VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590010

MOBILE APPLICATION DEVELOPMENT MINI PROJECT
ON

"LOST AND FOUND ITEMS"

Submitted in partial fulfillment for the requirements for the sixth semester

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

For the Academic Year 2022-2023

Submitted by:

KOUSTAV DAS	**1MV20CS051**
KUMAR KETAK	**1MV20CS052**
MAYANK SHARMA	**1MV20CS058**

Under the guidance of:
Assistant Professor
Ms. DHIVYA V
Department of CSE

Sir M. Visvesvaraya Institute of Technology, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
HUNASAMARANAHALLI, BENGALURU-562157

M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

Krishnadevaraya Nagar, International Airport Road,
Hunasmaranahalli, Bengaluru – 562157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

It is certified that the **PROJECT** on "LOST AND FOUND ITEMS" is carried out by KOUSTAV DAS (1MV20CS051), KUMAR KETAK(1MV20CS052), and MAYANK SHARMA (1MV20CS058) bonafide students of Sir M Visvesvaraya Institute of Technology in partial fulfillment for the 5th semester for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2022-2023. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

Name & Signature Of Guide	Name & Signature Of HOD	Name & Signature Of Principal
Ms. Dhivya V	**Dr. Anitha T N**	**Dr. Rakesh S.G**
Assistant Professor	HOD, Dept. of CSE,	Principal
Sir MVIT Bengaluru -562157	Sir MVIT Bengaluru -562157	Sir MVIT Bengaluru-562157

External Examination:

Name of Examiner	Signature with Date
1)	
2)	

DECLARATION

We hereby declare that the entire application work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru

Date:

SIGNATURE OF STUDENTS

KOUSTAV DAS(1MV20CS051)

KUMAR KETAK(1MV20CS052)

MAYANK SHARMA(1MV20CS058)

ACKNOWLEDGMENT

It gives us immense pleasure to express our sincere gratitude to the management of Sir M. Visvesvaraya Institute of Technology, Bengaluru for providing the opportunity and the resources to accomplish our project work in their premises.

On the path of learning, the presence of an experienced guide is indispensable, and we would like to thank our guide **Ms. DHIVYA V**, Asst. Professor Dept. of CSE, for her invaluable help and guidance.

Heartfelt and sincere thanks to **Dr. Anitha T N**, HOD, Dept. of CSE, for his suggestions, constant support and encouragement.

We would also like to convey our regards to **Dr. Rakesh S G**, Principal, Sir. MVIT for providing us with the infrastructure and facilities needed to develop our project.

We would also like to thank the staff of Department of Computer Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would like to thank all our friends for their help and suggestions without which completing this project would not have been possible.

ABSTRACT

Lost and Found app addresses the pressing need for an Android application to tackle the prevalent issue of lost and found items within our college campus. Currently, the absence of a centralized system leads to a lack of transparency and difficulties in reuniting lost items with their owners. Valuable possessions frequently go missing, causing inconvenience and frustration for the college community. The absence of a streamlined platform for reporting and tracking lost items hampers the overall campus experience and poses challenges to the administration's efforts to recover and return belongings. Therefore, the development of an innovative lost and found application is imperative to provide a user-friendly interface, simplify the reporting process, and enhance the chances of successful item retrieval, fostering a more secure and supportive college environment.

To address the a fore mentioned challenges, our proposed solution is the development and implementation of an Android application dedicated to lost and found items on our college campus. The application will feature a centralized system that enables students and staff to report lost items and track their status seamlessly. The user-friendly interface will ensure ease of use for individuals with varying levels of technological proficiency, promoting widespread adoption. The application will employ various features to enhance the retrieval process, such as item categorization, matching algorithms, and coordination with campus authorities. Additionally, notifications and alerts will be integrated to provide timely updates and communication to users regarding the status of their lost items. To ensure the application's success, a comprehensive marketing strategy will be implemented to raise awareness and encourage active participation within the college community. Overall, this Android application aims to streamline the lost and found process, facilitating the efficient recovery and return of belongings, thereby fostering a more secure and supportive college campus environment.

TABLE CONTENTS

Chapter 1: INTRODUCTION

Android Studio is an integrated development environment (IDE) used for creating Android applications. It includes features such as a code editor, layout editor, build system, emulator for testing, profiling tools, and version control integration. Android Studio simplifies the development process and provides resources for learning and troubleshooting.

Android Studio utilizes the Gradle build system to manage dependencies, compile code, and package your app into an APK (Android Package) file. Gradle is highly flexible and allows you to customize the build process according to your app's requirements. You can add external libraries, resources, and configurations through the Gradle script.

Android Studio includes a built-in Android Emulator that allows you to test your app on various virtual devices with different Android versions and configurations. Additionally, you can connect physical Android devices for testing purposes. Android Studio provides debugging tools to identify and fix issues during development.

Android Studio has a vast community of developers, and there are numerous online resources, tutorials, and documentation available to assist you in learning and troubleshooting. The official Android Developer website provides detailed guides, sample code, and best practices for Android app development.

Android Studio offers a rich and productive environment for developing Android applications. It streamlines the development process and provides essential tools and features to create high-quality, feature-rich apps for the Android platform.

Chapter 2 : PROBLEM STATEMENT

The Lost and Found app revolves around the need for an efficient lost and found application within a college campus. Currently, the absence of a centralized system leads to a lack of transparency and difficulties in reuniting lost items with their owners. With numerous students and a bustling environment, valuable possessions frequently go missing, causing inconvenience and frustration. The absence of a streamlined platform for reporting and tracking lost items hampers the overall campus experience and poses challenges to the administration's efforts to recover and return belongings. Therefore, an innovative lost and found application is required to provide a user- friendly interface, simplify the reporting process, and enhance the chances of successful item retrieval, fostering a more secure and supportive college community. Lost and found app showcase application should address the following challenges:

1. Lack of Centralized System: The application needs to overcome the challenge of the absence of a centralized system for lost and found items on the college campus. It should provide a unified platform where students and staff can report lost items and track their status.

2. Limited Awareness: The application should address the challenge of limited awareness among the college community about the existence and usage of such a platform. It should implement effective marketing strategies to promote the application and encourage its adoption.

3. User-Friendly Interface: The application should overcome the challenge of designing a user-friendly interface that is intuitive and easy to navigate. It should be accessible to individuals with varying levels of technological proficiency to ensure widespread use and participation.

4. Timely Updates and Communication: The application needs to address the challenge of providing timely updates and communication to users regarding the status of their lost items. It should incorporate features such as notifications and alerts to keep users informed throughout the process.

Chapter 3 : REQUIREMENT SPECIFICATION

System Requirements

- 64-bit Microsoft 8/10 or Any 64-bit Linux distribution that supports Gnome, KDE.
- X86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor.
- 8GB RAM or more
- 8GB of available disk space minimum (IDE + Android SDK + AndroidEmulator)
- 1280 x 800 minimum screen resolution.

Functional Requirements

- Implementation of Clipboard framework, for copying and pasting of string.
- Activity with two edit text control and twobuttons to trigger the copy and paste functionality.

Software Requirements

- Android studio version 21
- Android SDK version 21

Chapter 4 : Analysis

The ubiquity of smartphones and their indispensable role in our daily lives has made them powerful tools for addressing various needs, including the retrieval of lost items. Many educational institutions have recognized the potential of mobile applications to facilitate lost and found processes on their campuses. This article aims to analyze a specific Android application project developed for a campus environment, focusing on its features, usability, effectiveness, and potential for improvement.

Features and Functionality:

a. Item Reporting: Users can submit detailed descriptions of lost items, additional identifying information.

b. Item Search: The application provides a search feature that allows users to browse through the database of lost items based on various filters such as category or date.

c. Claiming Process: Once a user identifies a lost item as their own, they can initiate a claiming process through the application, facilitating communication with the item's current holder or the campus lost and found department.

Effectiveness and Potential Improvements:

a. Integration with Campus Security: The application can be further enhanced by integrating it with the campus security system, allowing for real-time notifications and improved coordination.

b. Enhanced Verification: Implementing a verification process for claimants can help ensure that items are returned to their rightful owners.

c. User Feedback and Rating System: Incorporating a feedback and rating system can encourage user participation and provide valuable insights for further improvements.

d. Expanding Outreach: Promoting the application to raise awareness among the campus community and encouraging its use can increase the chances of lost items being found and returned.

Chapter 5 : Design

In the design part of our project report, we have provided details about the overall design of the Mental Health Support app. Here are some key points to include:

1. User Interface Design:

- Discuss the design philosophy and principles followed in creating the app's user interface.

- Describe the visual aesthetics, color schemes, typography, and overall style used to create a cohesive and visually appealing design.

- Explain how the UI design aligns with the app's target audience and purpose, considering factors such as user preferences and accessibility.

2. Navigation and Information Architecture:

- Explain the app's navigation structure and how users move between screens and access different features.

- Discuss the use of navigation components such as bottom navigation, side drawer, or tabs to facilitate easy navigation.

- Describe the organization and hierarchy of information within the app, ensuring that the user can intuitively find and access the desired content.

3. Screen Layout and Components:

- Discuss the layout patterns used in the app, such as linear, grid, or relative layouts, and how they contribute to the overall user experience.

- Describe the placement and arrangement of UI components on each screen, ensuring logical grouping and intuitive interaction.

- Highlight any unique or custom-designed components that enhance the app's usability and visual appeal.

4. User Input:

- Explain how the app captures user input through various interactive elements like buttons.

consistent experience across a wide range of devices.

Chapter 6: IMPLEMENTATION

The Lost and Found App is implemented using Java programming language for the logic and XML for designing the user interface. The Java code handles user input validation, all inputs details, and generates the amortization schedule. The XML files define the layout and visual elements of the app's screens.

1. Requirement Definition :

Start by identifying the specific requirements for your Lost and Found app. Consider features such as user registration, item reporting, search functionality, communication channels, and integration with campus security.

2. User Interface :

Create intuitive and user-friendly interfaces to ensure ease of use. Focus on simplicity and clarity, allowing users to report lost items quickly and easily search for found items.

3. Development Tools selection :

For Android app development, you can use popular frameworks like Android Studio and Java or Kotlin as the programming language. Android Studio provides a comprehensive development environment with built-in tools for designing interfaces, writing code, and testing the app on emulators or physical devices.

4. Development of Backend :

Design and implement a robust backend system to support your app's functionality. Set up a database to store information about lost and found items, user profiles, and communication logs. Use technologies like Firebase, MySQL, or MongoDB to manage the data effectively.

5. Item Reporting:

Enable users to report lost items by providing details such as item description, location, and date.

Chapter 7: CODE

XML-CODE activity main:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.co
m/apk/res/android"
   xmlns:tools="http://schemas.andr
   oid.com/tools"
   android:layout_width="match_p
   arent"
   android:layout_height="match_p
   arent"
   android:orientation="vertical"
   android:padding="20dp"
   tools:context=".MainActivity">

   <Button
      android:id="@+id/btnCreateA
      dvert"
      android:layout_width="match_
      parent"
      android:layout_height="wrap_
      content"
      android:layout_centerInParent
      ="true"
      android:background="@drawa
      ble/rounded_button"
      android:text="Create a New
      Advert" />
```

```
<Button
  android:id="@+id/btnShowIte
  ms"
  android:layout_below="@id/bt
  nCreateAdvert"
  android:layout_width="match_
  parent"
  android:layout_height="wrap_
  content"
  android:layout_centerVertical=
  "true"
  android:layout_centerHorizont
  al="true"
  android:layout_marginTop="2
  0dp"
  android:background="@drawa
  ble/rounded_button"
  android:text="Show All Lost
  and Found Items" />
```

XML CODE for view details:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/
apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_widt
  h="match_parent"
  android:layout_heig
  ht="match_parent"
  android:orientation=
  "vertical"
  tools:context=".Vie
  wItems">
```

```
<TextView
  android:layout_width="ma
  tch_parent"
  android:layout_height="wr
  ap_content"
  android:text="LOST
  &amp;amp; FOUND ITEMS"
  android:textAlignment="c
  enter"
  android:textSize="25dp"
  />

<androidx.recyclerview.widget.RecyclerView
  android:layout_marginTop="10dp"
  android:id="@+id/recyclerView"
  android:layout_width="match_parent"
  android:layout_height="match_parent" />

</LinearLayout>
```

JAVA CODE (Main Activity):

```
package com.ashok.lostfound;
import
androidx.appcompat.app.AppCo
mpatActivity;import
android.content.Intent;
import
android.os.B
undle; import
android.view.
View; import
android.widg
et.Button;
```

```
import
android.widg
et.Toast;

public class MainActivity extends
  AppCompatActivity {
private Button btnCreateAdvert;
  private
  Button
  btnShowItem
  s;@Override
  protected void onCreate(Bundle
    savedInstanceState) {
    super.onCreate(savedInstanceS
    tate);
    setContentView(R.layout.activi
    ty_main);

    btnCreateAdvert =
    findViewById(R.id.btnCreateAdvert);
    btnShowItems =
    findViewById(R.id.btnShowItems);
    btnCreateAdvert.setOnClickListener(new
    View.OnClickListener() {
      @Override
      public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, Details.class);
        // Pass any necessary data using intent.putExtra() if needed
        startActivity(intent);
  }
});

      btnShowItems.setOnClickListener(new
      View.OnClickListener() {@Override
      public void onClick(View v) {
```

```
Intent intent = new
    Intent(MainActivity.this,
    ViewItems.class);startActivity(intent);
  }

  });
  }
  }
```

JAVA CODE details info:

```
package com.ashok.lostfound;

import
android.app.DatePic
kerDialog;import
android.content.Inte
nt;
import
android.os
.Bundle;
import
android.vi
ew.View;
import
android.view.inputmethod.InputMetho
dManager;import
android.widget.Button;
import
android.widget.Da
tePicker; import
android.widget.Ed
itText; import
android.widget.Ra
dioButton;import
android.widget.Ra
```

```
dioGroup;import
android.widget.To
ast;
import androidx.appcompat.app.AppCompatActivity;

import
com.ashok.lostfound.database.Da
tabaseHelper;import
com.ashok.lostfound.model.Lost
Found;

import
java.text.SimpleD
ateFormat;import
java.util.Calendar;
import java.util.Locale;

public class Details extends
   AppCompatActivity {
   private RadioButton
   radioLost, radioFound;
   private Button btnSave;
   private Calendar calendar;
   private EditText User_name, User_phone,
   User_desc,btnDatePicker,User_location;private
   RadioGroup radioGroup;


   @Override
   protected void onCreate(Bundle
      savedInstanceState) {
      super.onCreate(savedInstanceSt
      ate);
      setContentView(R.layout.activit
      y_details); radioGroup =
```

```
                findViewById(R.id.radioGroup);
                radioLost =
                findViewById(R.id.radioLost);
                radioFound =
                findViewById(R.id.radioFound
                ); btnDatePicker =
                findViewById(R.id.btnDatePic
                ker);btnSave =
                findViewById(R.id.btnSave);
                calendar =
                Calendar.getInstance();
                User_name =
                findViewById(R.id.etNa
                me); User_phone =
                findViewById(R.id.etPho
                ne);
                User_location=findViewById(R.id
                .etLocation);User_desc =
                findViewById(R.id.etDescription);

                btnDatePicker.setOnClickListener(new
                  View.OnClickListener() {@Override
                  public void
                    onClick(Vi
                    ew v) {
                    hideKeybo
                    ard();

                    showDatePickerDialog();
              }
            });

                btnSave.setOnClickListener(new
                  View.OnClickListener() {
```

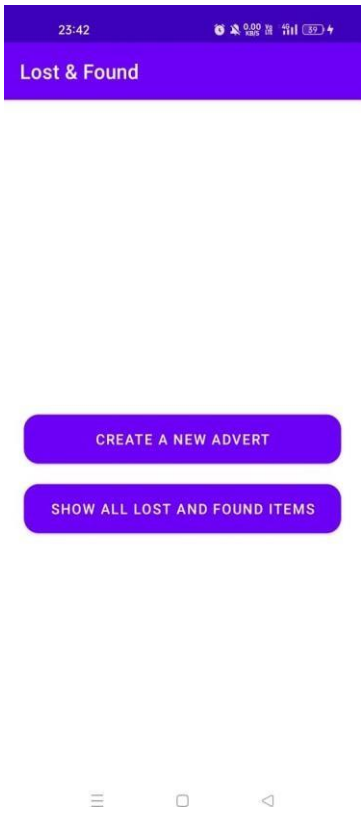
```
                @Override
                public void
                onClick(Vi
                ew v) {
                saveDetails
                ();
            }
        });
                radioGroup.setOnCheckedChangeListener((
                group, checkedId) -> {if (checkedId ==
                R.id.radioLost) {
                radioLost.set
                Checked(true
                );
                radioFound.s
                etChecked(fal
                se);
                } else if (checkedId ==
                R.id.radioFound) {
                radioLost.setChecked
                (false);
                radioFound.setCheck
                ed(true);
            }
        });
            }
            private void showDatePickerDialog() {
                DatePickerDialog datePickerDialog =
                new DatePickerDialog(this,
                new
                DatePickerDialog.OnDa
                teSetListener() {
                @Override
                public void onDateSet(DatePicker view, int
            year, int monthOfYear, intdayOfMonth) {
```

```
                    calendar.set(Calendar.YEAR, year);
                    calendar.set(Calendar.MONTH,
                    monthOfYear);
                    calendar.set(Calendar.DAY_OF_M
                    ONTH, dayOfMonth);if
                    (calendar.after(Calendar.getInstanc
                    e())) {
                      Toast.makeText(getApplicationContext(), "User
cant select FUTURE date",Toast.LENGTH_SHORT).show();
            return;
                    }
                    SimpleDateFormat dateFormat = new
SimpleDateFormat("yyyy-MMM-dd",Locale.getDefault());
                    String selectedDate =
                    dateFormat.format(calendar.getTime())
                    ;btnDatePicker.setText(selectedDate);


                  }
                },
                calendar.get(Calendar.Y
                EAR),
                calendar.get(Calendar.M
                ONTH),
                calendar.get(Calendar.D
                AY_OF_MONTH));


    datePickerDialog.show();
  }
  private void saveDetails() {
    String postType = radioLost.isChecked()
    ? "LOST" : "FOUND";String name
    =User_name.getText().toString();
    String
    phone=User_phone.getText().
    toString(); String desc =
```

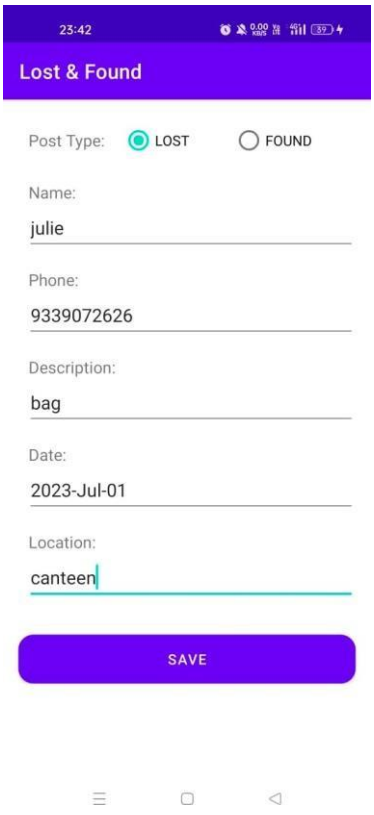
```
                User_desc.getText().toString()
                ; String date =
                btnDatePicker.getText().toStri
                ng(); String
                location=User_location.getTex
                t().toString();
                // Implementing database
                if
                  (postType.isEmpty()||date.isEmpty()
                  ){ Toast.makeText(this, "Name and
                  Date must not be empty",
            Toast.LENGTH
                  _SHORT)
                  .show();
                  return;
            }


                Long rowID= new
            DatabaseHelper(Details.this).insertData(new
            LostFound(postType,name, phone, desc,
            date,  location ));
                if (rowID != -1) {
                  Toast.makeText(Details.this,
            "Details saved successfully",
            Toast.LENGTH_SHORT).show();



                  clear();
                  Intent intent = new Intent(Details.this, MainActivity.class);
                  // Pass any necessary data using intent.putExtra() if needed
                  startActivity(intent);
                } else {
                  Toast.makeText(Details.this, "Failed to save details",
                  Toast.LENGTH_SHORT).show();
                }
```

```
        }
        private void
          clear(){
          radioLost.set
          Checked(fals
          e);
          radioFound.s
          etChecked(fal
          se);
          btnSave.setTe
          xt("");
          User_name.se
          tText("");
          User_phone.s
          etText("");
          User_location
          .setText("");
          User_desc.set
          Text("");
        }

        private void hideKeyboard() {
          InputMethodManager imm =
(InputMethodManager)
getSystemService(this.INPUT_MET
HOD_SERVICE);
          View focusedView =
          getCurrentFocus();if
          (focusedView !=
          null) {
            imm.hideSoftInputFromWindow(focusedView.getWindowToken(),
            0);
          }
        }
}
```

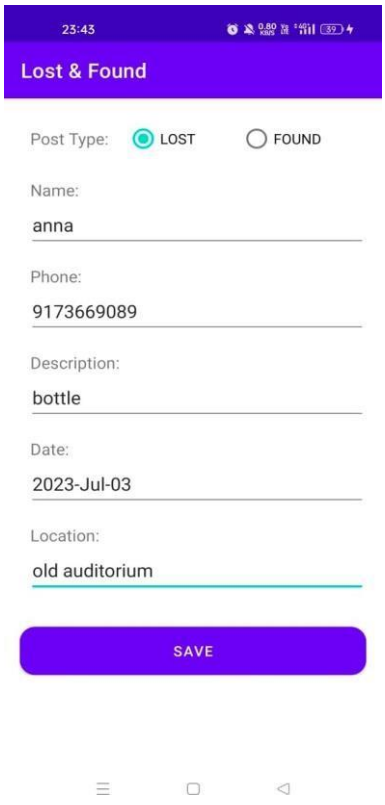
Chapter 8: SCREENSHOTS



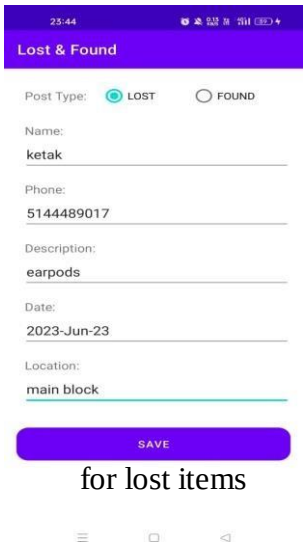
7.1 Opening Activity



7.2 Activity for lost items

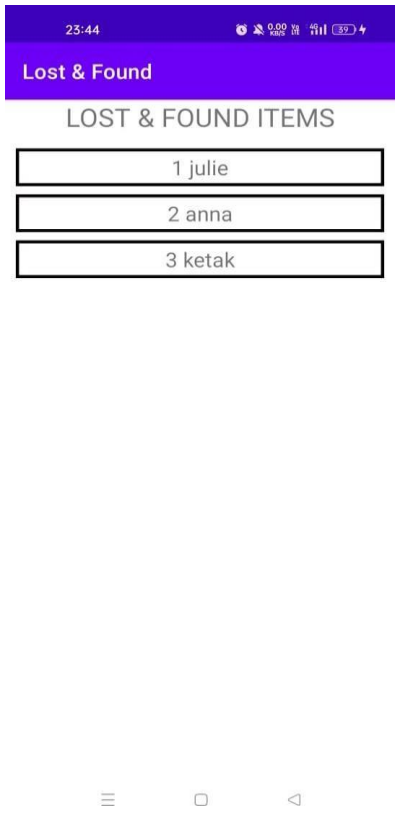


7.3 Activity for lost items

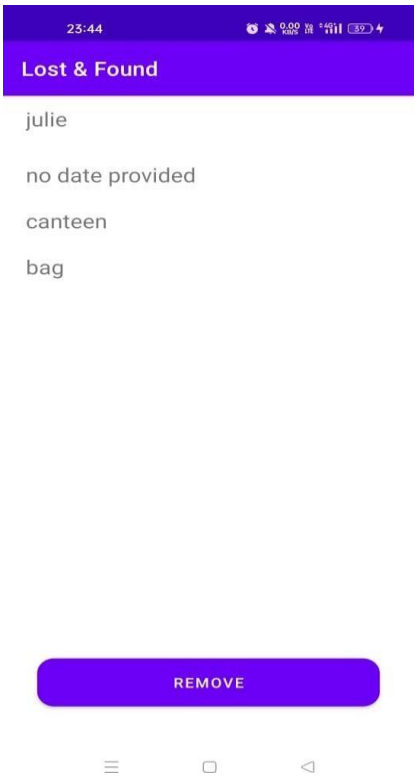


for lost items

7.4 Activity for entering details



7.5 Activity for displaying Lost items



7.6 Activity for displaying the items

Chapter 9: CONCLUSION

In conclusion, the development and implementation of an Android application for lost and found items in our college campus offer a significant solution to address the challenges posed by the students and the staff. By providing a centralized platform, this application will alter for good the way lost items are reported, tracked, and ultimately reunited with their owners.

The identified problem of inconvenience and frustration caused by missing belongings will be alleviated as students and staff can easily report their lost items through the user-friendly interface. This application will empower the college community to take an active role in the retrieval process, enhancing transparency and increasing the chances of successful item recovery.

Efficiency will be a key focus of this application, with features such as item categorization and matching algorithms streamlining the retrieval process. Timely updates and communication will keep users informed about the status of their lost items, reducing anxiety and uncertainty.

By fostering a more secure and supportive campus environment, this application will enhance the overall experience for students and staff.

In conclusion, the Android application for lost and found items in our college campus is a vital step towards establishing a centralized and efficient system. By harnessing the power of technology, this solution addresses the challenges associated with lost belongings, ultimately creating a safer and more supportive environment within our college community.

Chapter 10: FUTURE ENHANCEMENTS

Incorporating innovative features and functionalities will ensure that the application remains a valuable asset in reuniting lost items with their rightful owners. Here, we delve into several exciting possibilities for future enhancements.

Firstly, encouraging community collaboration through the application can be a compelling enhancement. By incorporating a rewards system, users who actively report found items or assist in the retrieval process could earn points or incentives. This gamification approach would incentivize and engage the college community, creating a sense of responsibility and camaraderie when it comes to reuniting lost items with their owners.

To further enhance the application's security and efficiency, integrating it with existing campus security systems would be a valuable step. This integration would enable security personnel to receive real-time alerts when lost items are reported or found, allowing for prompt action and coordination. Additionally, linking the application with surveillance cameras and access control systems could provide valuable insights and assistance in locating lost items.

Moreover, incorporating social media integration within the application can amplify its reach and impact. Users could easily share information about their lost items on popular social media platforms, increasing the chances of a successful recovery. Furthermore, incorporating social media elements within the application itself, such as a dedicated community forum or discussion board, would facilitate communication and collaboration among users.

REFERENCE

Book References-:

1. 1. Google Developer Training, "Android Developer Fundamentals Course – ConceptReference", Google Developer Training Team, 2017.

2. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, WileyIndia Pvt Ltd, 2014. ISBN-13: 978-8126547197.

3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition,O"Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341.

Web References-:

1. https://developer.android.com
2. https://www.geeksforgeeks.org
3. https://stackoverflow.com
4. https://www.w3schools.com