MINI PROJECT
(2021-22)

Build and Deploy an Application for Laundry (Glavage).

MID-TERM REPORT

Institute of Engineering & Technology

Submitted by:

Yashraj Verma
(181500832**)**
Danish Ali
(181500200**)**
Shraddha Gupta
(181500686**)**

Supervised By:

Mrs. Harvindar Kaur
Assistant Professor
Department of Computer Engineering & Applications

Contents

Abstract

An application for Laundry (**Glavage**) project is aimed to develop an online GLA Laundry application. The use of Laundry application is to automate the hectic task of manual dealing of the laundry system which was too faulty. Through this application whole GLA students, faculty and washer man could be connected and with the use of digital medium it is convenient to manage and look after the washing of the clothes.

In this busy world everyone wants their work to be done timely and systematically and one of the main chores is laundry. Everyone wants to wear neat and clean clothes and through this application this can be done.

This application gives every information about an individual's laundry. Every person will have his/her updated list of clothes in the app and can easily access them whenever they want.

1. Introduction:

1.1. General Introduction to the topic:

The **Glavage** is an android based application which is aimed to provide the digital platform for the GLA's laundry system which is currently being managed by the washer man itself and we are looking to automate the task of the washer man and of the students to get rid of the hectic task of managing the clothes count and which clothes they had given and check whether the cloth has lost or not.

Our app will have the login/signup feature where students can login through their university's email id and setup their account.

There will be an admin's panel for washer man to maintain the requests and clothes hostel wise and room no wise. Washer man needs to update the status of his presence in the college by just pressing a button to inform the students accordingly.

Students will receive a notification on their device 15 minutes before the time.

We present the design and implementation of a laundry management system (LMS)

used in a laundry establishment. Laundry firms are usually faced with difficulties in keeping detailed records of customers clothing; this little problem as seen

to most laundry firms is highly discouraging as customers are filled with disappointments, arising from issues such as customer clothes mix-ups and untimely retrieval of clothes. The aim of this application is to determine the number of clothes collected, in relation to their owners, as this also helps the users fix a date for the collection of their clothes. Also customer's information is secured, as a specific id is allocated per registration to avoid contrasting information.

1.2. Area of Computer Science:

Using various technological assets of computer science field we will develop this application.
Areas of computer science that helped us to build our project are:
Cloud computing.
Android Development.
Device Virtualization.
NoSQL Database.
Client Server Connection.
Push Notifications.

1.3. Hardware and Software Requirements:

❑ Hardware Requirements:

- ❑ **For Development:** A personal computer for developing the project with a minimum i5 core processor or AMD's Ryzen processor would be fine and 8 GB of RAM.
- ❑ **For testing purposes:** Smartphones built on Android OS or Virtual Device like Android Emulator on IDE itself, JUnit is a unit testing framework for the Java programming language.

❑ Software Requirements:

- ❑ An Integrated Development Environment (IDE) i.e., Android Studio to develop the project
- ❑ Java Development kit (Jdk13)
- ❑ Android SDK and NDK with minimum 21 of API level
- ❑ Cloud Services like Google Cloud Platform (GCP) (Google's Firebase to support the Backend services) and BAAS (Backend as a service).

Problem Definition

Laundry firms currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the laundry management infrastructure. Often information (on forms) is incomplete or does not follow management standards.

Records are often lost in transit during computation requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the laundry firm data and may lead to inconsistencies in data in various data stores.

A significant part of the operation of any laundry firm involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; customer personal information and clothing records history, user information, price of delivery and retrieval period, users scheduling as regards customers details and dealings in service rendered, also our products package waiting list. All of this information must be managed in an efficient and cost wise fashion so that the organization resources may be effectively utilized.

Objective

The Laundry Management System is designed for any Laundry firm to replace their existing manual, paper based system. The new system is in form of an e-registration system to control the following; customer information, products, services, users, carts and receipt.

These services are to be provided in an efficient, cost effective manner, with the goal of reducing the delay and resources currently required for such tasks as clothes details are bounded to a particular customer with a given ID.

Since the existing system makes use of tedious administrative tasks, lots paper work and time, in which full information cannot be gotten from busy customers. The goal of the laundry management system is to provide a computerized process that is stress free, reliable and quick through the use of asp.net computer programming language and SQL database application to the users and staffs in charge of the registration of customers and laundry management processes.

HTML would be at the front-end and provide the graphical user interface that relates with the user, while the SQL database will be at the back-end to handle the data storage process.

The objective of this work is to implement a management system that will streamline registration process, reduce administrative tasks and paper work so as to improve the registration cycle process flow.

Implementation Details

Part-1:- Identifying the Technologies required in managing, making and accomplishing the goals of this Project.

Android Studio is used as an Integrated Development Environment to develop the android application for the project.
Java and XML language is used for making the application functionable and implementing the design of the android application.
Google Cloud Platform is used to provide the backend support to this application.
Google Map's SDK to locate the nearby hospitals in the locality.
Firebase SDK is integrated in the application to provide the following Backend services:-

1. Firebase Authentication.
2. Firebase Real-time Database.
3. Firebase Storage.

Part-2:- Designing the User Interface plays an important role in shaping the final project. It defines the presentation and functionality of the application.
Figma, Photoshop and Adobe XD is used in designing the User Interface of the application.

Part-3:- After Design it's time to setup the development environment to implement the designed layout and to develop the logic of the app in the development environment.
XML scripting language forms the whole user interface part and Java helps in developing the functionality of the app.

Part-4:- After setting up the development environment we will integrate the Firebase SDK in the application.
Firebase Authentication is used to integrate the login module of the app.
When a user register on the app it becomes the part of the database which forms up the whole User database of the application on the Firebase console.

All the database is managed on the Firebase console and fetched on the application to display the relevant data about the apps activities.

Data in the app is fetched in the JSON format which then is arranged and decoded to display in the app by the Firebase SDK in the application itself.

Firebase Storage service contains all the required media which will be used in the app to make it more user friendly. Media like icons, images, svg files.

Part-5:- Managing the Database to collect the data of the users, Users database. User can post the Request for the clothes on the app and the same data and activity will be uploaded on the firebase database.

Progress

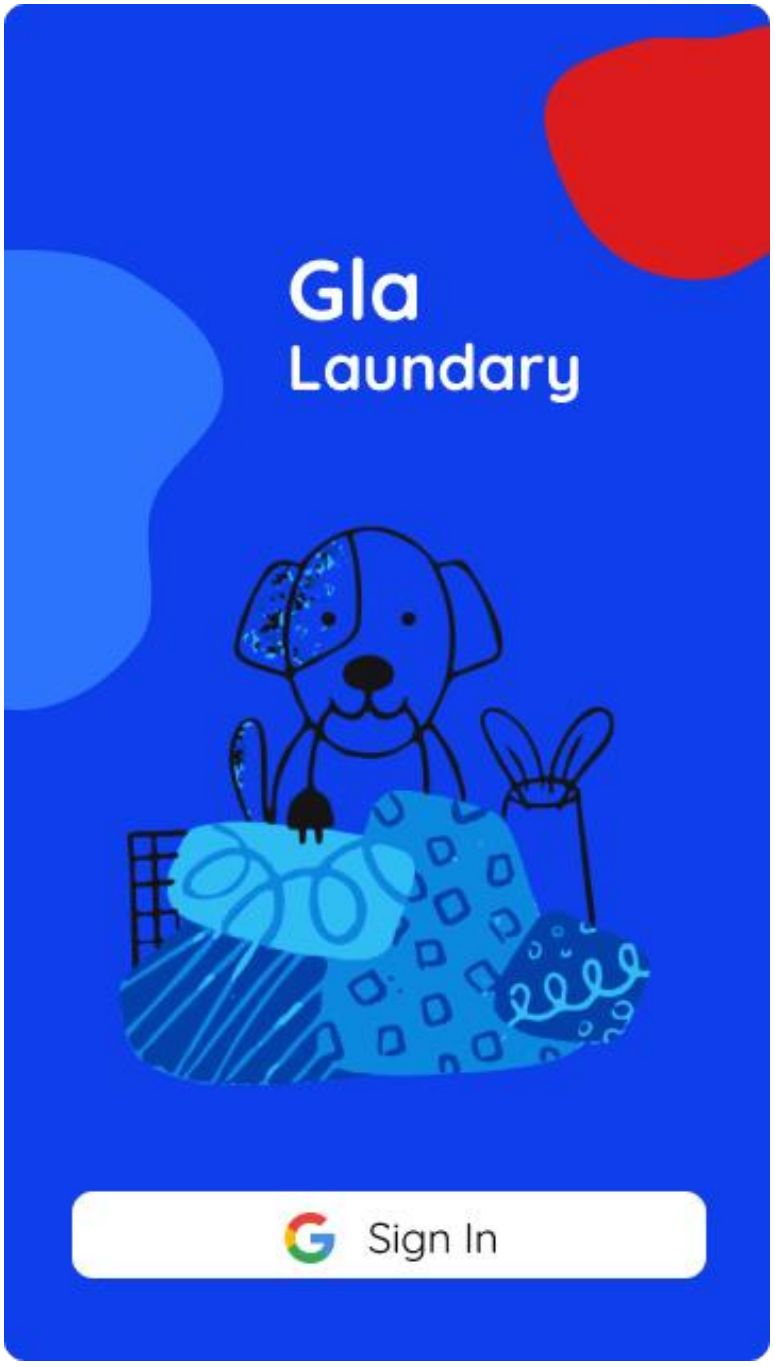
Completed:

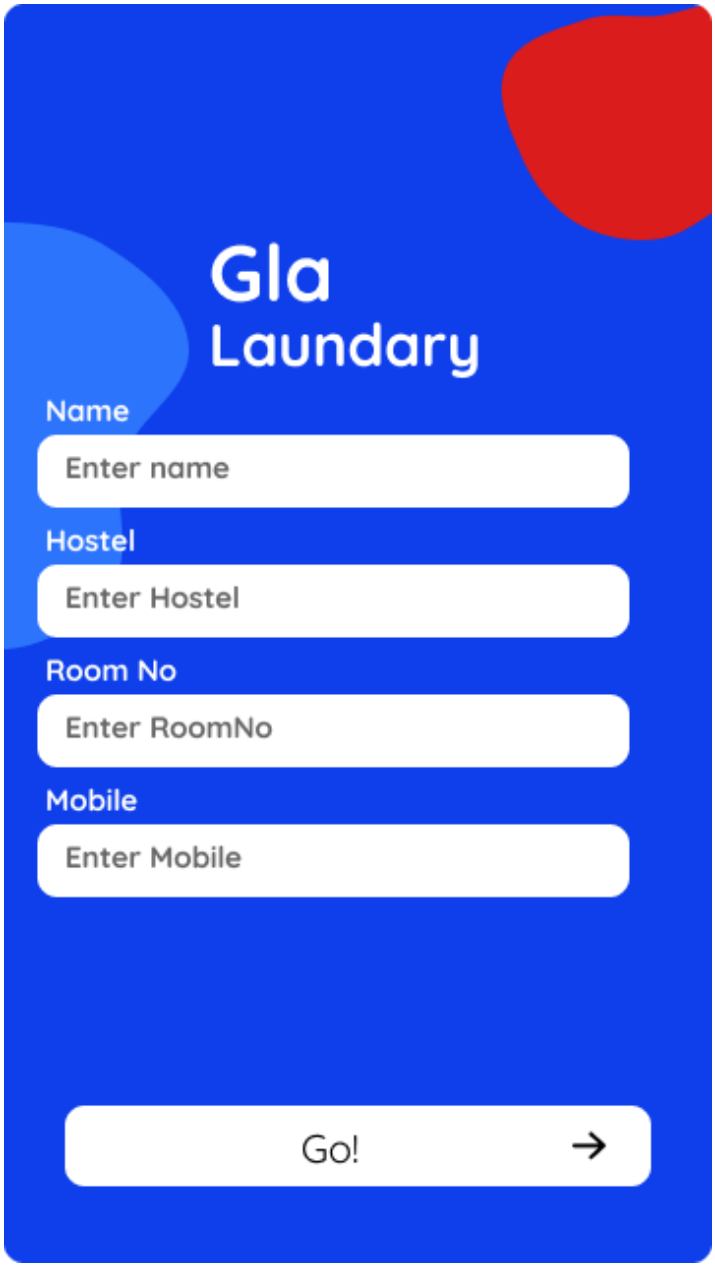
- Identifying the tech required to build the project.
- Designing of the User Interface.
- Implementing the Design layout.
- Integration of Firebase SDK.
- Login Module to Register user.
- Formation of database in Firebase Console.
- Details Module in app to take the details of the user to upload on database.

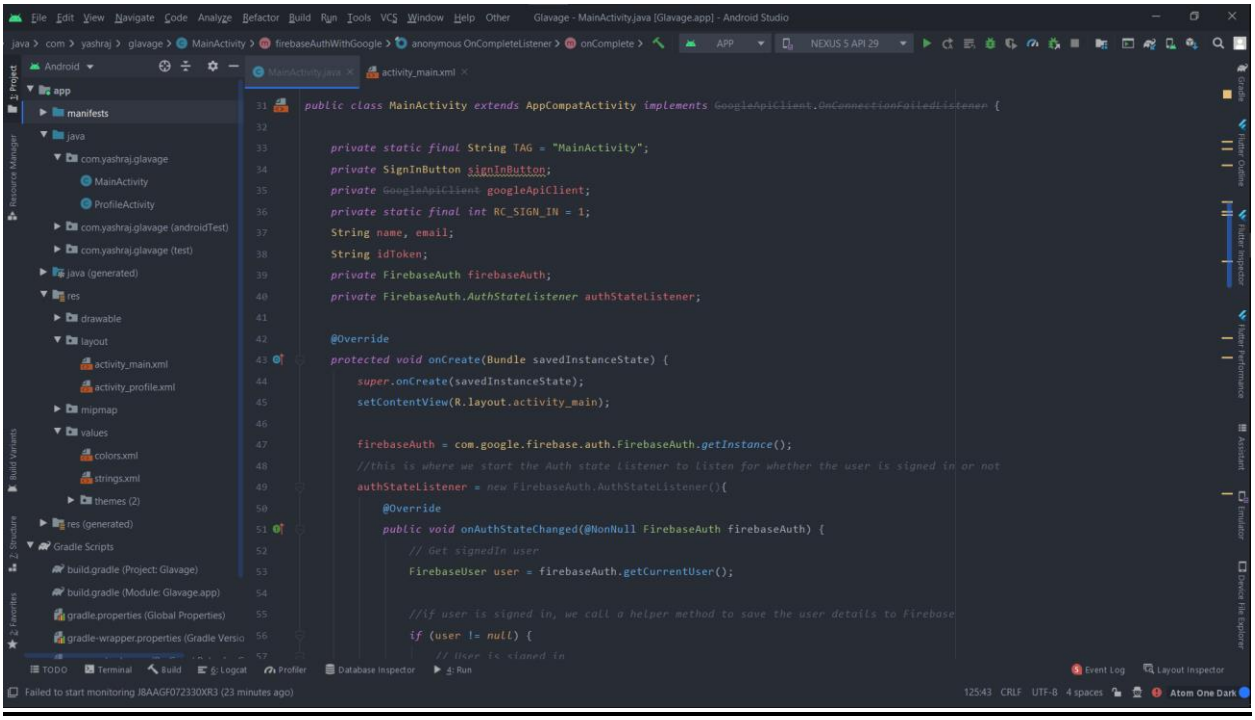
Not Completed:-

- Push Notifications.
- Payment module.

Screenshots




Gla
Laundary
Name
Enter name
Hostel
Enter Hostel
Room No
Enter RoomNo
Mobile
Enter Mobile
Go!


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Other
Glavage - MainActivity.java [Glavage.app] - Android Studio
java > com > yashraj > glavage > MainActivity > firebaseAuthWithGoogle > anonymous OnCompleteListener > onComplete >
APP
NEXUS 5 API 29
Android
MainActivity.java
activity_main.xml
app
manifests
java
com.yashraj.glavage
MainActivity
ProfileActivity
com.yashraj.glavage (androidTest)
com.yashraj.glavage (test)
java (generated)
res
drawable
layout
activity_main.xml
activity_profile.xml
mipmap
values
colors.xml
strings.xml
themes (2)
res (generated)
Gradle Scripts
build.gradle (Project: Glavage)
build.gradle (Module: Glavage.app)
gradle.properties (Global Properties)
gradle-wrapper.properties (Gradle Versio
public class MainActivity extends AppCompatActivity implements GoogleApiClient.OnConnectionFailedListener {
private static final String TAG = "MainActivity";
private SignInButton signInButton;
private GoogleApiClient googleApiClient;
private static final int RC_SIGN_IN = 1;
String name, email;
String idToken;
private FirebaseAuth firebaseAuth;
private FirebaseAuth.AuthStateListener authStateListener;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
firebaseAuth = com.google.firebase.auth.FirebaseAuth.getInstance();
//this is where we start the Auth state listener to listen for whether the user is signed in or not
authStateListener = new FirebaseAuth.AuthStateListener(){
@Override
public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
// Get signedIn user
FirebaseUser user = firebaseAuth.getCurrentUser();
//if user is signed in, we call a helper method to save the user details to Firebase
if (user != null) {
// User is signed in
TODO
Terminal
Build
6: Logcat
Profiler
Database Inspector
4: Run
Event Log
Layout Inspector
Failed to start monitoring J8AAGF072330XR3 (23 minutes ago)
125:43
CRLF
UTF-8
4 spaces
Atom One Dark

References

https://firebase.google.com/

https://code.visualstudio.com/