Wollo Kombolcha Institute of Technology

Software 4rth Year 1st Semester
Wollo University Clearance System Project

Group - 6

Group member	ID
1,Aemro Bekalu	4110/12
2,Wegene Argow	2004/12
3,Elsabet Bekele	0681/12
4,Wondwessen Befikadu	2021/12
5,Ibrahim Wondayehu	1109/12

Submitted date: 2/9/2015

Submitted to:Ms.Bezawit Endis

Table of Contents

1. Title Wollo University Clearance System

2. Problem of statement

We have developed a mobile application that aims to solve the main problem faced by students, teachers, staff members, and other users of the university. This problem is related to the inconvenience and time-consuming process of checking individuals' innocence when they enter or leave the university premises. The current system is inefficient and often leads to long queues, which is a waste of time and manpower. Our mobile application offers a solution to this problem by streamlining the clearance process and making it faster and more convenient for everyone involved.

3. The Project significance

The significance of this project lies in the fact that the university clearance system is a crucial aspect of maintaining the safety and security of the university premises. By preventing theft and ensuring that only authorized individuals are allowed on the premises, we can protect the property and wealth of the university. Moreover, the system also protects the personal belongings of students, such as their laptops and clothes, which can have a significant impact on their psychological and mental health if stolen. Overall, our project helps create a safer and more secure environment for everyone at the university.

To achieve our goal, we have identified the functional requirements that the user can interact with when using the mobile application system. These requirements include the ability to apply for clearance, access the system, and receive clearance in a timely and efficient manner. By providing these functional requirements, we aim to create a user-friendly and intuitive application that makes the clearance process as seamless as possible for the users.

4. Functional Requirement

4.1. FNRQ1: Login page

The LOGIN functional requirement in the provided code is to allow users to login to the application using their email and password.

When the login button is clicked, the code checks if all the required fields (email, password, name, and id) are filled. If any of the fields are empty, the code displays a message asking the user to fill all the fields.

If all the fields are filled, the code uses Firebase Authentication to authenticate the user's email and password. If the authentication is successful, the code retrieves the user's data from Firebase Realtime Database using the user's unique ID (uid).

The code checks the user's status (student, staff, or admin) from the retrieved data and compares it with the selected user type from the spinner (also student, staff, or admin). If the user is authorized to access the selected account type, the code redirects the user to the respective activity (student, staff, or admin). If the user is not authorized, the code displays a message informing the user that they are not authorized to access the selected account type.

If the authentication is not successful, the code displays a message informing the user that the email or password is invalid.

4.2. FNRQ2: Register Page

The functional requirements for this registration component include the following:

User ID: The user must provide a unique ID to register. Password: The user must provide a password to register. The password must be at least 6 characters long.

Name: The user must provide their name to register.

Email: The user must provide a valid email address to register.

Department: The user must select a department from a drop-down list to register.

Status: The user must select their status from a drop-down list to register.

Computer Number: The user must provide the serial number of their computer to register.

Validation: All fields must be filled out before the user can register. The user ID must be unique.

Authentication: The user's email address and password will be used to authenticate them in the future.

Storage: The user's registration information will be stored in a Realtime Database using Firebase.

Success/Failure Message: The user will be notified if their registration was successful or not.

Overall, this component provides a way for users to register with the app and save their information to a database, which can be used for authentication and other purposes.

4.3. FNRQ3: The admin make the clearance of the request sent

Display a layout file (activity_admin.xml) using the setContentView() method.

Initialize the FirebaseAuth and FirebaseFirestore objects for Firebase authentication and database access respectively.

Set click listeners on three buttons (CRUDEOP, staffreqest, and logout) to perform certain actions when clicked.

Implement CRUD (Create, Read, Update, and Delete) operations on some data using another activity named 'admincrudeoperation'.

Display information about staff requests in another activity named 'staffreqestinformation'.

Initialize a TableLayout object and populate it with data from a Firebase Realtime Database.

Update data in the database when a user clicks on a specific cell in the TableLayout.

Overall, the 'admin' class provides a user interface for the admin user to perform CRUD operations on some data, view staff requests, and manage the data displayed in a TableLayout object.

4.4. FNRQ4: CRUD Operation with both the student and staff

CRUD (Create, Read, Update, and Delete) functionality for a Firebase Realtime Database that stores information about computers. Here are the functional requirements of the code:

Displaying the App Interface

The app must display a user interface that allows the user to interact with it. The user interface must include EditText fields to enter computer ID, full name, and QR code data, and buttons to find, update, delete, and go back.

Finding Data

The user must be able to search for data in the Firebase Realtime Database by entering the computer ID in the EditText field and clicking on the "Find" button. If the data exists in the database, it must be displayed in the EditText fields for full name and QR code data.

Updating Data

The user must be able to update the data in the Firebase Realtime Database by entering the computer ID in the EditText field, updating the values for full name and QR code data in the corresponding EditText fields, and clicking on the "Update" button. The updated data must be stored in the database.

Deleting Data

The user must be able to delete the data in the Firebase Realtime Database by entering the computer ID in the EditText field and clicking on the "Delete" button. The deleted data must be removed from the database, and the EditText fields for full name and QR code data must be cleared.

Navigation

The user must be able to navigate back to the previous screen by clicking on the "Back" button.

4.5. FNRQ5: Register the computer with computer id and the qrCode Value by admin

QR Code Scanner:

The QR code scanner is provided by the ZXing library. When launched, it uses the device's camera to scan a QR code. When the scanner successfully reads a QR code, it returns the scanned data to the app. The app then stops the camera, saves the scanned data along with the ID entered by the user, and refreshes the activity.

Saving Data to Firebase:

The app uses the Firebase Realtime Database to save user data. When the scanned data is received, it is saved as a new entry in the "computerinfo" node of the database. The ID entered by the user is used as the key for the entry, and the scanned data is stored as the value.

Refresh Button:

The app provides a refresh button that allows the user to restart the activity. When clicked, the current activity is finished and a new instance of the activity is created.

Back Button:

The app provides a back button that allows the user to return to the previous activity. When clicked, an intent is created to launch the "admin" activity, and the current activity is finished.

4.6. FNRQ6 Student Send the Request to the admin

TextView: crudsop The user can click on this text view to go to the student crude operation activity.

Button: logout When the user clicks this button, they will be logged out of the app and redirected to the main activity.

Button: checkcomp Clicking on this button will take the user to the "let me check" activity.

EditText: myidtxt The user can enter their ID number in this text field to search for their information in the database.

EditText: myinfotxtedy The user can view their personal information in this text field after searching for their ID.

EditText: dormInfoEditText The user can enter information about the dormitory they want to request in this text field.

EditText: libraInfoEditText the user can enter information about the library they want to request in this text field.

EditText: caffeInfoEditText The user can enter information about the cafe they want to request in this text field.

EditText: securInfoEditText the user can enter information about the security they want to request in this text field.

EditText: idInfoEditText the user can enter their ID number in this text field to make a request for facilities.

EditText: blockInfoEditText the user can enter the block number in this text field to make a request for facilities.

EditText: dormNumInfoEditText the user can enter the dormitory number in this text field to make a request for facilities.

Button: requestButton when the user clicks this button, the information they entered in the above text fields will be stored in the Firebase database as a new request.

4.7. FNRQ7: Staff Send the Request To The admin

Request submission: The app allows staff members to submit requests for equipment by filling in the relevant fields (ID, Full Name, Status, Projector, and Security). Once all fields are filled in, the staff member can click on the "Request" button to submit the request. The app checks that the ID field is not empty before submitting the request. If the ID field is empty, the app displays an error message.

Request search: The app allows staff members to search for their requests by entering their ID in the "ID" field and clicking on the "Search Me" button. The app searches the Firebase Realtime Database for the staff member's ID and displays the request information (ID, Full Name, Status, Projector, Security) if a request with that ID is found. If no request with that ID is found, the app displays an error message.

Logout: The app allows staff members to log out by clicking on the "Logout" button. When the staff member logs out, the app displays a success message and returns to the login screen.

Crude operation: The app allows staff members to perform CRUD (Create, Read, Update, and Delete) operations on the requests in the Firebase Realtime Database by clicking on the "Crude Operation" button. This button takes the staff member to a new activity where they can perform the CRUD operations.

Navigation: The app allows staff members to navigate to other activities by clicking on the relevant buttons. For example, the staff member can click on the "Check Computer" button to navigate to the staff computer activity.

5. None Functional Requirement

Security: The university clearance system must be secure and protect the personal information of students, teachers, and staff members. It should use encryption and other security measures to prevent unauthorized access or data breaches.

Performance: The mobile application must perform efficiently and respond quickly to user requests. It should be able to handle a large number of requests at once without slowing down or crashing.

Usability: The mobile application should be user-friendly and easy to navigate. It should have clear instructions and feedback for users to understand how to use it.

Reliability: The university clearance system must be reliable and available at all times. It should not have any downtime or system failures that could cause delays or inconvenience to users.

Scalability: The university clearance system should be scalable to accommodate a growing number of users and data. It should be able to handle an increasing amount of data without compromising performance or security.

Speed: The mobile application must be fast and responsive, allowing users to access their clearance information quickly and easily.

6. Conclusion

Mobile application developed by the project team aims to solve the problem of long wait times and inefficient clearance processes at university gateways. The application provides a convenient and efficient way for students, teachers, staff members, and other university members to access and exit the university premises while also ensuring that security checks are carried out in a timely and effective manner.

The significance of the project lies in its ability to improve the overall clearance system of the university, which not only helps to prevent theft and cheating but also promotes the mental and

psychological well-being of students by reducing the stress and anxiety associated with long wait times and inefficient clearance procedures.

The project team has identified and incorporated various functional requirements to ensure that the application meets the needs of its users, such as the ability to interact with the system, apply for access, and access the application from a mobile device. Additionally, non-functional requirements, including speed, security, and performance, have been taken into consideration to ensure the application's overall effectiveness.

Overall, the mobile application developed by the project team has the potential to significantly improve the university's clearance system, providing a more efficient, convenient, and secure way for students, teachers, staff members, and other university members to access and exit the premises.