Table of Contents

[COMP1786 – Mobile Application Design and Development Report 3](#_Toc151535673)

[Section 1: CONCISE TABLE 3](#_Toc151535674)

[Section 2: SCREEN SHOTS/DESIGN (JAVA) 5](#_Toc151535675)

[Section 3: REFLECTION 22](#_Toc151535676)

[I. How the app was developed 22](#_Toc151535677)

[II. Lessons learnt: 23](#_Toc151535678)

[III. What I think went well in application development 23](#_Toc151535679)

[IV. Improvements to the app 23](#_Toc151535680)

[Section 4: EVALUATION 23](#_Toc151535681)

[I. Human computer interaction 23](#_Toc151535682)

[II. Security 24](#_Toc151535683)

[III. The application's ability to adjust to different screen sizes and future improvements 24](#_Toc151535684)

[IV. Changes need to be made 26](#_Toc151535685)

[Section 5: CODE 26](#_Toc151535686)

[File AddHike.java 26](#_Toc151535687)

[AddObservation.java 31](#_Toc151535688)

[EditHike.java 33](#_Toc151535689)

[EditObservation.java 39](#_Toc151535690)

[HikeAdapter.java 41](#_Toc151535691)

[ObservationAdapter.java 45](#_Toc151535692)

[DBHelper.java 48](#_Toc151535693)

[Hike.java 59](#_Toc151535694)

[Observation.java 64](#_Toc151535695)

[HikeDetail.java 68](#_Toc151535696)

[MainHike.java 71](#_Toc151535697)

[MainObservation.java 76](#_Toc151535698)

[ObservationDetail.java 79](#_Toc151535699)

[add\_hike.xml 81](#_Toc151535700)

[add\_observation.xml 88](#_Toc151535701)

[edit\_hike.xml 92](#_Toc151535702)

[edit\_observation.xml 97](#_Toc151535703)

[hike\_detail.xml 100](#_Toc151535704)

[item\_hike.xml 109](#_Toc151535705)

[item\_observation.xml 113](#_Toc151535706)

[main\_hike.xml 118](#_Toc151535707)

[main\_observation.xml 119](#_Toc151535708)

[observation\_detail.xml 121](#_Toc151535709)

[main\_top\_menu.xml 126](#_Toc151535710)

[arrayLevel.xml 127](#_Toc151535711)

[arrayVehicle.xml 127](#_Toc151535712)

[Figure 1: home and add page 5](#_Toc151535713)

[Figure 2: add hike 6](#_Toc151535714)

[Figure 3: fill in missing information 7](#_Toc151535715)

[Figure 4: add length the hike 8](#_Toc151535716)

[Figure 5: click button add 9](#_Toc151535717)

[Figure 6: add more 10](#_Toc151535718)

[Figure 7: hike detail 11](#_Toc151535719)

[Figure 8: click button back 12](#_Toc151535720)

[Figure 9: delete hike 13](#_Toc151535721)

[Figure 10: edit hike 14](#_Toc151535722)

[Figure 11: add observation 15](#_Toc151535723)

[Figure 12: observation detail 16](#_Toc151535724)

[Figure 13: button back in observation 17](#_Toc151535725)

[Figure 14: delete observation 18](#_Toc151535726)

[Figure 15: edit observation 19](#_Toc151535727)

[Figure 16: back to home page 20](#_Toc151535728)

[Figure 17: search hike 21](#_Toc151535729)

[Figure 18: delete all 22](#_Toc151535730)

[Table 1: CONCISE TABLE 5](#_Toc151535731)

# COMP1786 – Mobile Application Design and Development Report

| **Your name** | **Nguyen Duc Minh** | **Your Student ID** | **001357686** |
| --- | --- | --- | --- |

# Section 1: CONCISE TABLE

|  |  |  |
| --- | --- | --- |
| **Feature** | **Status** | **Your Comments** |
| **Enter details of hikes** | Fully completed **✔** | I successfully completed the data import task. These fields included: Name of hike, Location, Date of the hike, Parking available, Length of the hike, Level of difficulty, Description. Furthermore, in addition to the fields you initially mentioned, you've introduced a new field: Vehicle |
| **Store the database** | Fully completed✔ | The user-provided information should be initially saved on the device within an SQLite database. |
| **view the database** | Fully completed✔ | Users have the capability to view a complete list of all the hike details that have been input into the application. |
| **delete hike** | Fully completed✔ | Users have the option to either delete individual hikes or clear all the details from the database. |
| **Create hike** | Fully completed✔ | Users can create a new hike and save it to the SQLite database. |
| **Update hike** | Fully completed✔ | Users can Update a hike and save it to the SQLite database. |
| **Add observations to a hike** | Fully completed✔ | Hikers have the option to choose a hiking excursion and subsequently input the following information: Observation, Timestamp of the observation, Additional comments.  Users should have the ability to record multiple observations for a single hiking trip. The app save all the data locally in an SQLite database. Furthermore, users should be able to choose a specific hike, view a complete list of observations, and perform actions like creating viewing, editing, or deleting particular observations. |
| **Search** | Fully completed✔ | Users have the functionality to search for a specific hike in the database using the name of the hike as a search parameter. This feature enables efficient and convenient information retrieval, making it easy for users to find and access detailed information about their desired hiking trip quickly and efficiently. |
| **Create a cross-platform prototype of the app using Xamarin/MAUI** | Not implemented ✔ | I regret to inform you that I am unable to complete this quest. |
| **Implement persistence using Xamarin/MAUI** | Not implemented ✔ | I regret to inform you that I am unable to complete this quest. |
| **Integrate supplementary functionalities into either the Android or Xamarin iteration of the application.** | Not implemented ✔ | I regret to inform you that I am unable to complete this quest. |
| **Link to recorded video (if you record your application before submitting the report)** | | |
| [**https://drive.google.com/drive/folders/1-k2SKaUCJPjvextiZA\_rqZR\_yzD-HmaP?usp=sharing**](https://drive.google.com/drive/folders/1-k2SKaUCJPjvextiZA_rqZR_yzD-HmaP?usp=sharing) | | |

Table 1: CONCISE TABLE

# Section 2: SCREEN SHOTS/DESIGN (JAVA)

Screens screenshot of a screenshot of a screenshot

Description automatically generated

Figure 1: home and add page

Above is the function to add a new hike. The user clicks on the plus sign in the lower right corner and the screen switches to a screen containing an information input box and two additional buttons, the ADD button and the CANCEL button. If the user wants to save after entering, press the ADD button (Figure 4) to save and when pressing CANCEL, the information will not be saved and returned to the main screen.

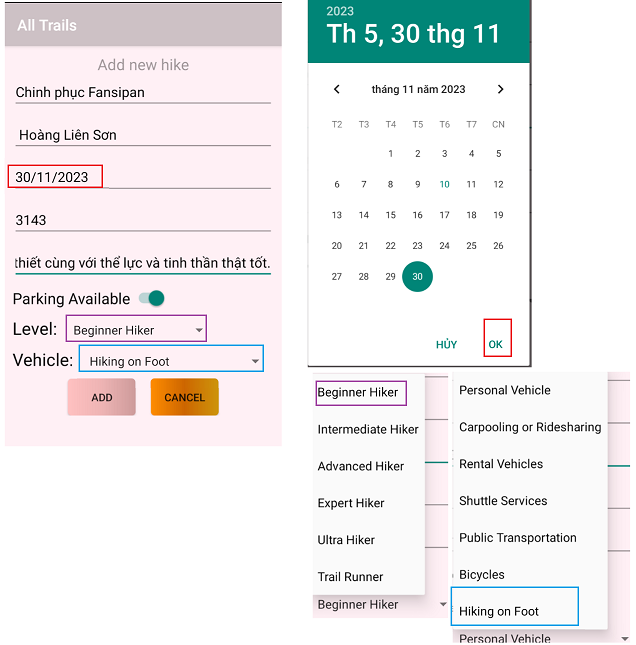


Figure 2: add hike

More information and fields. When the user clicks on the date input box, the application will display a calendar to select the date. The 3 fields Parking Available, Level and Vehicle do not have to be entered but will be selected.

A screenshot of a phone

Description automatically generated

Figure 3: fill in missing information

When the user fills in missing information, the application displays the message: "Please complete all information" to notify the user that they have not filled in all the information. In addition, if the last 3 fields are: Parking Available, Level and Vehicle, if not selected, the default values ​​will be No, Beginner Hiker and Personal Vehicle.

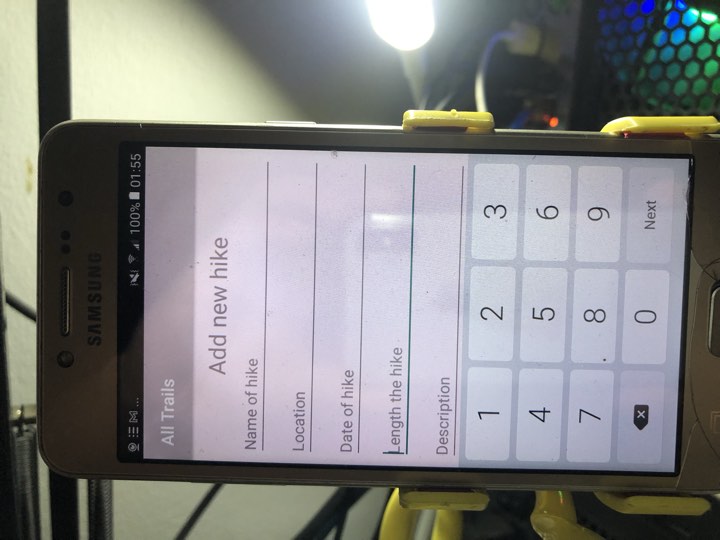


Figure 4: add length the hike

When entering information in the Length the hike line, the keyboard does not display letters but displays a numeric keypad (only for entering numbers).

Screens screenshot of a phone

Description automatically generated

Figure 5: click button add

Screens screenshot of a phone

Description automatically generated

Figure 6: add more

The main screen will display the hiker's hike list and information including name and address to the right of the red number is date, month, year. Below there are 2 buttons: 1 button to edit information and 1 button for users to add observations for each trip.

A screenshot of a phone

Description automatically generated

Figure 7: hike detail

When clicking on any trip, detailed information of that trip will appear and there are 2 buttons below to delete if necessary (Figure 8). In addition, you can return if you have finished viewing (Figure 7).

Screens screenshot of a phone

Description automatically generated

Figure 8: click button back

Screens screenshot of a phone

Description automatically generated

Figure 9: delete hike

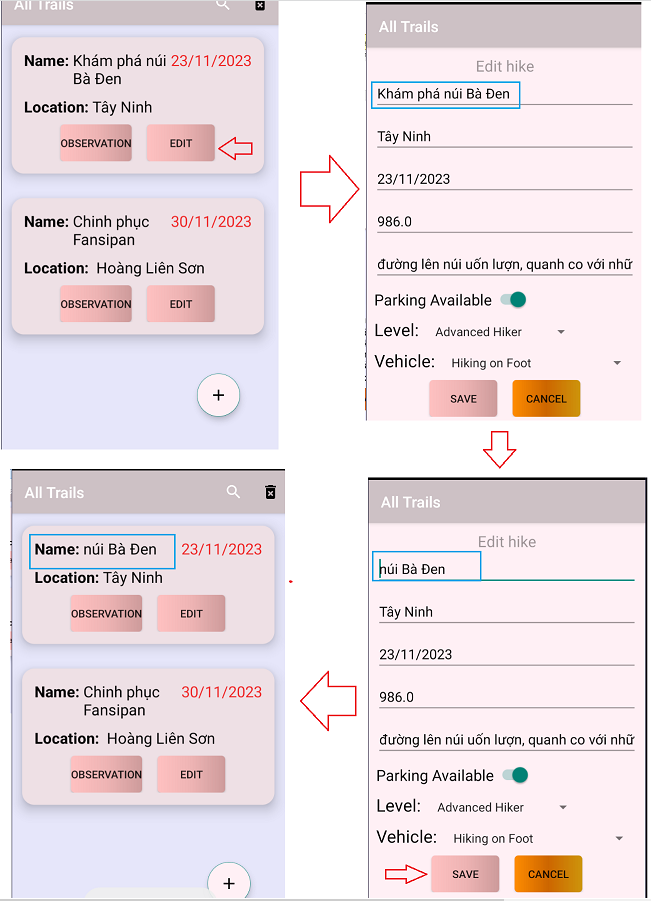


Figure 10: edit hike

When the user wants to edit trip information, click the Edit button on that trip. The screen will appear as when adding a new item, but information will be displayed corresponding to each cell so that users can easily change the information that needs to be edited. When finished editing, the user clicks SAVE to save the newly changed information.

Screens screenshot of a computer

Description automatically generated

Figure 11: add observation

When you want to add observations to each trip, the user presses the OBSERVATION button of that trip and adds a new observation. Here we will add Observation and Comment. time is the actual date and time.

A screenshot of a phone

Description automatically generated

Figure 12: observation detail

When you click on each Observation, you can see the details. Click back (BACK button) to return to the Observations list (Figure 12). If the user wants to delete, click DELETE (Figure 13).

A screenshot of a phone

Description automatically generated

Figure 13: button back in observation

A screenshot of a phone

Description automatically generated

Figure 14: delete observation

Screenshot of a screenshot of a screenshot of a screenshot of a screenshot of a screenshot of a screenshot of a screenshot of a screenshot of a screenshot of

Description automatically generated

Figure 15: edit observation

If the user wants to edit the Observation's information, click the EDIT button of that Observation to edit and click SAVE if the changes have been made.

Screens screenshot of a phone

Description automatically generated

Figure 16: back to home page

The bottom left corner of the Observations list with an arrow to the left is the button to return to the Hikes list

Screens screenshot of a phone

Description automatically generated

Figure 17: search hike

When users want to search for information about a certain trip, they can click on the magnifying glass in the upper right corner and enter the name of that trip.

Screens screenshot of a chat

Description automatically generated

Figure 18: delete all

Next to the magnifying glass is a trash icon to delete the existing Hikes list.

# Section 3: REFLECTION

## How the app was developed

The hike management application development process involves many stages, from ideation and planning to implementation and refinement. I defined the purpose and target audience of the app, identified the core features and functions. The design and user experience are meticulously crafted to ensure seamless and visually appealing interactions. The application is built in Java language, SQLite database is integrated for effective data management. Strict testing and quality assurance procedures have been implemented to identify and correct errors. This application also includes data management features, such as storing, adding, editing, and deleting information in the SQLite database. The application's search function has also been enhanced.

## Lessons learnt:

Completion of the hike management application development project was an important milestone that provided valuable lessons for future endeavours. Key lessons include the importance of user feedback, prioritizing security, ensuring comprehensive documentation, adopting agile methods, continuous testing, communicating effectively, investing in User testing, budgeting for application development. Reflecting on these lessons and incorporating them into future projects will contribute to my growth and success.

## What I think went well in application development

I think the success of an application development project is often due to clear requirements, effective project management, application of best development practices, responsive design, effective communication, Thorough testing, security considerations, agile development, and user training and support. These factors contribute to creating a solid foundation for me to develop the project.

## Improvements to the app

I will need to improve some features for my application. Firstly, the app can be integrated with sensors and measuring devices to update location, speed, and distance data automatically. Secondly, email or social networking sites are simple ways for users to tell friends and family about their travel plans. In the community, this strengthens bonds and establishes hubs for communication. Finally, data security becomes a significant concern because the app contains a large amount of sensitive personal information. Leakage of personal information may occur from an intrusion or lost password.

# Section 4: EVALUATION

## Human computer interaction

I review the hiking management app:

First, User Experience (UX): Colors are given to the interface to highlight information and make it easy to see. Contrast between text and background can improve user readability and comprehension. Use color to highlight important elements such as function buttons, links, or important notifications. This color will stand out and be easy to see. provides alternative input methods, corrects information, and deletes unnecessary data. Search features and a clear navigation system help users easily access the information and features they need. The data storage feature also helps users compare different trips so they can find patterns, developments, and adjustments to improve their efficiency.

Second, social aspect: the impact of Hike app management on social relationships reflects broader trends in the digital era. By leveraging the positive aspects of Hike while being mindful of its potential limitations, individuals can navigate the digital landscape in a way that enhances rather than detracts from the fabric of connection between people together.

Finally, The Hike management application, when informed by a deep understanding of human physical and psychological characteristics, can go beyond mere functionality to become an integral part of user's outdoor experience. By seamlessly integrating with the user's abilities and limitations, the app not only becomes more user-friendly but also enhances the overall satisfaction and effectiveness of the hiking experience.

## Security

Security in the development of my Hike management application was an important aspect. for example, input validation: Validate and sanitize all input to prevent security vulnerabilities. Information that can only be numeric will only be entered (numeric keypad is displayed). Fields such as Parking Available, Vehicle and Level will be selected to avoid users making mistakes or entering incorrect information. Security is an ongoing process and regular assessment, and updates are essential. So, I'm still improving the security of my application

## The application's ability to adjust to different screen sizes and future improvements

The ability for an app to run effectively across multiple screen sizes is critical to delivering a consistent and user-friendly experience across a variety of devices. Here are the things I evaluate and improve my app's responsiveness to different screen sizes:

**First, check on devices:**

*Current state:* I have tested on devices with various screen sizes and the results are that there are no input boxes or letters overlapping each other. The only problem is that with a large screen, the text on the application is small and vice versa.

Improvement: Expanded testing efforts to cover more device types, including smartphones, tablets, and various desktop displays.

**Second, consistent user interface (UI):**

*Current state*: Ensure that the user interface remains consistent across different screen sizes.

*Enhancement:* Icons, buttons and navigation adapt to different screen sizes, without affecting usability.

**Third, content priority:**

*Current state*: content on different screen sizes is a priority. I have displayed important information for each trip first and detailed information is displayed when users click on each trip.

Improvement: Prioritize essential content for smaller screens to ensure users get the most important information without scrolling.

**Fourth, font size and readability:**

*Current state:* text legibility on different screen sizes. In this matter I have done well, the font color is easy to see and read. Light and bright tones highlight the black tones.

*Enhancement*: Optimized font size and spacing for readability on different devices.

**Final, user feedback:**

*Current state*: I have been collecting user feedback on their experience across different devices.

*Improvement*: Use user feedback to identify specific issues related to screen size. From there I can improve and perfect my application.

By focusing on these aspects, I can enhance the app's ability to run seamlessly across various screen sizes, providing a better user experience for my audience.

## Changes need to be made

*Automation*: The application can integrate with measuring devices and sensors, automatically updating information about distance, speed, and location. allowing for the automatic updating of information related to distance, speed, and location. This integration not only enhances the accuracy of data but also significantly improves the overall user experience by reducing manual input and providing real-time, dynamic information.

*Share and connect*: If the "Share and Connect" feature in my app is developed it will represent a strong social dimension that goes beyond the individual user experience, allowing users to easily share trip information with their friends and family. This function not only supports communication but also establishes sharing points within the community. The "Share and Connect" feature, when thoughtfully designed and continuously improved, not only enhances the app's functionality but also fosters a sense of community among users. By providing social interactions and shared experiences, the app becomes a platform that goes beyond utility, contributing to building meaningful connections within the user community.

*Data security:* Data security is a top concern not only for each of my apps, but for any app that handles large amounts of personal information and sensitive data. In the case of my application that involves managing trip information, the importance of strong data security measures is undeniable. The potential risks associated with data breaches, lost passwords or unauthorized entry highlight the need for a comprehensive and prudent approach to securing user information. In short, securing user data in applications is a multifaceted and ongoing commitment. Regularly evaluating and enhancing security measures, along with proactive user education and compliance with regulatory standards, will contribute to building and maintaining user trust.

# Section 5: CODE

## File AddHike.java

public class AddHike extends AppCompatActivity {

public Button btnAdd, btncancel;

public EditText h\_name, h\_location, h\_date, h\_length, h\_description;

public Switch h\_parking;

public Spinner h\_level, h\_vehicle;

private DBHelper db;

private ArrayList<Hike> hikeList = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.add\_hike);

db = new DBHelper(this);

mapping();

btncancel.setOnClickListener(view -> {

finish();

});

h\_date.setOnClickListener(view -> {

MyDatePicker dlg = new MyDatePicker();

dlg.setDateField(h\_date);

dlg.show(getSupportFragmentManager(), "Hike Date!");

});

btnAdd.setOnClickListener(view -> {

String nameValue = h\_name.getText().toString();

String locationValue = h\_location.getText().toString();

String dateValue = h\_date.getText().toString();

String selectedLevel = h\_level.getSelectedItem().toString();

String selectedVehicle = h\_vehicle.getSelectedItem().toString();

String descriptionValue = h\_description.getText().toString();

String lengthValue = h\_length.getText().toString();

if (nameValue.isEmpty() || locationValue.isEmpty() ||

dateValue.isEmpty() || descriptionValue.isEmpty() ||

lengthValue.isEmpty()) {

Toast.makeText(AddHike.this, "Please complete all information", Toast.LENGTH\_SHORT).show();

} else {

int parkingValue = h\_parking.isChecked() ? 1 : 0;

try {

double length = Double.parseDouble(lengthValue);

long id = db.addHike(nameValue, locationValue, dateValue, selectedLevel, descriptionValue,selectedVehicle, length, parkingValue);

Toast.makeText(AddHike.this, "Create Successfully " + id, Toast.LENGTH\_SHORT).show();

Intent i = new Intent();

setResult(RESULT\_OK, i);

finish();

} catch (NumberFormatException e) {

Toast.makeText(AddHike.this, "Invalid length format", Toast.LENGTH\_SHORT).show();

}

}

});

}

public void mapping(){

btnAdd = findViewById(R.id.btn\_add);

btncancel = findViewById(R.id.btn\_cancel);

h\_name = findViewById(R.id.name);

h\_location = findViewById(R.id.location);

h\_date = findViewById(R.id.date);

h\_length = findViewById(R.id.length);

h\_description = findViewById(R.id.description);

h\_parking = findViewById(R.id.parking);

h\_level = findViewById(R.id.level);

h\_vehicle = findViewById(R.id.vehicle);

}

public static class MyDatePicker extends DialogFragment implements DatePickerDialog.OnDateSetListener {

public void setDateField(EditText dateField) {

this.dateField = dateField;

}

@Override

public Dialog onCreateDialog(Bundle savedInstanceState) {

if (dateField.getText().length() != 0) {

String date = dateField.getText().toString();

String[] separated = date.split("/");

int year = Integer.parseInt(separated[2]);

int month = Integer.parseInt(separated[1]);

int day = Integer.parseInt(separated[0]);

return new DatePickerDialog(getActivity(), this, year, month - 1, day);

} else {

final Calendar c = Calendar.getInstance();

int year = c.get(Calendar.YEAR);

int month = c.get(Calendar.MONTH);

int day = c.get(Calendar.DAY\_OF\_MONTH);

return new DatePickerDialog(getActivity(), this, year, month, day);

}

}

private EditText dateField;

@Override

public void onDateSet(DatePicker datePicker, int selectedYear,

int selectedMonth, int selectedDay) {

String dateReturn = selectedDay + "/" + (selectedMonth + 1) + "/"

+ selectedYear;

dateField.setText(dateReturn);

}

}

}

## AddObservation.java

public class AddObservation extends AppCompatActivity {

public EditText o\_observation, o\_dateOfTime, o\_comment;

public Button btnAdd, btnCancel;

public DBHelper db;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.add\_observation);

mapping();

Date currentTime = new Date();

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

String formattedTime = sdf.format(currentTime);

o\_dateOfTime.setText("" + formattedTime);

db = new DBHelper(this);

Intent intent = getIntent();

int hikeID = intent.getIntExtra("hikeID", -1);

btnCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

btnAdd.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String observationText = o\_observation.getText().toString().trim();

String dateOfTimeText = o\_dateOfTime.getText().toString().trim();

String commentText = o\_comment.getText().toString().trim();

if (observationText.isEmpty() || dateOfTimeText.isEmpty() || commentText.isEmpty()) {

Toast.makeText(AddObservation.this, "Please complete all information", Toast.LENGTH\_SHORT).show();

} else {

long id = db.addObservation(

observationText,

dateOfTimeText,

commentText,

hikeID

);

Toast.makeText(AddObservation.this, "Create Successfully " + id, Toast.LENGTH\_SHORT).show();

Intent resultIntent = new Intent();

setResult(RESULT\_OK, resultIntent);

finish();

}

}

});

}

public void mapping(){

o\_observation = findViewById(R.id.observation);

o\_dateOfTime = findViewById(R.id.date);

o\_comment = findViewById(R.id.comment);

btnAdd = findViewById(R.id.btn\_add);

btnCancel = findViewById(R.id.btn\_cancel);

}

}

## EditHike.java

public class EditHike extends AppCompatActivity {

public Button btnSave, btnCancel;

public EditText h\_name, h\_location, h\_date, h\_length, h\_description;

public Switch h\_parking;

public Spinner h\_level, h\_vehicle;

private DBHelper db;

private ArrayList<Hike> hikeList = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.edit\_hike);

db = new DBHelper(this);

mapping();

Intent i = getIntent();

String name = i.getStringExtra("name");

String location = i.getStringExtra("location");

String date = i.getStringExtra("date");

double length = i.getDoubleExtra("length", 0.0);

String level = i.getStringExtra("level");

String vehicle = i.getStringExtra("vehicle");

String description = i.getStringExtra("description");

boolean parkingValue = getIntent().getBooleanExtra("parking", false);

ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.level, android.R.layout.simple\_spinner\_item);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

h\_level.setAdapter(adapter);

ArrayAdapter<CharSequence> adapter1 = ArrayAdapter.createFromResource(this, R.array.vehicle, android.R.layout.simple\_spinner\_item);

adapter1.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

h\_vehicle.setAdapter(adapter1);

h\_name.setText(name);

h\_location.setText(location);

h\_date.setText(date);

h\_level.setAdapter(adapter);

h\_vehicle.setAdapter(adapter1);

h\_length.setText(String.valueOf(length));

h\_description.setText(description);

h\_parking.setChecked(parkingValue);

if (level != null) {

int spinnerPosition = adapter.getPosition(level);

h\_level.setSelection(spinnerPosition);

}

if (vehicle != null) {

int spinnerPosition = adapter1.getPosition(vehicle);

h\_vehicle.setSelection(spinnerPosition);

}

h\_date.setOnClickListener(view -> {

MyDatePicker dlg = new MyDatePicker();

dlg.setDateField(h\_date);

dlg.show(getSupportFragmentManager(), "Hike Date!");

});

btnSave.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String name = h\_name.getText().toString();

String location = h\_location.getText().toString();

String date = h\_date.getText().toString();

String lengthStr = h\_length.getText().toString();

String level = h\_level.getSelectedItem().toString();

String vehicle = h\_vehicle.getSelectedItem().toString();

String description = h\_description.getText().toString();

boolean parkingChecked = h\_parking.isChecked();

if (name.isEmpty() || location.isEmpty() || date.isEmpty() || lengthStr.isEmpty() || description.isEmpty()) {

Toast.makeText(EditHike.this, "Please complete all information", Toast.LENGTH\_SHORT).show();

} else {

try {

double length = Double.parseDouble(lengthStr);

int parkingValue = parkingChecked ? 1 : 0;

int id = getIntent().getIntExtra("id", 0);

db.editHike(id, name, location, date, level, description,vehicle, length, parkingValue);

Toast.makeText(EditHike.this, "Update Successfully", Toast.LENGTH\_SHORT).show();

Intent i = new Intent();

setResult(RESULT\_OK, i);

finish();

} catch (NumberFormatException e) {

Toast.makeText(EditHike.this, "Invalid length format. Please enter a valid number.", Toast.LENGTH\_SHORT).show();

}

}

}

});

btnCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

}

public void mapping(){

btnSave = findViewById(R.id.btn\_save);

btnCancel = findViewById(R.id.btn\_cancel);

h\_name = findViewById(R.id.name);

h\_location = findViewById(R.id.location);

h\_date = findViewById(R.id.date);

h\_length = findViewById(R.id.length);

h\_description = findViewById(R.id.description);

h\_parking = findViewById(R.id.parking);

h\_level = findViewById(R.id.level);

h\_vehicle = findViewById(R.id.vehicle);

}

public static class MyDatePicker extends DialogFragment implements DatePickerDialog.OnDateSetListener {

public void setDateField(EditText dateField) {

this.dateField = dateField;

}

@Override

public Dialog onCreateDialog(Bundle savedInstanceState) {

if (dateField.getText().length() != 0) {

String date = dateField.getText().toString();

String[] separated = date.split("/");

int year = Integer.parseInt(separated[2]);

int month = Integer.parseInt(separated[1]);

int day = Integer.parseInt(separated[0]);

return new DatePickerDialog(getActivity(), this, year, month - 1, day);

} else {

final Calendar c = Calendar.getInstance();

int year = c.get(Calendar.YEAR);

int month = c.get(Calendar.MONTH);

int day = c.get(Calendar.DAY\_OF\_MONTH);

return new DatePickerDialog(getActivity(), this, year, month, day);

}

}

private EditText dateField;

@Override

public void onDateSet(DatePicker datePicker, int selectedYear,

int selectedMonth, int selectedDay) {

String dateReturn = selectedDay + "/" + (selectedMonth + 1) + "/"

+ selectedYear;

dateField.setText(dateReturn);

}

}

}

## EditObservation.java

public class EditObservation extends AppCompatActivity {

public EditText O\_observation, O\_date, O\_comment;

private Button btnSave, btnCancel;

private DBHelper db;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.edit\_observation);

mapping();

db = new DBHelper(this);

//

Intent i = getIntent();

String name = i.getStringExtra("observation");

String date = i.getStringExtra("date");

String comment = i.getStringExtra("comment");

O\_observation.setText(name);

O\_date.setText(date);

O\_comment.setText(comment);

//

btnSave.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String observation = O\_observation.getText().toString().trim();

String date = O\_date.getText().toString().trim();

String comment = O\_comment.getText().toString().trim();

if (observation.isEmpty() || date.isEmpty() || comment.isEmpty()) {

Toast.makeText(EditObservation.this, "Please complete all information", Toast.LENGTH\_SHORT).show();

} else {

int id = getIntent().getIntExtra("id", 0);

db.eitObservation(id, observation, date, comment);

Toast.makeText(EditObservation.this, "Update Successfully", Toast.LENGTH\_SHORT).show();

Intent resultIntent = new Intent();

setResult(RESULT\_OK, resultIntent);

finish();

}

}

});

//

btnCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

}

public void mapping(){

O\_observation = findViewById(R.id.observation);

O\_date = findViewById(R.id.date);

O\_comment = findViewById(R.id.comment);

btnSave = findViewById(R.id.btn\_save);

btnCancel = findViewById(R.id.btn\_cancel);

}

}

## HikeAdapter.java

public class HikeAdapter extends RecyclerView.Adapter<HikeAdapter.HikeViewHolder> {

public Context context;

public ArrayList<Hike> hikeList;

public MainHike mainActivity;

public HikeAdapter(Context context, ArrayList<Hike> hikeList, MainHike mainActivity) {

this.context = context;

this.hikeList = hikeList;

this.mainActivity = mainActivity;

}

@NonNull

@Override

public HikeViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item\_hike, parent, false);

HikeViewHolder hikeViewHolder = new HikeViewHolder(view);

return hikeViewHolder;

}

@Override

public void onBindViewHolder(@NonNull HikeViewHolder holder, int position) {

Hike hike = hikeList.get(position);

int idH = hike.getId();

String nameH = hike.getName();

String descriptionH = hike.getDescription();

String dateH = hike.getDate();

String locationH = hike.getLocation();

String levelH = hike.getLevel();

String vehicleH = hike.getVehicle();

double lengthH = hike.getLength();

int parkingH = hike.isParking();

holder.h\_name.setText(nameH);

holder.h\_location.setText(locationH);

holder.h\_date.setText(dateH);

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(mainActivity, HikeDetail.class);

i.putExtra("hikeID", idH);

mainActivity.startActivityForResult(i, 1);

}

});

holder.btnEdit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(mainActivity, EditHike.class);

i.putExtra("id", idH);

i.putExtra("name", nameH);

i.putExtra("location", locationH);

i.putExtra("date", dateH);

i.putExtra("level", levelH);

i.putExtra("vehicle", vehicleH);

i.putExtra("description", descriptionH);

i.putExtra("length", lengthH);

i.putExtra("parking", parkingH == 1);

mainActivity.startActivityForResult(i, 2);

}

});

holder.btnObservation.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(mainActivity, MainObservation.class);

i.putExtra("hikeID", idH);

context.startActivity(i);

}

});

}

@Override

public int getItemCount() {

if(hikeList != null){

return hikeList.size();

}

return 0;

}

class HikeViewHolder extends RecyclerView.ViewHolder{

public TextView h\_name, h\_location, h\_date;

public Switch parking;

public Button btnEdit, btnObservation;

public HikeViewHolder(@NonNull View itemView) {

super(itemView);

this.h\_name = itemView.findViewById(R.id.name);

this.h\_date = itemView.findViewById(R.id.date);

this.h\_location = itemView.findViewById(R.id.location);

this.parking = itemView.findViewById(R.id.parking);

this.btnEdit = itemView.findViewById(R.id.btn\_edit);

this.btnObservation = itemView.findViewById(R.id.btn\_observation);

}

}

}

## ObservationAdapter.java

public class ObservationAdapter extends RecyclerView.Adapter<ObservationAdapter.ObservationViewHolder> {

public Context context;

public ArrayList<Observation> obList;

public MainObservation viewObservationActivity;

class ObservationViewHolder extends RecyclerView.ViewHolder{

public TextView o\_observation, o\_date, o\_comment;

public Button btnUpdate;

public ObservationViewHolder(@NonNull View itemView) {

super(itemView);

this.o\_observation = itemView.findViewById(R.id.observation);

this.o\_date = itemView.findViewById(R.id.date);

this.o\_comment = itemView.findViewById(R.id.comment);

this.btnUpdate = itemView.findViewById(R.id.btn\_edit);

}

}

public ObservationAdapter(Context context, ArrayList<Observation> obList, MainObservation viewObservationActivity) {

this.context = context;

this.obList = obList;

this.viewObservationActivity = viewObservationActivity;

}

@NonNull

@Override

public ObservationViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item\_observation, parent, false);

ObservationViewHolder observationViewHolder = new ObservationViewHolder(view);

return observationViewHolder;

}

@Override

public void onBindViewHolder(@NonNull ObservationViewHolder holder, int position) {

Observation observation = obList.get(position);

int id = observation.getId();

String ob = observation.getObservation();

String date = observation.getDateOfTime();

String comment = observation.getComment();

holder.o\_observation.setText(ob);

holder.o\_date.setText(date);

holder.o\_comment.setText(comment);

holder.btnUpdate.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(viewObservationActivity, EditObservation.class);

i.putExtra("id", id);

i.putExtra("observation", ob);

i.putExtra("date", date);

i.putExtra("comment", comment);

viewObservationActivity.startActivityForResult(i, 1);

}

});

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(viewObservationActivity, ObservationDetail.class);

i.putExtra("observationID", id);

viewObservationActivity.startActivityForResult(i, 1);

}

});

}

@Override

public int getItemCount() {

if(obList != null){

return obList.size();

}

return 0;

}

}

## DBHelper.java

public class DBHelper extends SQLiteOpenHelper {

private static final int DATABASE\_VERSION = 2;

private static final String DATABASE\_TABLE = "hike\_db";

public DBHelper(@Nullable Context context) {

super(context, DATABASE\_TABLE, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

sqLiteDatabase.execSQL(Hike.CREATE\_TABLE);

sqLiteDatabase.execSQL(Observation.CREATE\_TABLE);

}

@Override

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + Hike.TABLE\_NAME);

sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + Observation.TABLE\_NAME);

onCreate(sqLiteDatabase);

}

public long addHike(String name, String location, String date, String level, String description,String vehicle, double length, int parking){

SQLiteDatabase db = this.getWritableDatabase();

ContentValues cv = new ContentValues();

cv.put(Hike.COLUMN\_NAME, name);

cv.put(Hike.COLUMN\_LOCATION, location);

cv.put(Hike.COLUMN\_DATE, date);

cv.put(Hike.COLUMN\_LEVEL, level);

cv.put(Hike.COLUMN\_DESCRIPTION, description);

cv.put(Hike.COLUMN\_VEHICLE, vehicle);

cv.put(Hike.COLUMN\_LENGTH, length);

cv.put(Hike.COLUMN\_PARKING, parking);

long id = db.insert(Hike.TABLE\_NAME, null, cv);

db.close();

return id;

}

public void editHike(int id, String name, String location, String date, String level, String description,String vehicle, double length, int parking){

SQLiteDatabase db = this.getWritableDatabase();

ContentValues cv = new ContentValues();

cv.put(Hike.COLUMN\_NAME, name);

cv.put(Hike.COLUMN\_LOCATION, location);

cv.put(Hike.COLUMN\_DATE, date);

cv.put(Hike.COLUMN\_LEVEL, level);

cv.put(Hike.COLUMN\_DESCRIPTION, description);

cv.put(Hike.COLUMN\_VEHICLE, vehicle);

cv.put(Hike.COLUMN\_LENGTH, length);

cv.put(Hike.COLUMN\_PARKING, parking);

db.update(Hike.TABLE\_NAME, cv, Hike.COLUMN\_ID + "=?",

new String[]{String.valueOf((id))});

db.close();

}

public void deleteHike(int id){

SQLiteDatabase db = getWritableDatabase();

db.delete(Hike.TABLE\_NAME, Hike.COLUMN\_ID + "=?",

new String[]{String.valueOf(id)});

db.close();

}

public void deleteAllHike(){

SQLiteDatabase db = getWritableDatabase();

db.execSQL("DELETE FROM " + Hike.TABLE\_NAME);

db.close();

}

public ArrayList<Hike> getAllHike(){

ArrayList<Hike> hikes = new ArrayList<>();

String selectQuery = " SELECT \* FROM " + Hike.TABLE\_NAME + " ORDER BY " +

Hike.COLUMN\_ID + " DESC ";

SQLiteDatabase db = this.getReadableDatabase();

Cursor cursor = db.rawQuery(selectQuery, null);

if(cursor.moveToFirst()){

do {

Hike hike = new Hike();

hike.setId(cursor.getInt(cursor.getColumnIndexOrThrow(Hike.COLUMN\_ID)));

hike.setName(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_NAME)));

hike.setLocation(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LOCATION)));

hike.setDate(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_DATE)));

hike.setDescription(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_DESCRIPTION)));

hike.setLevel(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LEVEL)));

hike.setVehicle(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_VEHICLE)));

hike.setLength(Double.parseDouble(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LENGTH))));

int parkingValue = cursor.getInt(cursor.getColumnIndexOrThrow(Hike.COLUMN\_PARKING));

hike.setParking(parkingValue);

hikes.add(hike);

}while (cursor.moveToNext());

}

db.close();

return hikes;

}

public ArrayList<Hike> getSearchHike(String query){

ArrayList<Hike> hikeList = new ArrayList<>();

SQLiteDatabase db = this.getReadableDatabase();

String queryToSearch = "SELECT \* FROM " + Hike.TABLE\_NAME + " WHERE " +

Hike.COLUMN\_NAME + " LIKE '%" + query + "%'";

Cursor cursor = db.rawQuery(queryToSearch, null);

if(cursor.moveToFirst()){

do {

Hike hike = new Hike();

hike.setId(cursor.getInt(cursor.getColumnIndexOrThrow(Hike.COLUMN\_ID)));

hike.setName(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_NAME)));

hike.setLocation(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LOCATION)));

hike.setDate(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_DATE)));

hike.setDescription(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_DESCRIPTION)));

hike.setLevel(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LEVEL)));

hike.setVehicle(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_VEHICLE)));

hike.setLength(Double.parseDouble(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LENGTH))));

int parkingValue = cursor.getInt(cursor.getColumnIndexOrThrow(Hike.COLUMN\_PARKING));

hike.setParking(parkingValue);

hikeList.add(hike);

}while (cursor.moveToNext());

}

db.close();

return hikeList;

}

public Hike getHikeById(int id) {

SQLiteDatabase database = this.getReadableDatabase();

String query = "SELECT \* FROM " + Hike.TABLE\_NAME + " WHERE " + Hike.COLUMN\_ID + " = " + id;

Cursor cursor = database.rawQuery(query, null);

Hike hike = null;

if (cursor.moveToFirst()) {

hike = new Hike();

hike.setId(cursor.getInt(cursor.getColumnIndexOrThrow(Hike.COLUMN\_ID)));

hike.setName(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_NAME)));

hike.setLocation(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LOCATION)));

hike.setDate(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_DATE)));

hike.setDescription(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_DESCRIPTION)));

hike.setLevel(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LEVEL)));

hike.setVehicle(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_VEHICLE)));

hike.setLength(Double.parseDouble(cursor.getString(cursor.getColumnIndexOrThrow(Hike.COLUMN\_LENGTH))));

int parkingValue = cursor.getInt(cursor.getColumnIndexOrThrow(Hike.COLUMN\_PARKING));

hike.setParking(parkingValue);

}

cursor.close();

database.close();

return hike;

}

//Data of Observation

public long addObservation(String observation, String dateOfTime, String comment, int hikeID) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues cv = new ContentValues();

cv.put(Observation.COLUMN\_NAME, observation);

cv.put(Observation.COLUMN\_DATE, dateOfTime);

cv.put(Observation.COLUMN\_COMMENT, comment);

cv.put(Observation.COLUMN\_FOREIGN\_KEY, hikeID);

long id = db.insert(Observation.TABLE\_NAME, null, cv);

db.close();

return id;

}

public void eitObservation(int id, String observation, String dateOfTime, String comment) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues cv = new ContentValues();

cv.put(Observation.COLUMN\_NAME, observation);

cv.put(Observation.COLUMN\_DATE, dateOfTime);

cv.put(Observation.COLUMN\_COMMENT, comment);

db.update(Observation.TABLE\_NAME, cv, Observation.COLUMN\_ID + " = ?",

new String[]{String.valueOf(id)});

db.close();

}

public void deleteObservation(int id) {

SQLiteDatabase db = getWritableDatabase();

db.delete(Observation.TABLE\_NAME, Observation.COLUMN\_ID + " = ?",

new String[]{String.valueOf(id)});

db.close();

}

public ArrayList<Observation> getObservationsForHike(int hikeID) {

ArrayList<Observation> observations = new ArrayList<>();

SQLiteDatabase db = this.getReadableDatabase();

String selectQuery = "SELECT \* FROM " + Observation.TABLE\_NAME + " WHERE " +

Observation.COLUMN\_FOREIGN\_KEY + " = " + hikeID;

Cursor cursor = db.rawQuery(selectQuery, null);

if (cursor.moveToFirst()) {

do {

Observation observation = new Observation();

observation.setId(cursor.getInt(cursor.getColumnIndexOrThrow(Observation.COLUMN\_ID)));

observation.setObservation(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLUMN\_NAME)));

observation.setDateOfTime(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLUMN\_DATE)));

observation.setComment(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLUMN\_COMMENT)));

observation.setHikeID(cursor.getInt(cursor.getColumnIndexOrThrow(Observation.COLUMN\_FOREIGN\_KEY)));

observations.add(observation);

} while (cursor.moveToNext());

}

db.close();

return observations;

}

public Observation getObservationById(int id) {

SQLiteDatabase database = this.getReadableDatabase();

String query = "SELECT \* FROM " + Observation.TABLE\_NAME + " WHERE " + Hike.COLUMN\_ID + " = " + id;

Cursor cursor = database.rawQuery(query, null);

Observation observation = null;

if (cursor.moveToFirst()) {

observation = new Observation();

observation.setId(cursor.getInt(cursor.getColumnIndexOrThrow(Observation.COLUMN\_ID)));

observation.setObservation(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLUMN\_NAME)));

observation.setDateOfTime(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLUMN\_DATE)));

observation.setComment(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLUMN\_COMMENT)));

observation.setHikeID(cursor.getInt(cursor.getColumnIndexOrThrow(Observation.COLUMN\_FOREIGN\_KEY)));

}

cursor.close();

database.close();

return observation;

}

}

## Hike.java

public class Hike {

public static final String TABLE\_NAME = "hike";

public static final String COLUMN\_ID = "hike\_id";

public static final String COLUMN\_NAME = "hike\_name";

public static final String COLUMN\_LOCATION = "hike\_location";

public static final String COLUMN\_DATE = "hike\_date";

public static final String COLUMN\_LEVEL = "hike\_level";

public static final String COLUMN\_DESCRIPTION = "hike\_description";

public static final String COLUMN\_LENGTH = "hike\_length";

public static final String COLUMN\_PARKING = "hike\_parking";

public static final String COLUMN\_VEHICLE = "hike\_vehicle";

private int id;

private String name;

private String location;

private String date;

private String level;

private String vehicle;

private String description;

private double length;

private int parking;

public Hike() {

}

public Hike(int id, String name, String location, String date, String level, String description,String vehicle, double length, int parking) {

this.id = id;

this.name = name;

this.location = location;

this.date = date;

this.level = level;

this.description = description;

this.vehicle = vehicle;

this.length = length;

this.parking = parking;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getLocation() {

return location;

}

public void setLocation(String location) {

this.location = location;

}

public String getDate() {

return date;

}

public void setDate(String date) {

this.date = date;

}

public String getLevel() {

return level;

}

public void setLevel(String level) {

this.level = level;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public double getLength() {

return length;

}

public void setLength(double length) {

this.length = length;

}

public int isParking() {

return parking;

}

public void setParking(int parking) {

this.parking = parking;

}

public String getVehicle() {

return vehicle;

}

public void setVehicle(String vehicle) {

this.vehicle = vehicle;

}

public static final String CREATE\_TABLE =

"CREATE TABLE " + TABLE\_NAME + "("

+ COLUMN\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"

+ COLUMN\_NAME + " TEXT,"

+ COLUMN\_LOCATION + " TEXT,"

+ COLUMN\_DATE + " TEXT,"

+ COLUMN\_LEVEL + " TEXT,"

+ COLUMN\_DESCRIPTION + " TEXT,"

+ COLUMN\_VEHICLE + " TEXT,"

+ COLUMN\_LENGTH + " REAL,"

+ COLUMN\_PARKING + " INTEGER"

+ ")";

@Override

public String toString() {

return "Hike{" +

"id=" + id +

", name='" + name + '\'' +

", location='" + location + '\'' +

", date='" + date + '\'' +

", level='" + level + '\'' +

", description='" + description + '\'' +

", vehicle='" + vehicle + '\'' +

", length=" + length +

", parking=" + parking +

'}';

}

}

## Observation.java

public class Observation {

public static final String TABLE\_NAME = "observations";

public static final String COLUMN\_ID = "observation\_id";

public static final String COLUMN\_NAME = "observation";

public static final String COLUMN\_DATE = "observation\_date\_time";

public static final String COLUMN\_COMMENT = "observation\_comment";

public static final String COLUMN\_FOREIGN\_KEY = "hike\_id";

private int id;

private String observation;

private String dateOfTime;

private String comment;

private int hikeID;

public Observation(){

}

public Observation(int id, String observation, String dateOfTime, String comment, int hikeID) {

this.id = id;

this.observation = observation;

this.dateOfTime = dateOfTime;

this.comment = comment;

this.hikeID = hikeID;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getObservation() {

return observation;

}

public void setObservation(String observation) {

this.observation = observation;

}

public String getDateOfTime() {

return dateOfTime;

}

public void setDateOfTime(String dateOfTime) {

this.dateOfTime = dateOfTime;

}

public String getComment() {

return comment;

}

public void setComment(String comment) {

this.comment = comment;

}

public int getHikeID() {

return hikeID;

}

public void setHikeID(int hikeID) {

this.hikeID = hikeID;

}

public static final String CREATE\_TABLE =

"CREATE TABLE " + TABLE\_NAME + "("

+ COLUMN\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"

+ COLUMN\_NAME + " TEXT,"

+ COLUMN\_DATE + " TEXT,"

+ COLUMN\_COMMENT + " TEXT,"

+ COLUMN\_FOREIGN\_KEY + " INTEGER,"

+ " FOREIGN KEY (" + COLUMN\_FOREIGN\_KEY + ") REFERENCES hike(hike\_id) "

+ ")";

@Override

public String toString() {

return "Hike{" +

"id=" + id +

", observation='" + observation + '\'' +

", dateOfTime='" + dateOfTime + '\'' +

", comment='" + comment + '\'' +

", hikeID='" + hikeID + '\'' +

'}';

}

}

## HikeDetail.java

public class HikeDetail extends AppCompatActivity {

public TextView h\_id, h\_name, h\_location, h\_date, h\_length, h\_level, h\_description, h\_parking,h\_vehicle;

public DBHelper dbHelper;

public int id;

public Button btnDelete, btnBack;

public Hike hike;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.hike\_detail);

//

mapping();

//

dbHelper = new DBHelper(this);

//

Intent i = getIntent();

id = i.getIntExtra("hikeID", 0);

LoadData();

btnDelete.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

dbHelper.deleteHike(id);

Intent resultIntent = new Intent();

resultIntent.putExtra("deletedHikeId", id);

setResult(RESULT\_OK, resultIntent);

finish();

}

});

btnBack.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

}

public void mapping(){

h\_id = findViewById(R.id.tv\_id);

h\_name = findViewById(R.id.tv\_name);

h\_location = findViewById(R.id.tv\_location);

h\_date = findViewById(R.id.tv\_date\_of\_hike);

h\_length = findViewById(R.id.tv\_length\_the\_hike);

h\_level = findViewById(R.id.tv\_level\_of\_difficulty);

h\_vehicle = findViewById(R.id.tv\_vehicle);

h\_description = findViewById(R.id.tv\_description);

h\_parking = findViewById(R.id.tv\_parking\_available);

btnDelete = findViewById(R.id.btn\_delete);

btnBack = findViewById(R.id.btn\_back);

}

private void LoadData() {

hike = dbHelper.getHikeById(id);

if (hike != null) {

h\_id.setText(String.valueOf(hike.getId()));

h\_name.setText(hike.getName());

h\_location.setText(hike.getLocation());

h\_date.setText(hike.getDate());

h\_description.setText(hike.getDescription());

h\_level.setText(hike.getLevel());

h\_vehicle.setText(hike.getVehicle());

h\_length.setText(String.valueOf(hike.getLength()));

h\_parking.setText(hike.isParking() == 1 ? "Yes" : "No");

}

}

}

## MainHike.java

public class MainHike extends AppCompatActivity {

private FloatingActionButton btnPlus;

private RecyclerView rcvHike;

private HikeAdapter hikeAdapter;

private DBHelper db;

private ArrayList<Hike> hikeList = new ArrayList<>();

private ActionBar actionBar;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main\_hike);

actionBar = getSupportActionBar();

btnPlus = findViewById(R.id.btnPlus);

rcvHike = findViewById(R.id.rcvHike);

//

db = new DBHelper(this);

//

btnPlus.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(MainHike.this, AddHike.class);

startActivityForResult(i, 0);

}

});

refreshHikeList();

}

private void refreshHikeList() {

hikeList.clear();

hikeList.addAll(db.getAllHike());

hikeAdapter = new HikeAdapter(this, db.getAllHike(), MainHike.this);

rcvHike.setAdapter(hikeAdapter);

hikeAdapter.notifyDataSetChanged();

}

@Override

protected void onResume() {

super.onResume();

refreshHikeList();

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == 0 && resultCode == RESULT\_OK) {

refreshHikeList();

}

if (requestCode == 1 && resultCode == RESULT\_OK) {

int deletedHikeId = data.getIntExtra("deletedHikeId", -1);

if (deletedHikeId != -1) {

for (Hike hike : hikeList) {

if (hike.getId() == deletedHikeId) {

hikeList.remove(hike);

break;

}

}

hikeAdapter.notifyDataSetChanged();

}

}

if (requestCode == 2 && resultCode == RESULT\_OK) {

refreshHikeList();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.main\_top\_menu, menu);

MenuItem item = menu.findItem(R.id.searchHike);

SearchView searchView = (SearchView) item.getActionView();

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {

@Override

public boolean onQueryTextSubmit(String query) {

searchHike(query);

return true;

}

@Override

public boolean onQueryTextChange(String query) {

searchHike(query);

return true;

}

});

return true;

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

int id = item.getItemId();

if (id == R.id.deleteAllHike) {

db.deleteAllHike();

Toast.makeText(this, "Delete All!!", Toast.LENGTH\_SHORT).show();

onResume();

return true;

}

return super.onOptionsItemSelected(item);

}

private void searchHike(String query) {

ArrayList<Hike> searchResults = db.getSearchHike(query);

hikeAdapter = new HikeAdapter(this, searchResults, MainHike.this);

rcvHike.setAdapter(hikeAdapter);

hikeAdapter.notifyDataSetChanged();

}

}

## MainObservation.java

public class MainObservation extends AppCompatActivity {

private FloatingActionButton btnPlus, btnBack;

private RecyclerView recyclerViewObservation;

private ObservationAdapter observationAdapter;

private DBHelper dbHelper;

private ArrayList<Observation> Observations = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main\_observation);

btnPlus = findViewById(R.id.btnPlus);

btnBack = findViewById(R.id.back);

btnBack.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

recyclerViewObservation = findViewById(R.id.rcvObservation);

dbHelper = new DBHelper(this);

Intent intent = getIntent();

int hikeID = intent.getIntExtra("hikeID", -1);

btnPlus.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(MainObservation.this, AddObservation.class);

i.putExtra("hikeID", hikeID);

startActivityForResult(i, 0);

}

});

refreshList();

}

private void refreshList() {

Observations.clear();

Intent intent = getIntent();

int hikeID = intent.getIntExtra("hikeID", -1);

if (hikeID != -1) {

ArrayList<Observation> observationList = dbHelper.getObservationsForHike(hikeID);

observationAdapter = new ObservationAdapter(this, observationList, this);

recyclerViewObservation.setAdapter(observationAdapter);

observationAdapter.notifyDataSetChanged();

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == 0 && resultCode == RESULT\_OK){

refreshList();

}

if (requestCode == 1 && resultCode == RESULT\_OK) {

refreshList();

}

if (requestCode == 2 && resultCode == RESULT\_OK){

int deleteObservationId = data.getIntExtra("deleteById", -1);

if (deleteObservationId != -1){

for (Observation observation : Observations){

if (observation.getId() == deleteObservationId){

Observations.remove(observation);

break;

}

}

refreshList();

}

}

}

}

## ObservationDetail.java

public class ObservationDetail extends AppCompatActivity {

private TextView o\_observation, o\_date, o\_comment,o\_id;

private Button btnDelete, btnCancel;

private DBHelper db;

private ArrayList<Observation> obList = new ArrayList<>();

private int id;

Observation ob;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.observation\_detail);

o\_observation = findViewById(R.id.tv\_observation);

o\_id = findViewById(R.id.tv\_id);

o\_date = findViewById(R.id.tv\_date\_of\_time);

o\_comment = findViewById(R.id.tv\_comment);

btnDelete = findViewById(R.id.btn\_delete);

btnCancel = findViewById(R.id.btn\_back);

db = new DBHelper(this);

Intent i = getIntent();

id = i.getIntExtra("observationID", 0);

loadData();

btnDelete.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

db.deleteObservation(id);

Intent i = new Intent();

i.putExtra("deleteById", id);

setResult(RESULT\_OK, i);

finish();

}

});

btnCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

}

private void loadData() {

ob = db.getObservationById(id);

if (ob != null) {

o\_id.setText(String.valueOf(ob.getId()));

o\_comment.setText(ob.getComment());

o\_observation.setText(ob.getObservation());

o\_date.setText(ob.getDateOfTime());

}

}

}

## add\_hike.xml

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/LavenderBlush1"

tools:context=".activity.AddHike">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="10dp"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:textColor="#000000"

android:hint="Add new hike"

android:textSize="40px" />

<EditText

android:id="@+id/name"

android:layout\_width="match\_parent"

android:textColor="#000000"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:hint="Name of hike"

android:inputType="text" />

<EditText

android:id="@+id/location"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:layout\_marginBottom="10dp"

android:hint="Location"

android:inputType="text" />

<EditText

android:id="@+id/date"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:textColor="#000000"

android:hint="Date of hike"

android:inputType="text"

/>

<EditText

android:id="@+id/length"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:layout\_marginBottom="10dp"

android:hint="Length the hike"

android:inputType="number"

/>

<EditText

android:id="@+id/description"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:layout\_marginBottom="10dp"

android:hint="Description"

android:inputType="text"

/>

<Switch

android:id="@+id/parking"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:text="Parking Available"

android:textSize="40px" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingEnd="8dp"

android:text="Level: "

android:textColor="#000000"

android:textSize="45px" />

<Spinner

android:id="@+id/level"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:entries="@array/level"

android:hint="Select Difficulty"

tools:ignore="TouchTargetSizeCheck" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingEnd="8dp"

android:text="Vehicle: "

android:textSize="45px"

android:textColor="#000000"

/>

<Spinner

android:id="@+id/vehicle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:entries="@array/vehicle"

android:hint="Select Difficulty"

tools:ignore="TouchTargetSizeCheck" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="horizontal">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_add"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/gradient\_normal"

android:text="Add" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_cancel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/btn\_cancel"

android:text="Cancel" />

</LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

## add\_observation.xml

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/LavenderBlush1"

tools:context=".activity.AddHike">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.261">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="60px"

android:gravity="center"

android:hint="Add Observation"/>

<EditText

android:id="@+id/observation"

android:layout\_width="match\_parent"

android:textColor="#000000"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="20dp"

android:hint="Observation"

android:inputType="text" />

<EditText

android:id="@+id/comment"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="20dp"

android:textColor="#000000"

android:hint="Comment"

android:inputType="text" />

<EditText

android:id="@+id/date"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="20dp"

android:textColor="#000000"

android:hint="Date of time"

android:inputType="text" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="horizontal">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_add"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/gradient\_normal"

android:text="Add" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_cancel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/btn\_cancel"

android:text="Cancel" />

</LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

## edit\_hike.xml

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/LavenderBlush1"

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="10dp"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:textColor="#000000"

android:hint="Edit hike"

android:textSize="40px" />

<EditText

android:id="@+id/name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:hint="Name of hike"

android:textColor="#000000"

android:inputType="text" />

<EditText

android:id="@+id/location"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:hint="Location"

android:textColor="#000000"

android:inputType="text" />

<EditText

android:id="@+id/date"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:hint="Date of hike"

android:textColor="#000000"

android:inputType="text" />

<EditText

android:id="@+id/length"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:hint="Length the hike"

android:textColor="#000000"

android:inputType="number" />

<EditText

android:id="@+id/description"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:hint="Description"

android:textColor="#000000"

android:inputType="text" />

<Switch

android:id="@+id/parking"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:text="Parking Available"

android:textSize="40px" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingEnd="8dp"

android:text="Level: "

android:textColor="#000000"

android:textSize="45px"/>

<Spinner

android:id="@+id/level"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:entries="@array/level"

android:hint="Select Difficulty"

tools:ignore="TouchTargetSizeCheck" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingEnd="8dp"

android:text="Vehicle: "

android:textSize="45px"

android:textColor="#000000"

/>

<Spinner

android:id="@+id/vehicle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:entries="@array/vehicle"

android:hint="Select Difficulty"

tools:ignore="TouchTargetSizeCheck" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="horizontal">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_save"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/gradient\_normal"

android:text="Save" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_cancel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/btn\_cancel"

android:text="Cancel" />

</LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

## edit\_observation.xml

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/LavenderBlush1"

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.261">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:hint="Edit observation"

android:textSize="60px" />

<EditText

android:id="@+id/observation"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="20dp"

android:hint="Observation"

android:inputType="text" />

<EditText

android:id="@+id/comment"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="20dp"

android:hint="Comment"

android:inputType="text"/>

<EditText

android:id="@+id/date"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="20dp"

android:hint="Date of time"

android:inputType="text" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="horizontal">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_save"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/gradient\_normal"

android:text="Save" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_cancel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/btn\_cancel"

android:text="Cancel" />

</LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

## hike\_detail.xml

<LinearLayout xmlns:

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/Lavender"

android:orientation="vertical">

<ImageView

android:id="@+id/ivImage"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:padding="10dp"

android:src="@drawable/ic\_launcher\_background"

tools:srcCompat="@tools:sample/avatars" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:layout\_marginEnd="5dp"

android:orientation="vertical"

android:padding="10dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/id"

android:textColor="#000000"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hike ID: "

android:textSize="20dp"

android:textStyle="bold"/>

<TextView

android:id="@+id/tv\_id"

android:layout\_width="match\_parent"

android:textColor="#000000"

android:layout\_height="wrap\_content"

android:textSize="20dp" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:text="Name: "

android:textSize="20dp" />

<TextView

android:id="@+id/tv\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/location"

android:layout\_width="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:layout\_height="wrap\_content"

android:text="Location: "

android:textSize="20dp" />

<TextView

android:id="@+id/tv\_location"

android:layout\_width="match\_parent"

android:textColor="#000000"

android:layout\_height="wrap\_content"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/date\_of\_hike"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:text="Date of hike: "

android:textSize="20dp"/>

<TextView

android:id="@+id/tv\_date\_of\_hike"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/parking\_available"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:text="Parking available: "

android:textSize="20dp" />

<TextView

android:id="@+id/tv\_parking\_available"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/length\_the\_hike"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:text="Length the hike: "

android:textSize="20dp" />

<TextView

android:id="@+id/tv\_length\_the\_hike"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" m"

android:textColor="#000000"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/level\_of\_difficulty"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Level of difficulty: "

android:textColor="#000000"

android:textStyle="bold"

android:textSize="20dp" />

<TextView

android:id="@+id/tv\_level\_of\_difficulty"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="20dp"

android:textColor="#000000"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/description"

android:layout\_width="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:layout\_height="wrap\_content"

android:text="Description: "

android:textSize="20dp"/>

<TextView

android:id="@+id/tv\_description"

android:layout\_width="match\_parent"

android:textColor="#000000"

android:layout\_height="wrap\_content"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/vehicle"

android:layout\_width="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:layout\_height="wrap\_content"

android:text="Vehicle: "

android:textSize="20dp" />

<TextView

android:id="@+id/tv\_vehicle"

android:layout\_width="match\_parent"

android:textColor="#000000"

android:layout\_height="wrap\_content"

android:textSize="20dp" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="horizontal">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_back"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/btn\_cancel"

android:text="Back" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_delete"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:background="@drawable/btn\_delete"

android:text="Delete" />

</LinearLayout>

</LinearLayout>

</LinearLayout>

## item\_hike.xml

<androidx.cardview.widget.CardView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

xmlns:app="http://schemas.android.com/apk/res-auto"

app:cardBackgroundColor="@color/LavenderBlush2"

app:cardElevation="12dp"

app:cardCornerRadius="16dp"

android:layout\_margin="16dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

android:orientation="vertical">

<LinearLayout

android:layout\_gravity="end"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:textColor="#000000"

android:text="Name: "

android:textStyle="bold"

android:gravity="start"

android:textSize="20sp"/>

<TextView

android:id="@+id/name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:layout\_marginBottom="10dp"

android:text=""

android:textSize="20sp"

android:layout\_weight="1" />

<TextView

android:id="@+id/date"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:text="12"

android:textColor="#EE2C2C"

android:gravity="end"

android:textSize="20sp"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:textColor="#000000"

android:text="Location: "

android:textStyle="bold"

android:textSize="20sp" />

<TextView

android:id="@+id/location"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:textColor="#000000"

android:textSize="20sp"

android:layout\_weight="1" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:gravity="center"

android:orientation="horizontal">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_observation"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/gradient\_normal"

android:text="Observation">

</androidx.appcompat.widget.AppCompatButton>

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_edit"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/gradient\_normal"

android:layout\_marginStart="20dp"

android:text="Edit">

</androidx.appcompat.widget.AppCompatButton>

</LinearLayout>

</LinearLayout>

</androidx.cardview.widget.CardView>

## item\_observation.xml

<androidx.cardview.widget.CardView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

xmlns:app="http://schemas.android.com/apk/res-auto"

app:cardBackgroundColor="@color/LavenderBlush2"

app:cardCornerRadius="16dp"

android:layout\_margin="16dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

android:orientation="vertical">

<LinearLayout

android:layout\_gravity="end"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:layout\_marginBottom="10dp"

android:text="Observation: "

android:gravity="start"

android:textSize="20sp"/>

<TextView

android:id="@+id/observation"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:layout\_marginBottom="10dp"

android:text=""

android:textSize="20sp"

android:layout\_weight="1" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:layout\_marginBottom="10dp"

android:text="Comment: "

android:textSize="20sp" />

<TextView

android:id="@+id/comment"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:layout\_marginBottom="10dp"

android:textSize="20sp"

android:layout\_weight="1" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:textColor="#000000"

android:textStyle="bold"

android:text="Date of time: "

android:textSize="20sp" />

<TextView

android:id="@+id/date"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:text="12"

android:textColor="#000000"

android:gravity="end"

android:textSize="20sp"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:gravity="center"

android:orientation="horizontal">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_edit"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/gradient\_normal"

android:layout\_marginStart="20dp"

android:text="Edit">

</androidx.appcompat.widget.AppCompatButton>

</LinearLayout>

</LinearLayout>

</androidx.cardview.widget.CardView>

## main\_hike.xml

<RelativeLayout xmlns:

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/Lavender"

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rcvHike"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_marginStart="-3dp"

android:layout\_marginTop="0dp"

android:layout\_marginEnd="3dp"

app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager" />

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/btnPlus"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:backgroundTint="@color/LavenderBlush1"

android:layout\_marginEnd="50dp"

android:layout\_marginBottom="50dp"

android:src="@drawable/baseline\_add\_24"

tools:ignore="ContentDescription">

</com.google.android.material.floatingactionbutton.FloatingActionButton>

</RelativeLayout>

## main\_observation.xml

<RelativeLayout xmlns

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/Lavender"

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rcvObservation"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_marginStart="-3dp"

android:layout\_marginTop="0dp"

android:layout\_marginEnd="3dp"

app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager" />

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/btnPlus"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_marginEnd="50dp"

android:layout\_marginBottom="50dp"

tools:ignore="ContentDescription"

android:backgroundTint="@color/LavenderBlush1"

android:src="@drawable/baseline\_add\_24">

</com.google.android.material.floatingactionbutton.FloatingActionButton>

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/back"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_alignParentBottom="true"

android:layout\_marginStart="50dp"

android:layout\_marginBottom="50dp"

android:clickable="true"

tools:ignore="ContentDescription"

android:backgroundTint="@color/LavenderBlush3"

app:srcCompat="@drawable/baseline\_arrow\_back\_24" />

</RelativeLayout>

## observation\_detail.xml

<LinearLayout

android:layout\_width="match\_parent"

android:orientation="vertical"

android:background="@color/Lavender"

android:layout\_height="match\_parent">

<ImageView

android:id="@+id/ivImage"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:layout\_marginTop="10dp"

android:src="@drawable/ic\_launcher\_background"

tools:srcCompat="@tools:sample/avatars" />

<LinearLayout

android:layout\_width="match\_parent"

android:orientation="vertical"

android:padding="10dp"

android:layout\_marginTop="5dp"

android:layout\_marginEnd="5dp"

android:layout\_height="wrap\_content">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/id"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:text="Observation ID: "

android:textSize="20dp"

/>

<TextView

android:id="@+id/tv\_id"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/observation"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:text="Name: "

android:textSize="20dp"

/>

<TextView

android:id="@+id/tv\_observation"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/comment"

android:layout\_width="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:layout\_height="wrap\_content"

android:text="Location: "

android:textSize="20dp"

/>

<TextView

android:id="@+id/tv\_comment"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/date\_of\_time"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textStyle="bold"

android:text="Date of time: "

android:textSize="20dp"

/>

<TextView

android:id="@+id/tv\_date\_of\_time"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#000000"

android:textSize="20dp" />

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:orientation="horizontal"

android:gravity="center"

android:layout\_height="wrap\_content">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_back"

android:text="Back"

android:background="@drawable/btn\_cancel"

android:layout\_marginHorizontal="10dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/btn\_delete"

android:text="Delete"

android:background="@drawable/btn\_delete"

android:layout\_marginHorizontal="10dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>

</LinearLayout>

## main\_top\_menu.xml

<item

android:id="@+id/searchHike"

android:title="Search"

android:icon="@drawable/ic\_baseline\_search\_24"

app:showAsAction="always"

app:actionViewClass="android.widget.SearchView">

</item>

<item

android:id="@+id/deleteAllHike"

android:title="Delete All"

android:icon="@drawable/baseline\_delete\_forever\_24"

app:showAsAction="always">

</item>

</menu>

## arrayLevel.xml

<resources>

<string-array name="level">

<item>Beginner Hiker</item>

<item>Intermediate Hiker</item>

<item>Advanced Hiker</item>

<item>Expert Hiker</item>

<item>Ultra Hiker</item>

<item>Trail Runner</item>

</string-array>

</resources>

## arrayVehicle.xml

<resources>

<string-array name="vehicle">

<item>Personal Vehicle</item>

<item>Carpooling or Ridesharing</item>

<item>Rental Vehicles</item>

<item>Shuttle Services</item>

<item>Public Transportation</item>

<item>Bicycles</item>

<item>Hiking on Foot</item>

</string-array>

</resources>