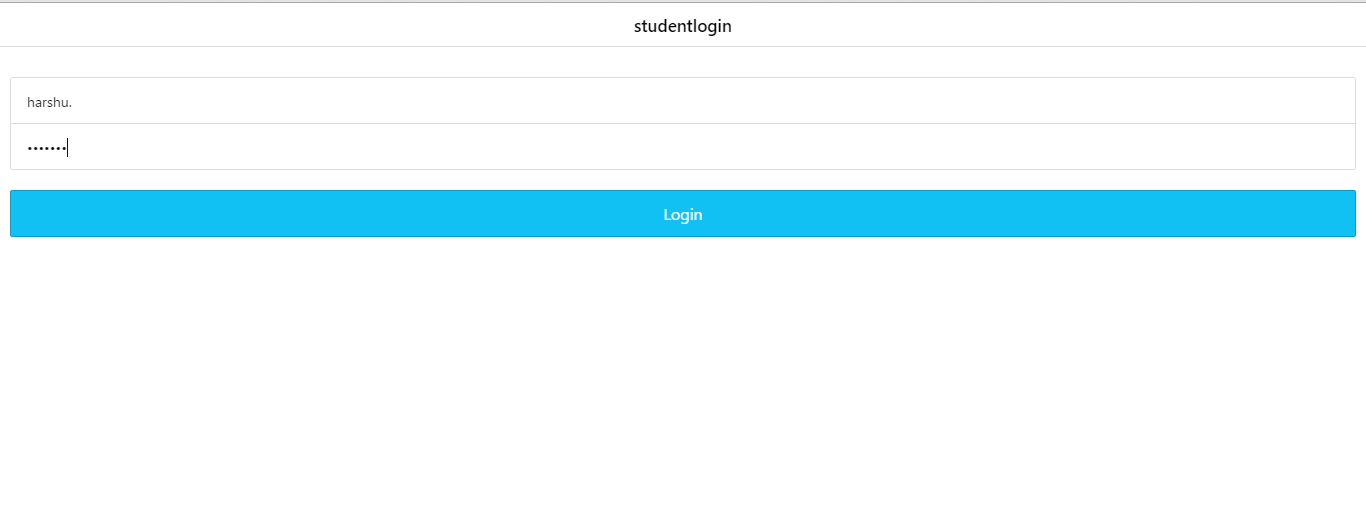


Figure 8 Student Login Page

The student can login using the username and password from this page

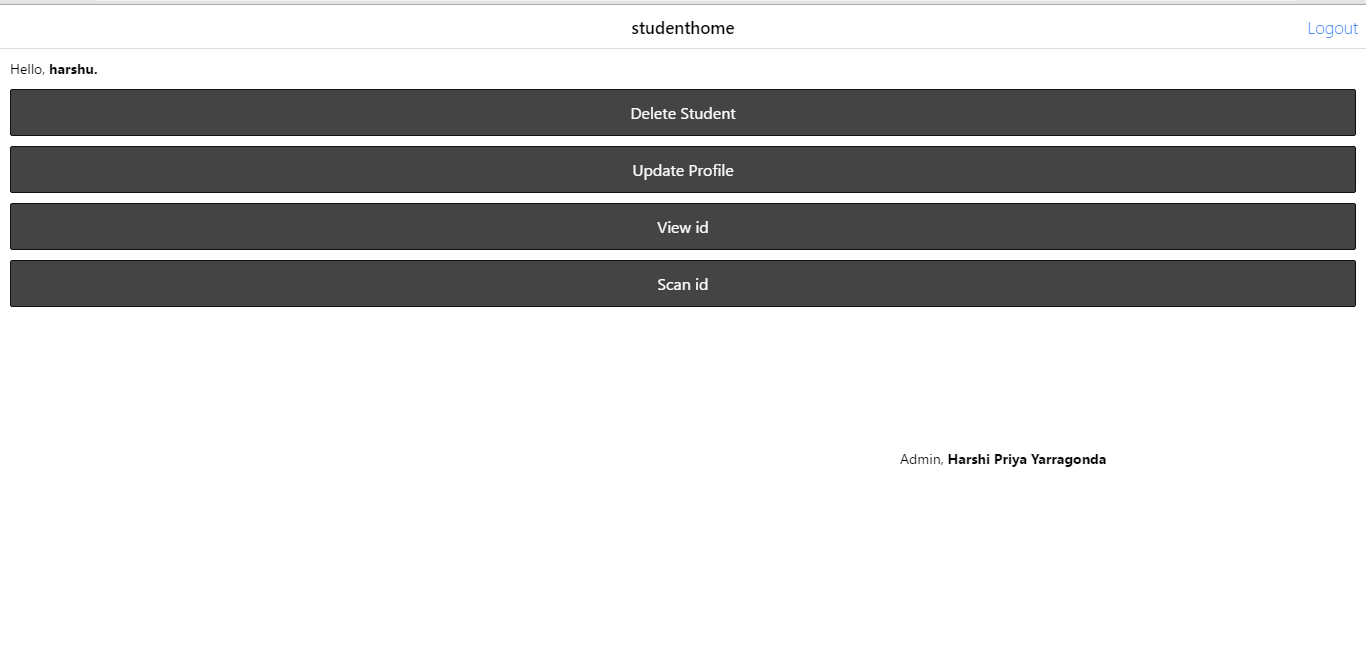


Figure 9 Student Home Page

Student has the options to delete, update the profile, view the ID and Scan the ID

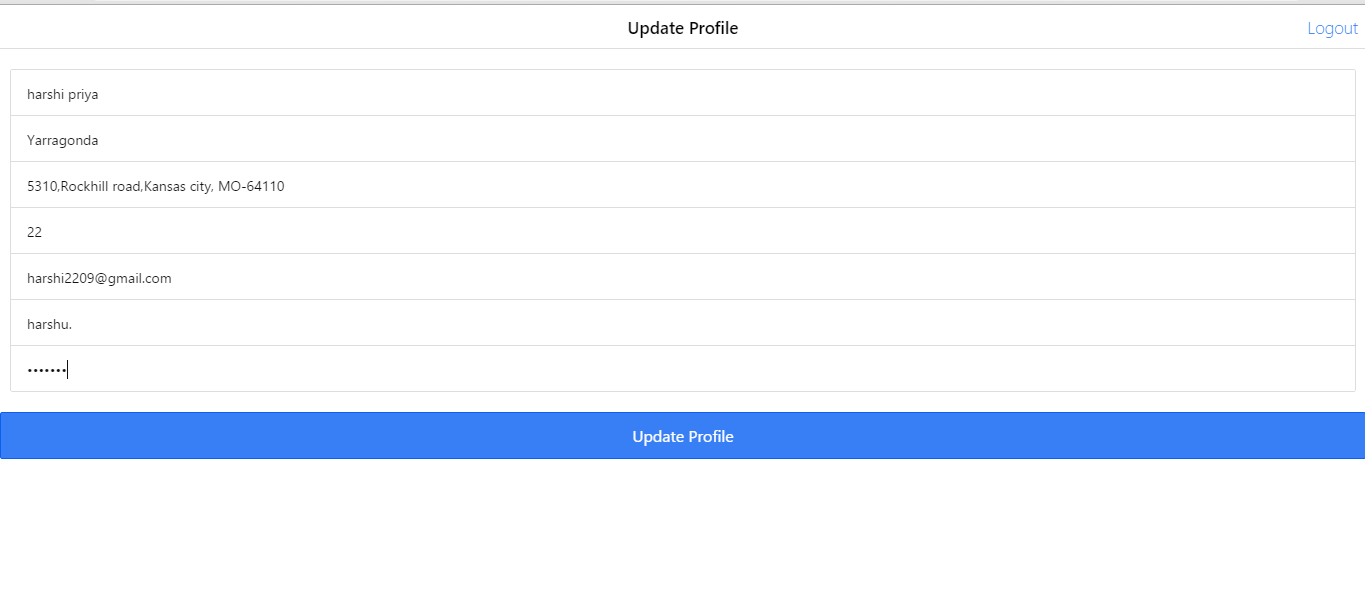


Figure 10 Admin Home Page

Student can update his/her own profile

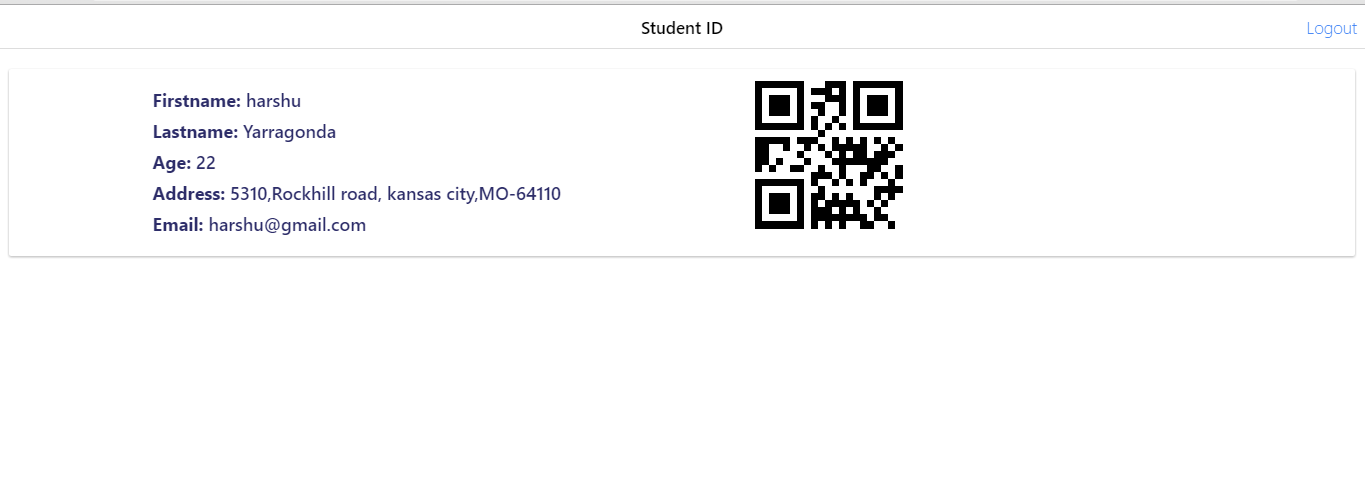


Figure 11 Student ID Page

The student Id along with unique QR code can be displayed in this page

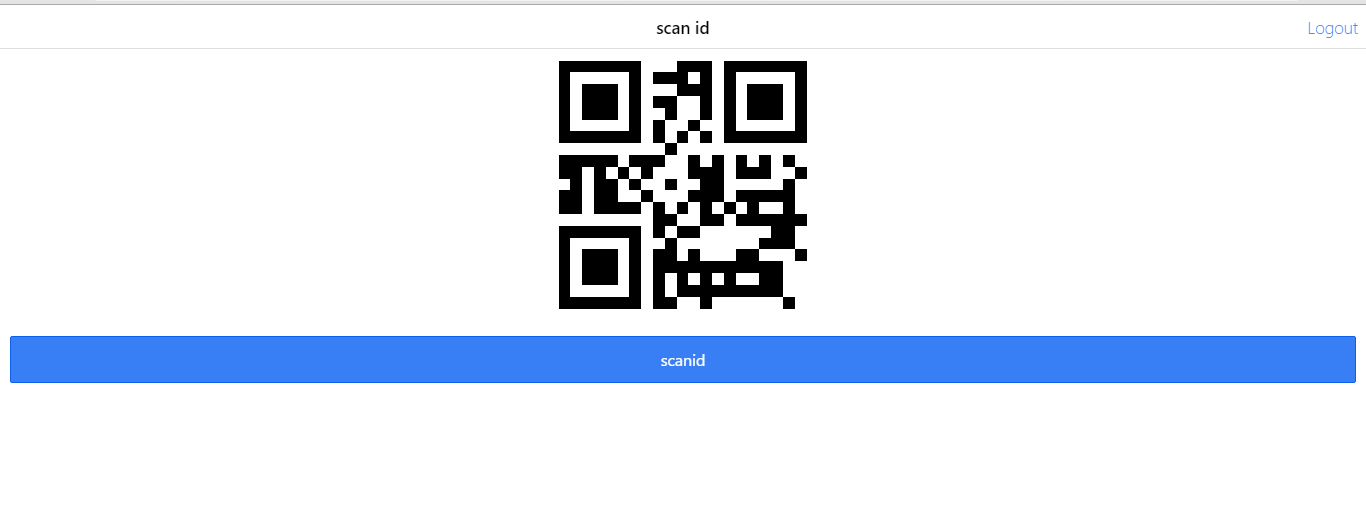


Figure 12 Scan ID Page

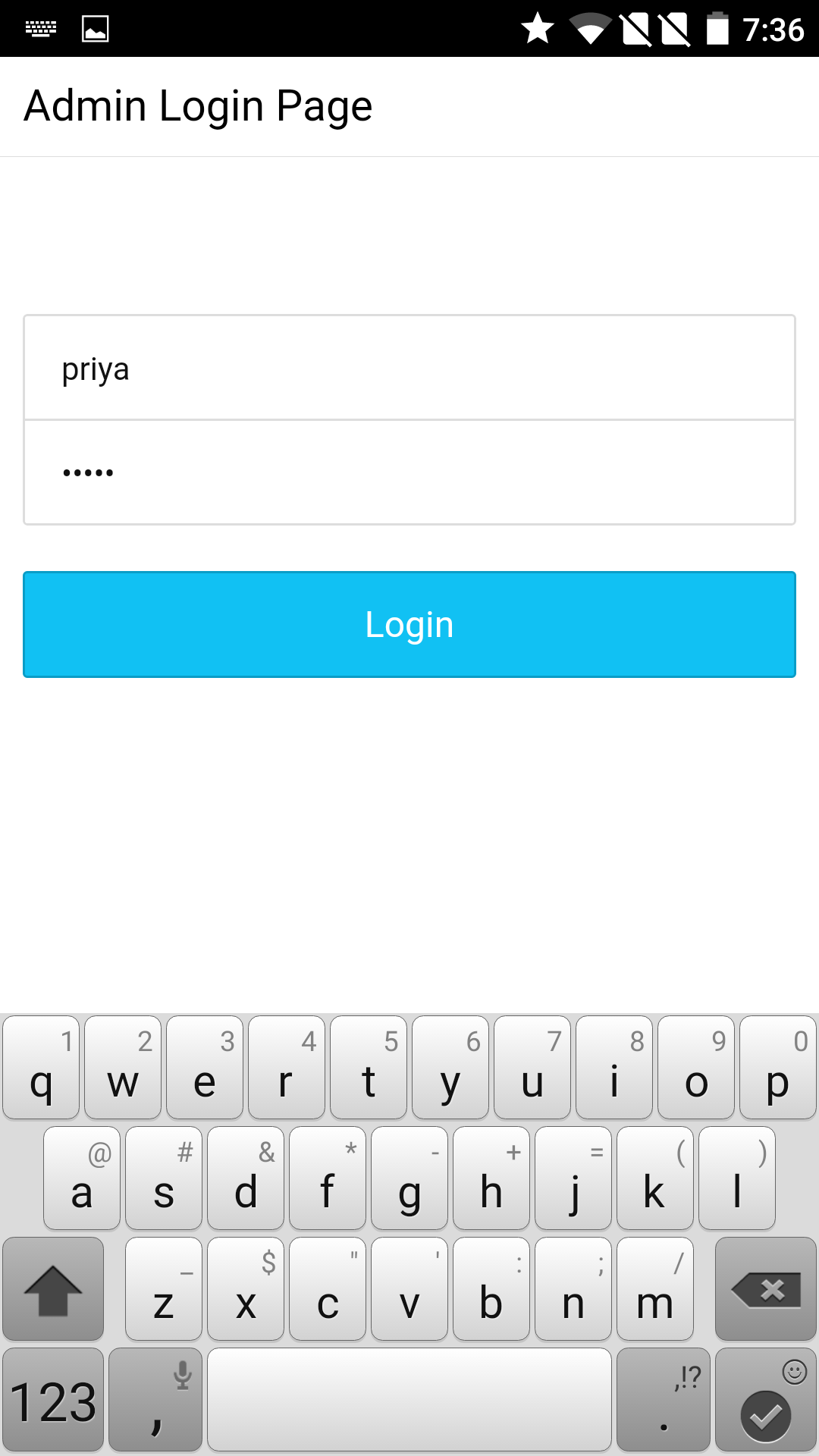
In this page the student ID can be scanned.

**Deployment of Mobile App in Smart Phone**

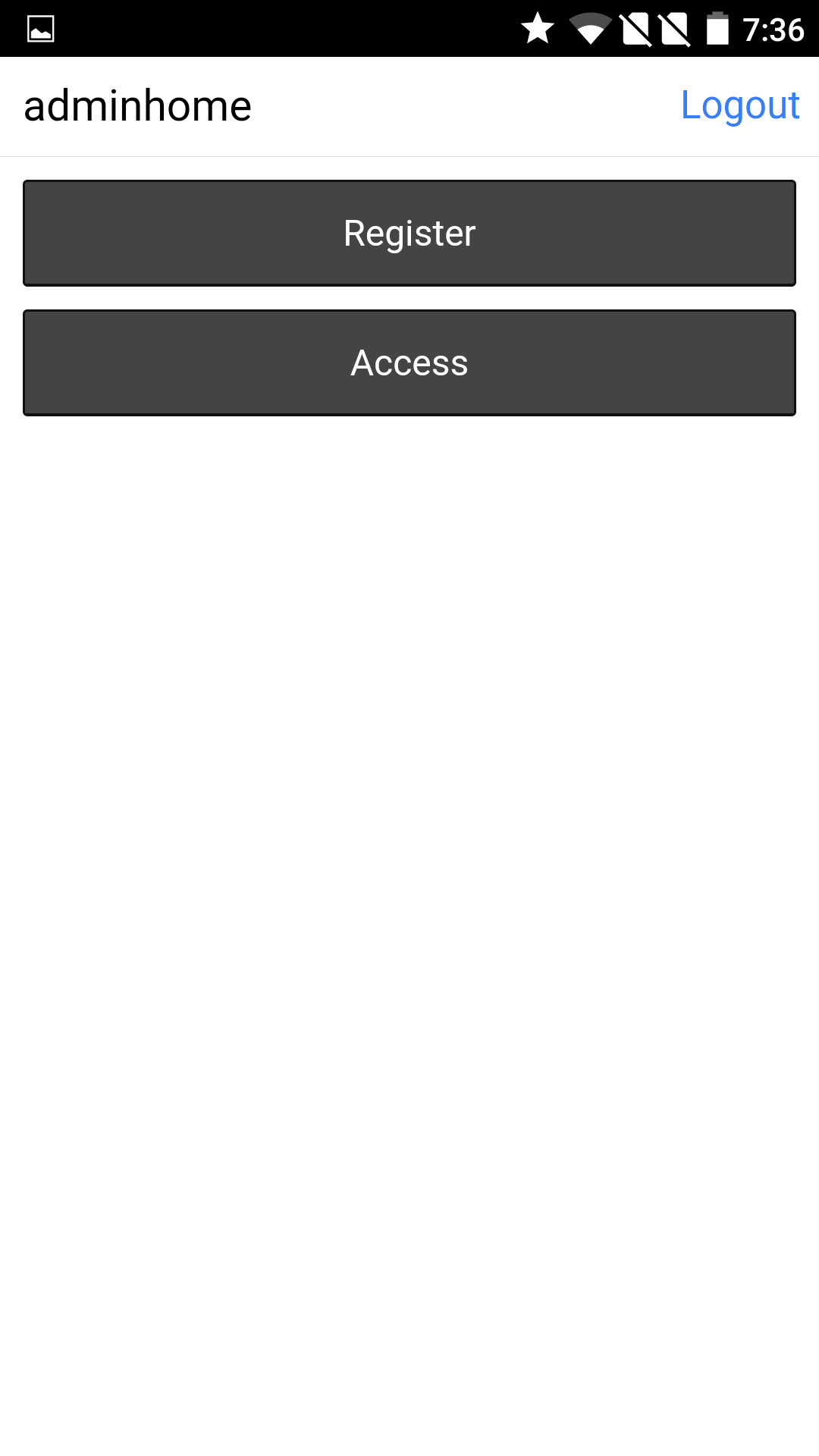
**1. Main page**: It has two buttons and the user has the option to choose based on whether the person is the student or the admin. On click of the Admin button it opens to Admin Login page and on click of student it opens to Student Login page.

****

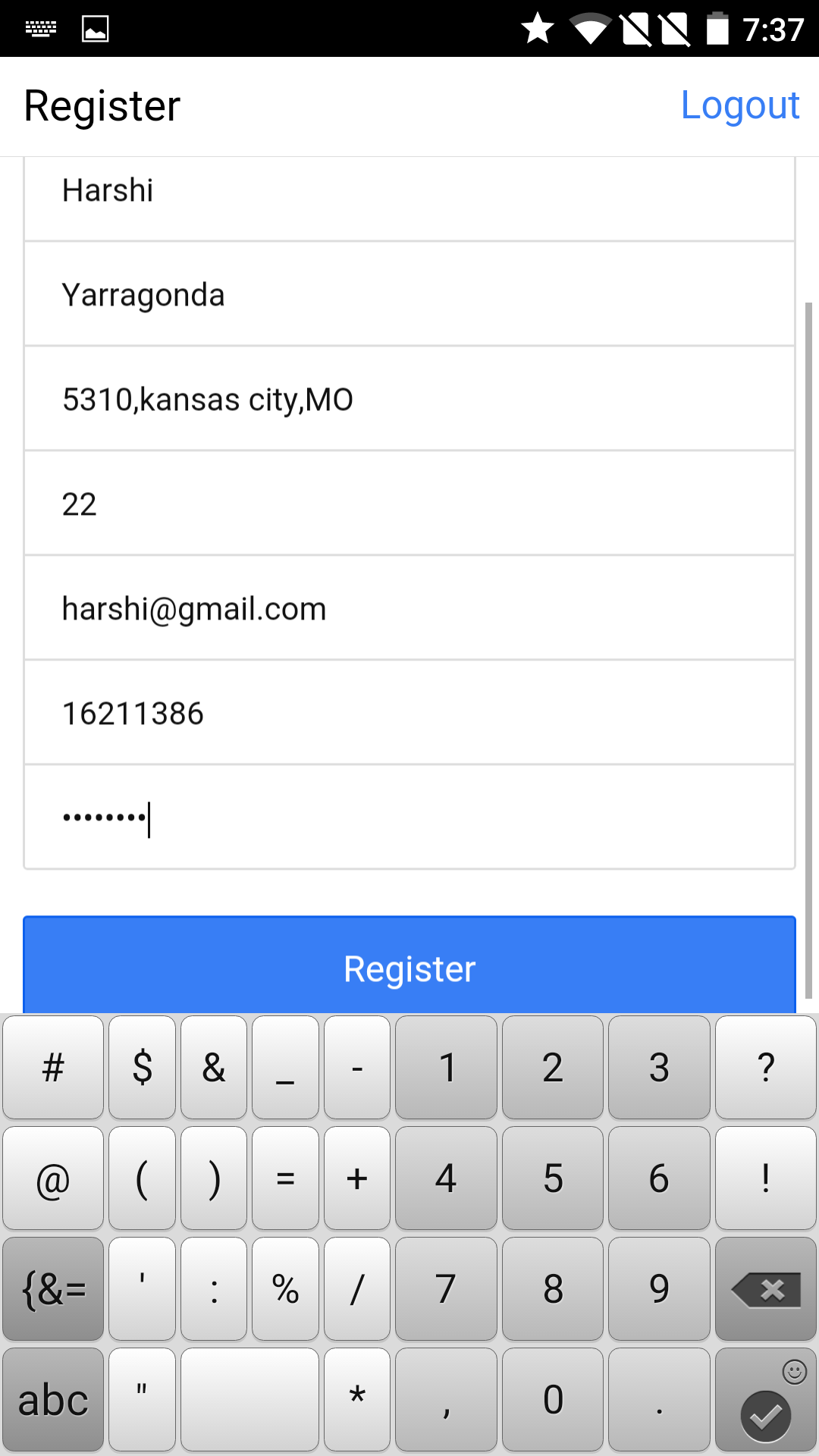
**Admin Login page**: The unique Admin Username and password are to be entered to login to the account. On click of the login button, it opens to Admin home page



**Admin Home:** The admin can select one of the options Register or Access the student Account. If Register button is clicked, then it opens to Student register page where the admin can register the students and Access button opens to access page



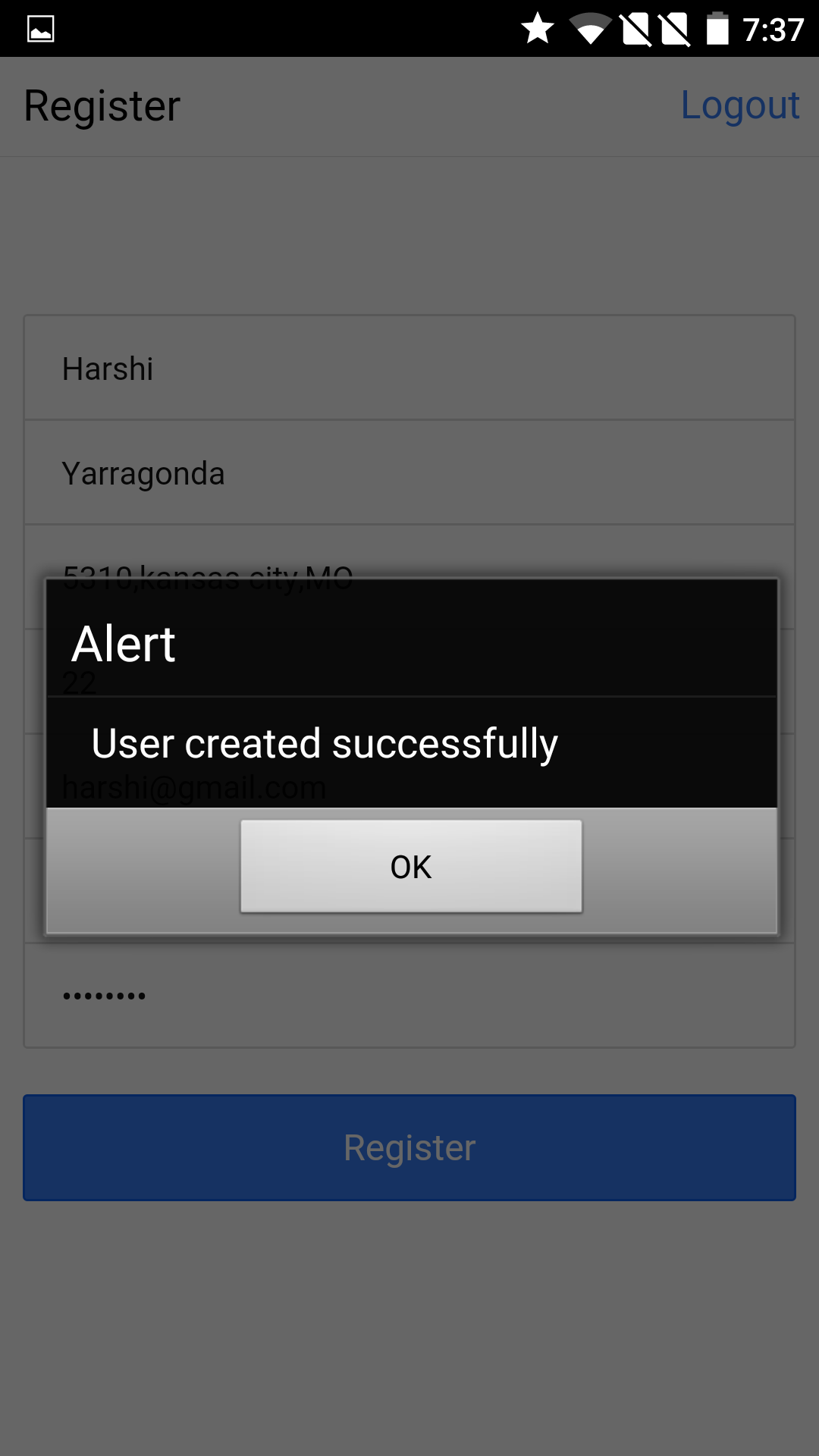
**Register Page:** Admin can register student from the register page. It contains various details of the student along with unique username and password. On click of register button, the student can be registered



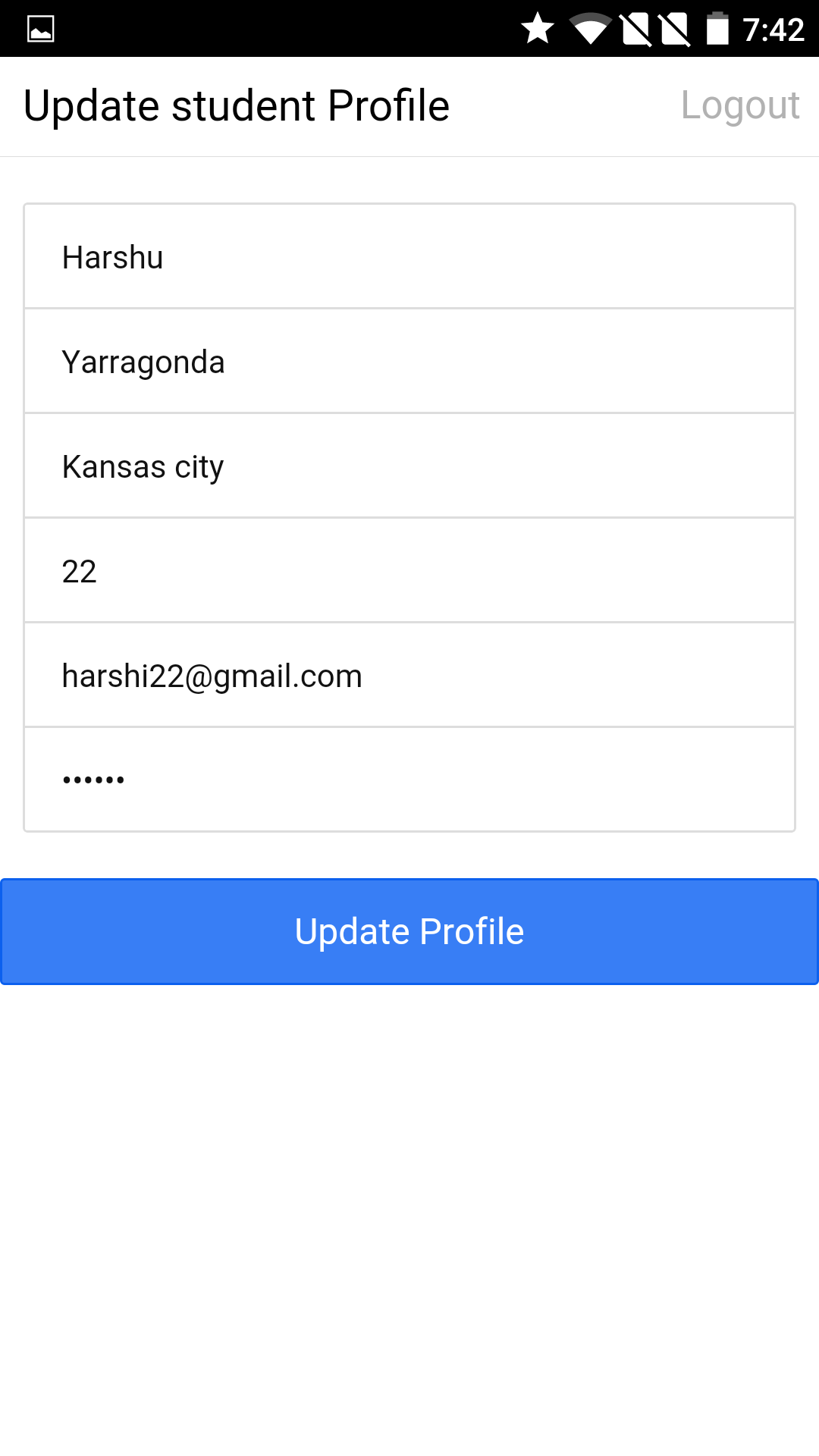
**Admin access page:**

****

**In this page we can see that the admin registered the student successfully**



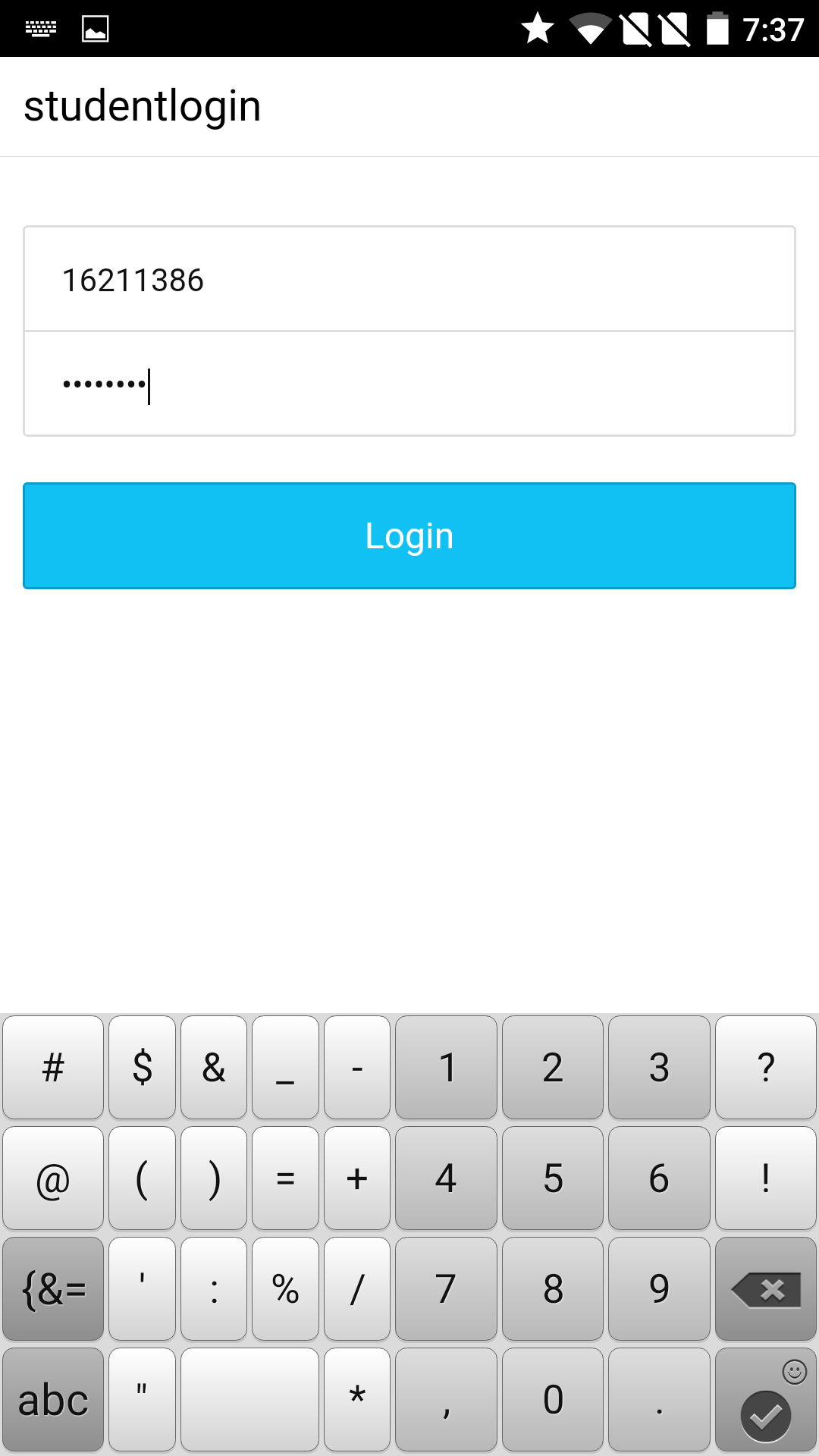
**Update student Profile Page:** Admin can edit the student information and can update students profile on click of admin edit.

****

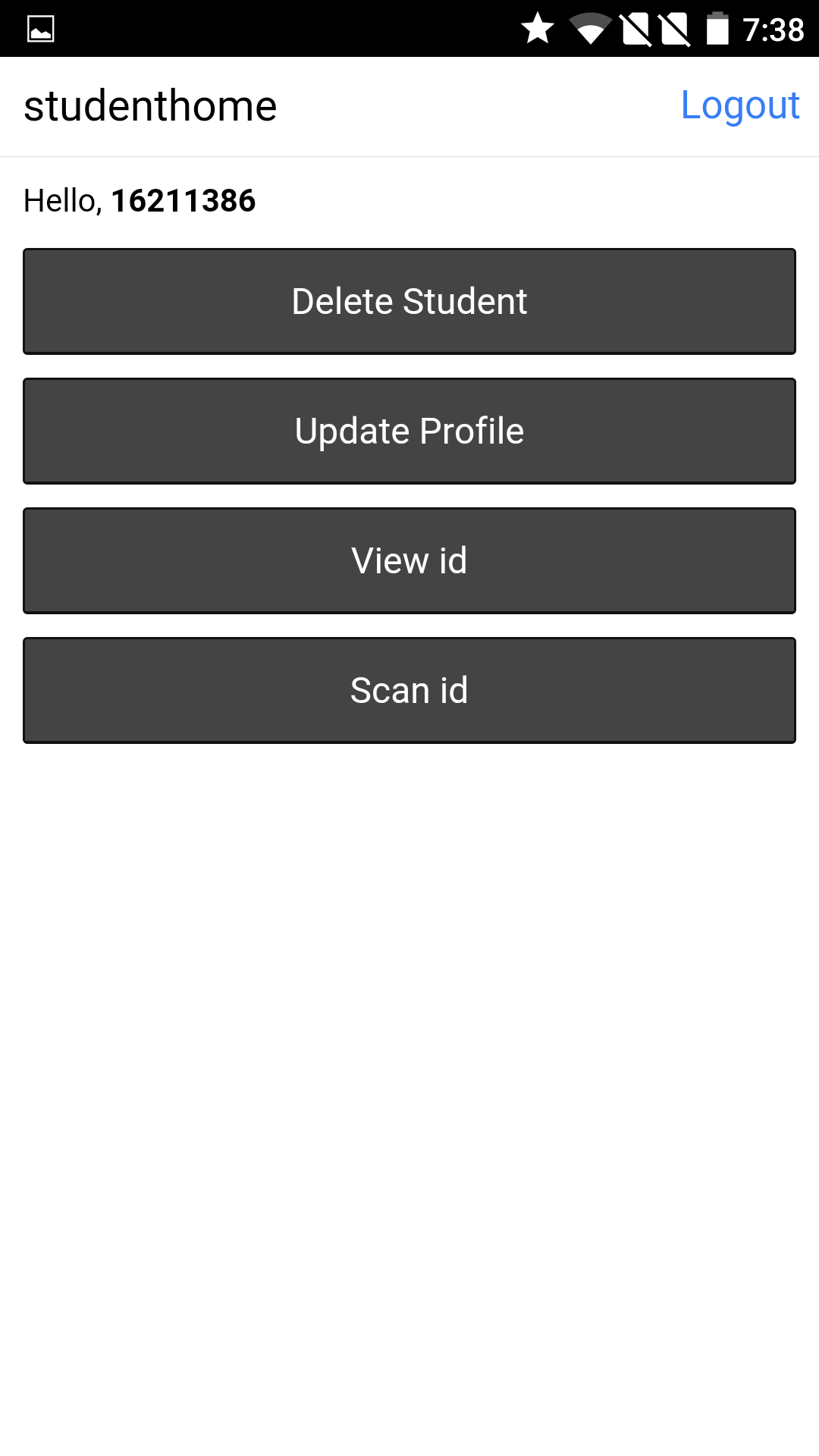
In this page, the admin delete the student record from the database.

****

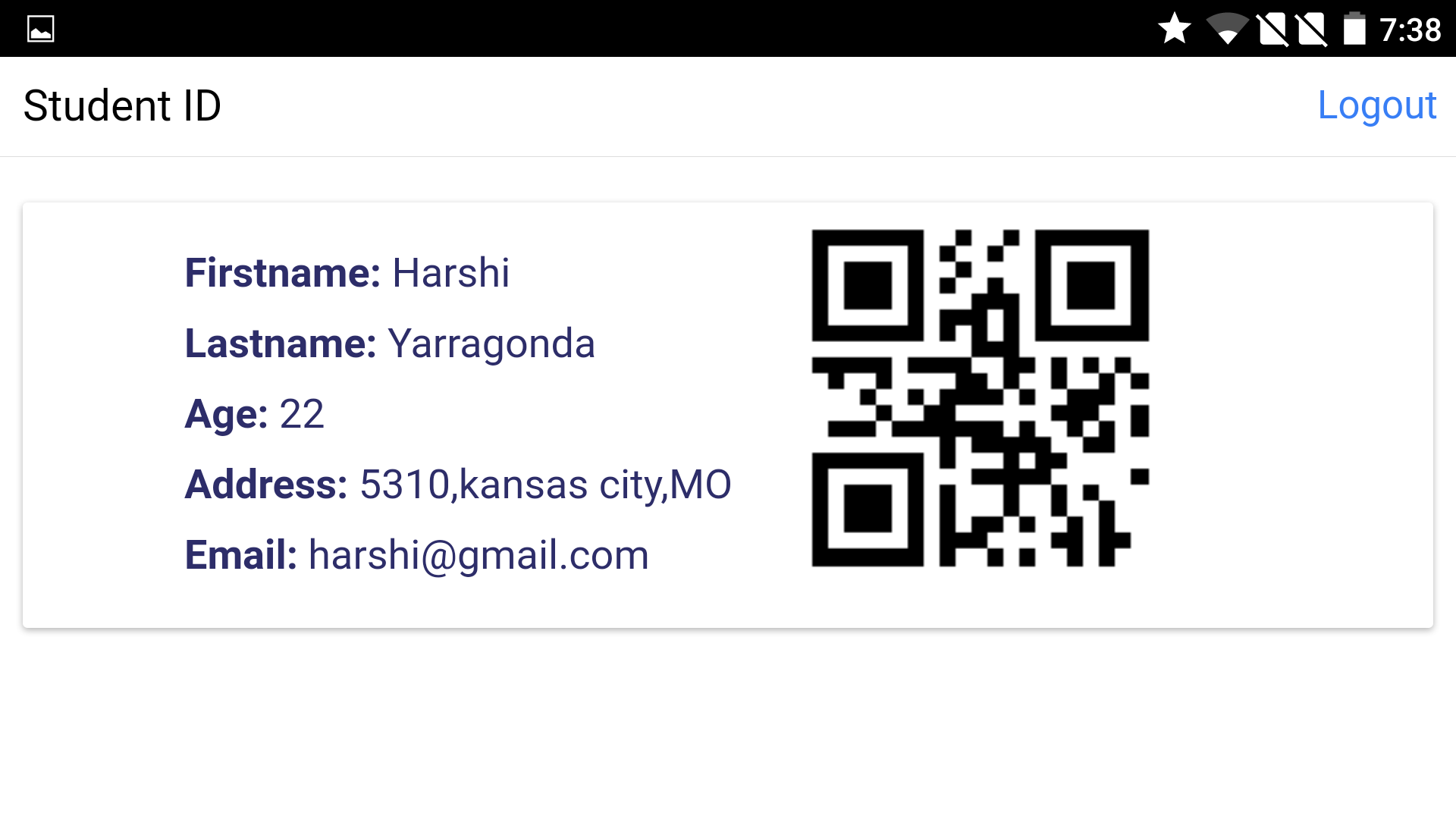
**Student Login Page:** Student can login with unique username and password using the login button which opens to Student Home page.



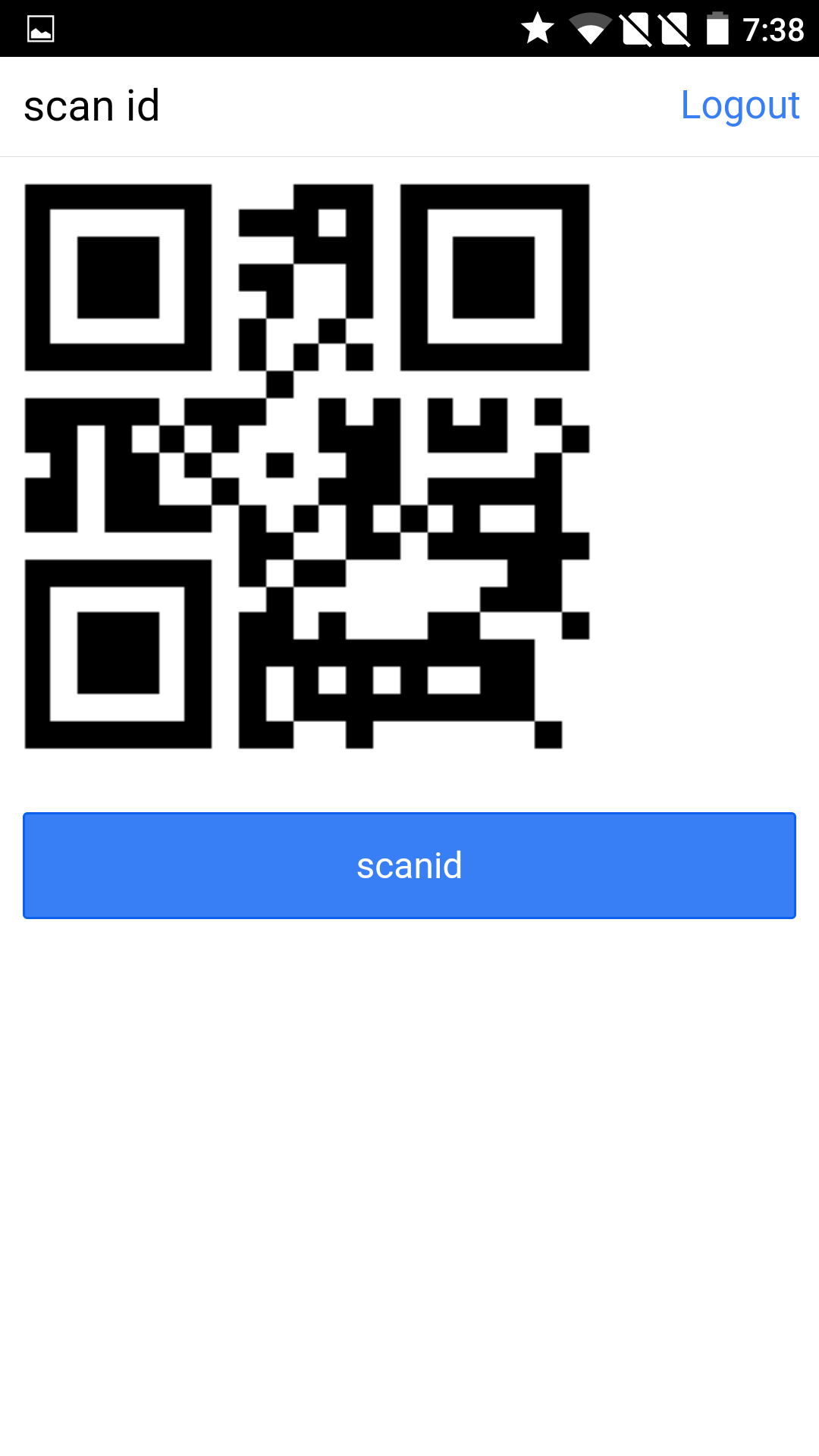
**Student Home:** In the student home page, after the student login it shows the username and has four buttons to edit or delete his/her profile, view his/her id and scan id. On click of delete Student the student can delete his profile and on click of update it opens to update page, view id button opens to view id page and scan id button opens to scan id page.



**Student id:** The student details along with unique QR code can be displayed in the Student id page



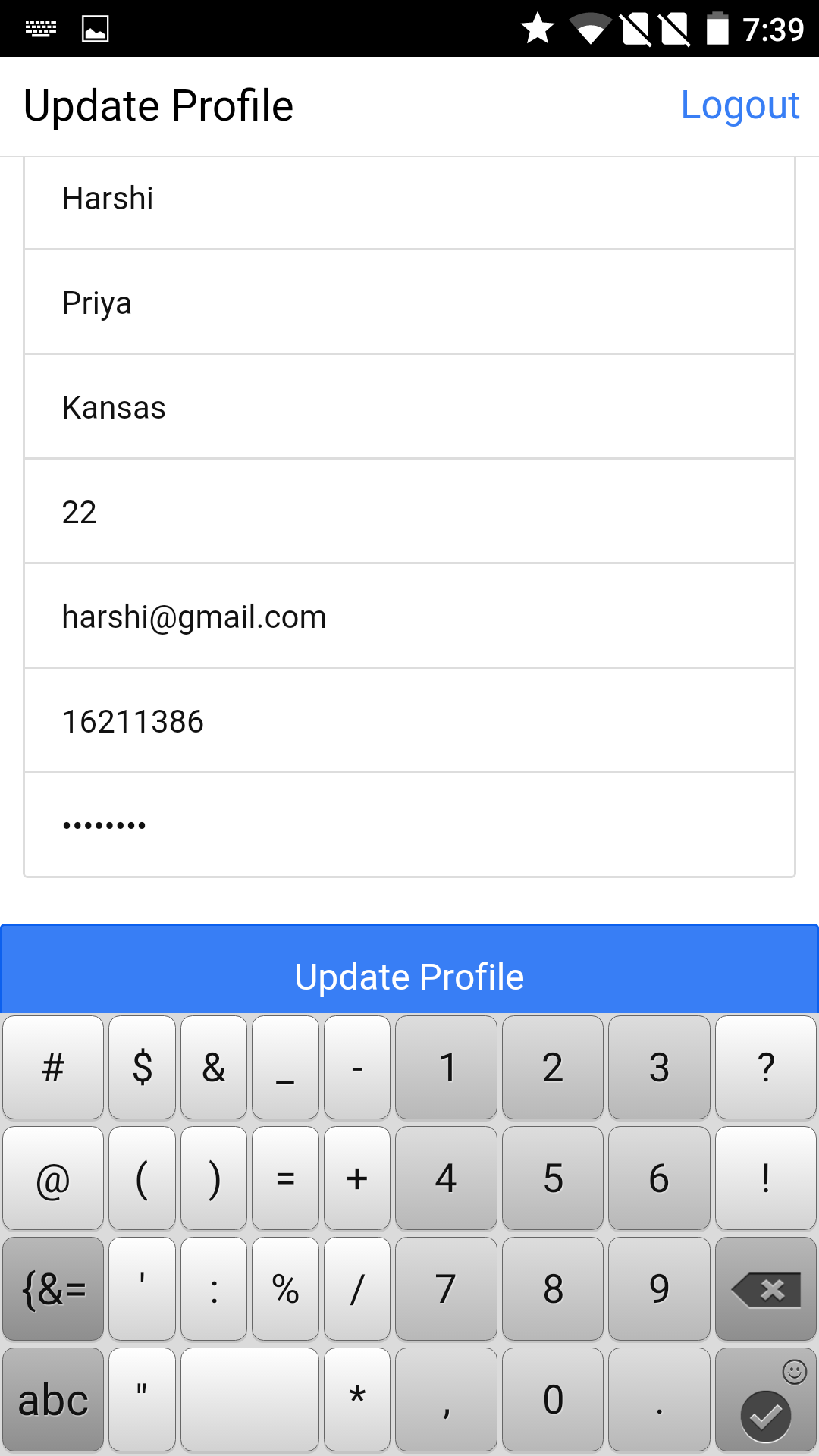
**Scan ID**: In the scan ID page, the qrcode can be scanned and read from the database to verify whether the student is enrolled in the particular college.



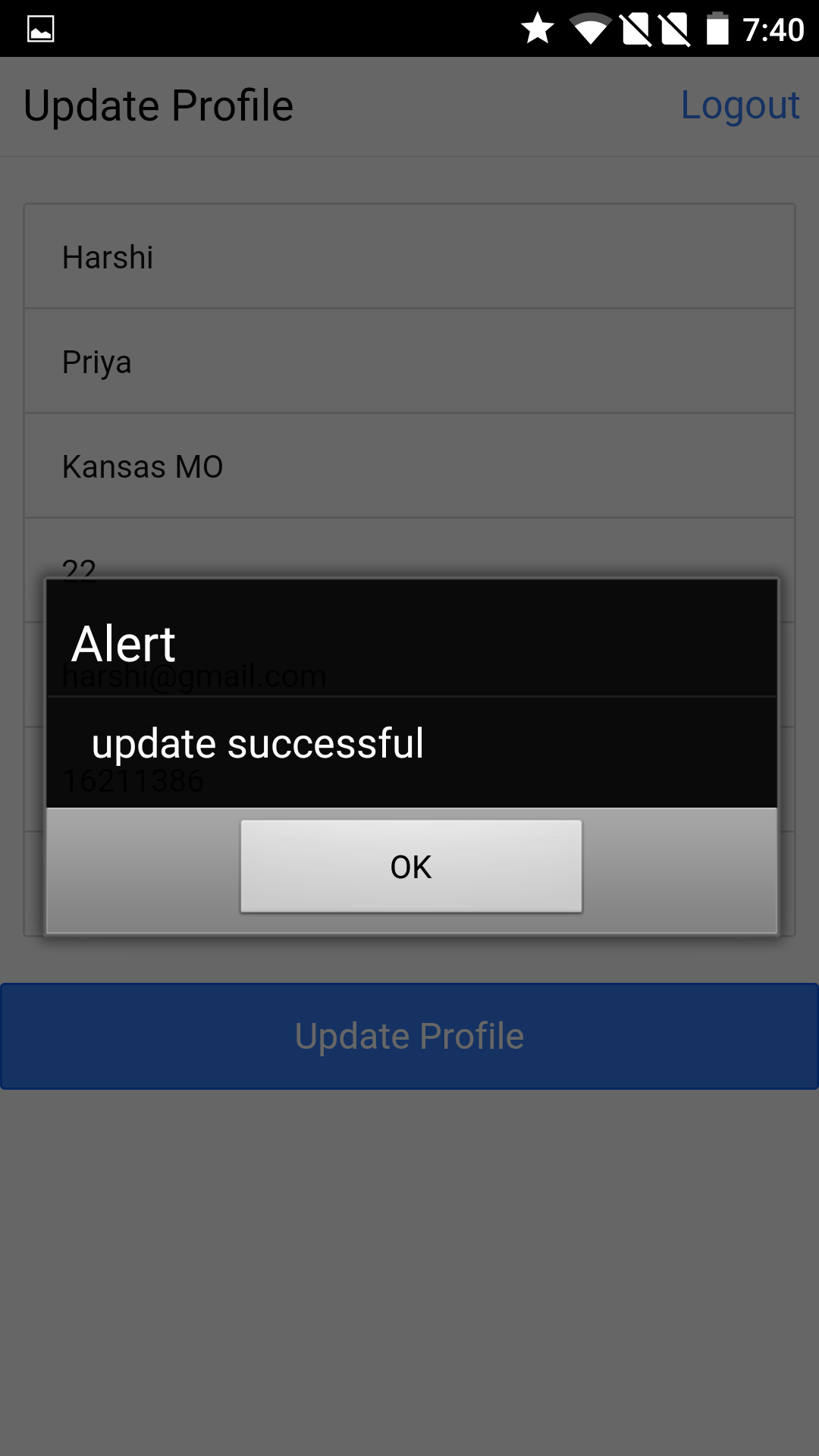
From the Scan id button to sacn Qr code

****

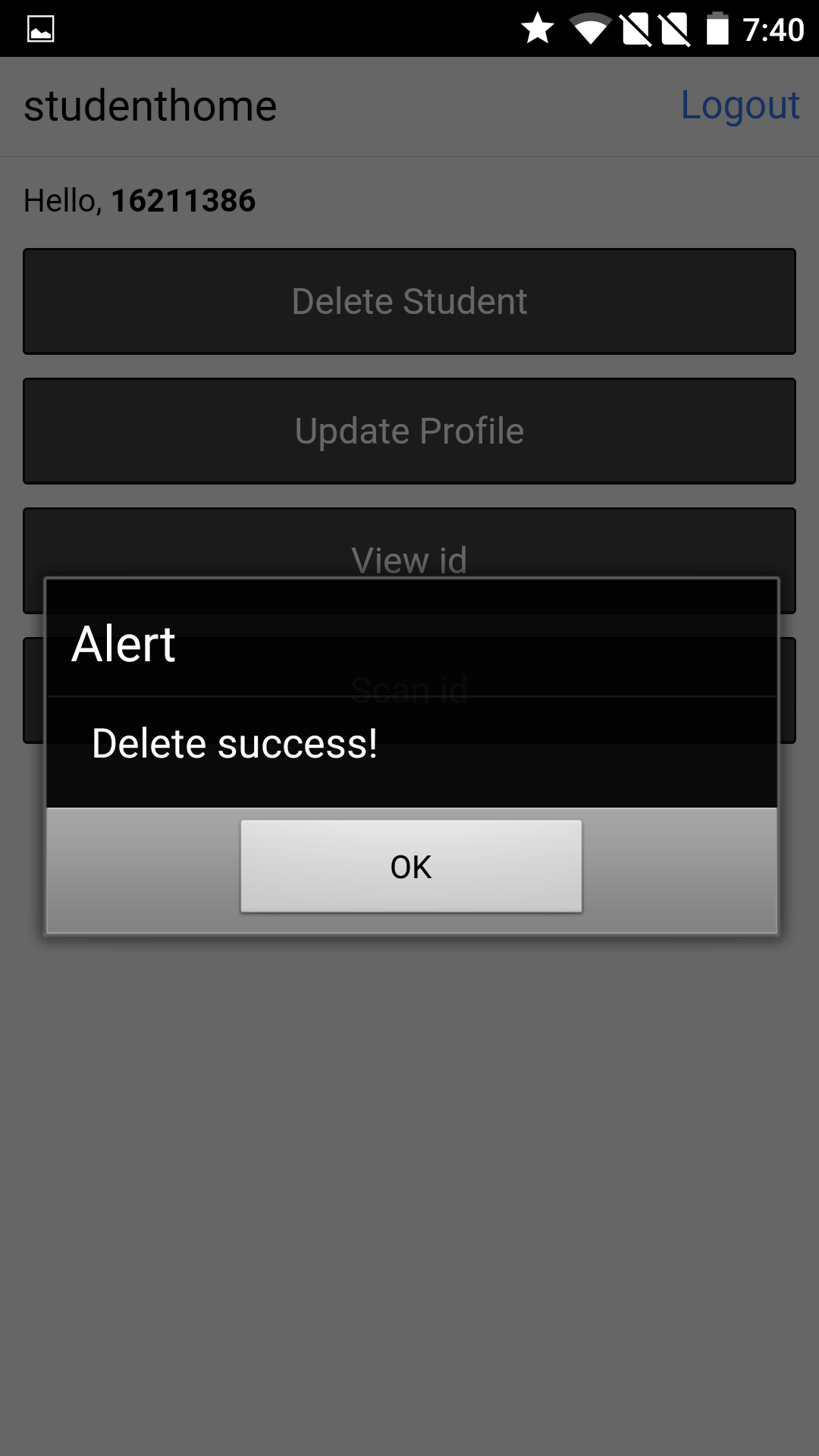
**Update Profile page:** Student can edit the information and can update student profile on click of update profile button.



In this page we can see that the student profile is updated



Using delete button from the home page the user can delete his account



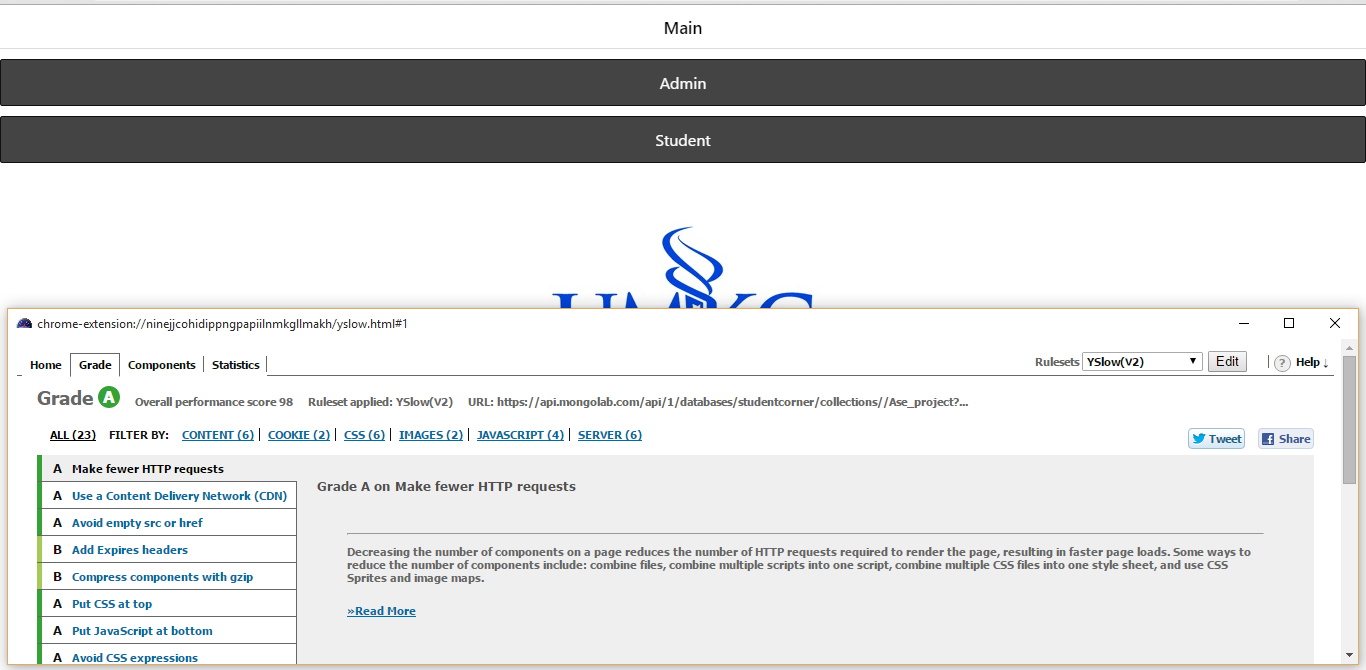
**IBM Bluemix URL:** <http://bluemixdeploymentharshi.mybluemix.net/>

**V.Testing :**

Unit testing:

Testing angularjs application Student e-ID, tools used are karma and jasmine.

Performance testing:



**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 is

[https://api.qrserver.com/v1/create-qr-code/?data="+$scope.username+"&amp;size=500x500](https://api.qrserver.com/v1/create-qr-code/?data=%22+$scope.username+%22&amp;size=500x500)

The plugin used to scan the QR code is

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

The MongoDB 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 qrcode and scan ID page is opening to the qr-code which is to be able to scan.

**VII. Project Management:**

Work completed:

* Created designs, User interface for home page, login page and main page, view page of the mobile application.
* Validated all the screens in the mobile application.
* Created a database in mongo DB.
* Used generation of QR code API.
* Verified using Unit testing, Performance testing.
* Used a scanner plugin to scan Qr code.
* Used the services from the mongo DB
* Retrieved and displayed the student data in Student ID

Responsibilities:

Abhiram: Unit testing, creating and register student using services in the mongo DB. Used QR code API to get unique QR code for the user. Created designs, User Interface for main, login, admin home, admin access pages. Implemented singleton pattern.

Harshi Priya: Implementation of Mixin pattern. Update services using mango dB for the student and retrived and displaying student details in view id page, implemented middleware using servlets. Created designs, user interface for student view ID page.

Harsha Komalla: Implementation of factory pattern. Validations of the admin pages and implementation of delete service using mongo dB, implemented scan id plugin. Created designs, User interface for scan id, student home page, student login page.

Anusha Gadepalli: Validations for student login and student home pages, performance testing, register service of the student using mongo dB, implemented scan id plugin. Created designs, User interface for register pages, access pages, Update pages.

Work to be completed:

* + Rest of the screen for the application.
  + Reading the Qr code using the REST API.
  + Verifying the details of the user with the database after scanning the Qr code.