Table of Contents

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		

Table 1: CONCISE TABLE

Section 2: SCREEN SHOTS/DESIGN (JAVA)

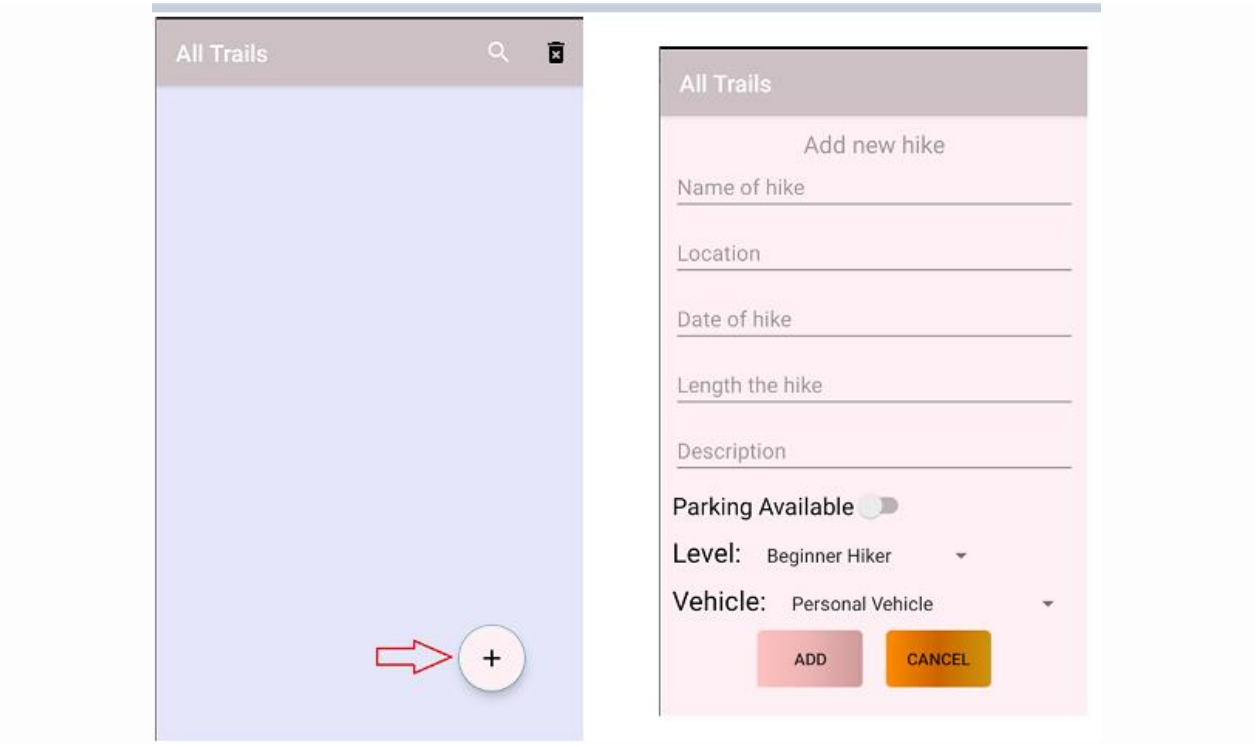

Figure 1: home and add page

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.

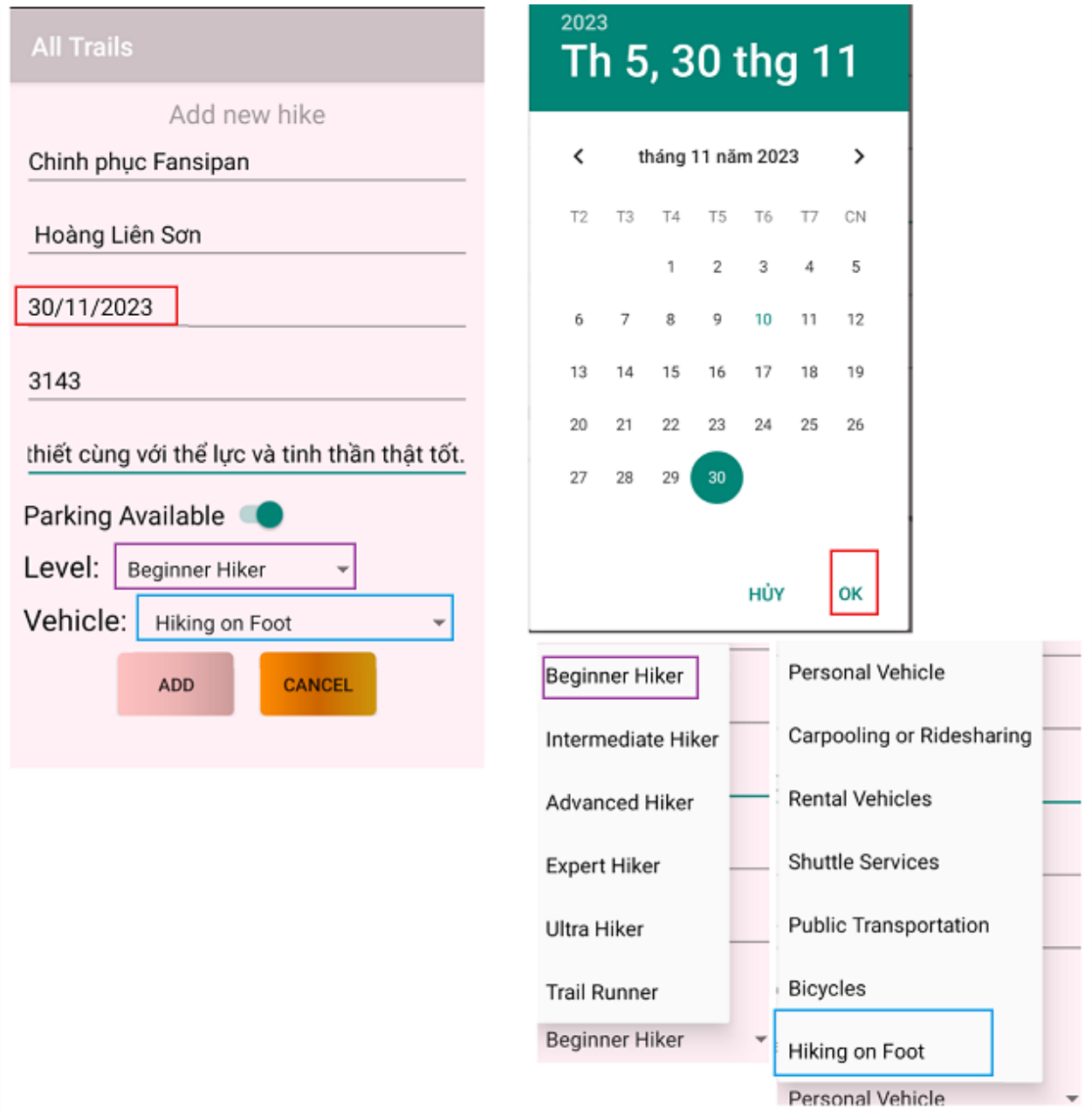

Figure 2: add hike

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.

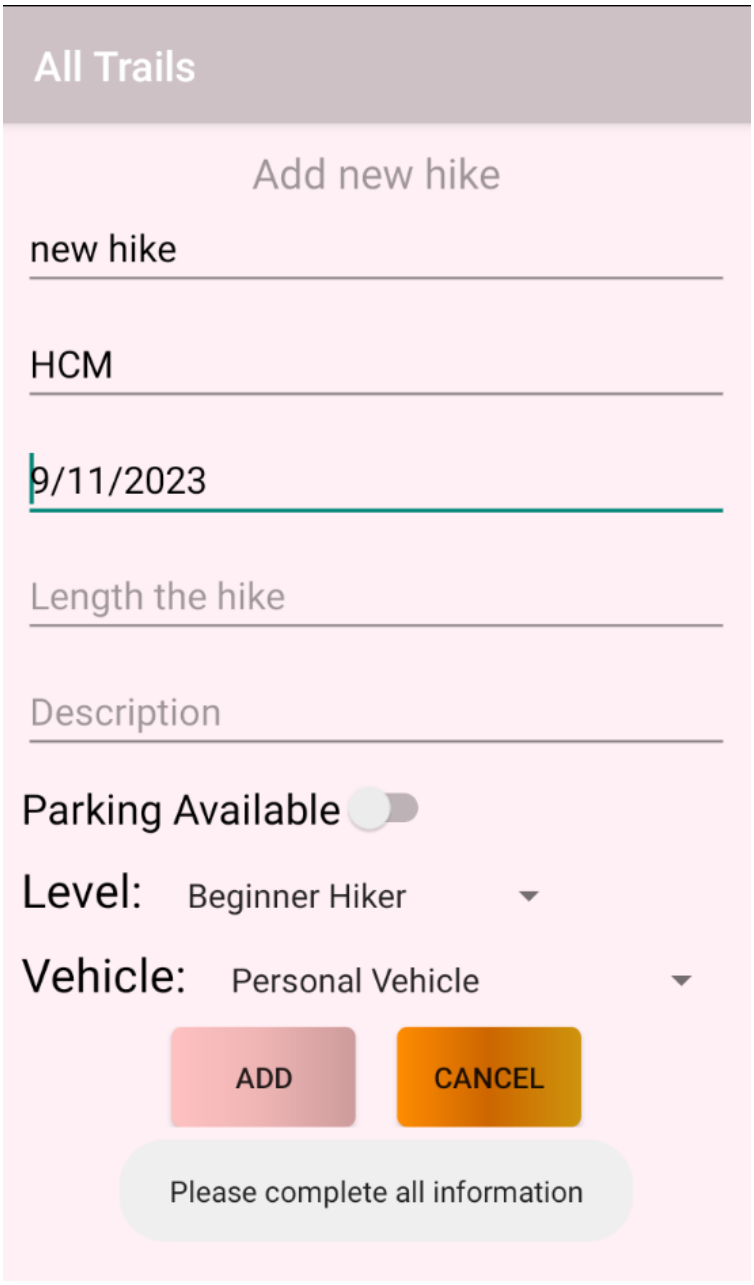

Figure 3: fill in missing information

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.

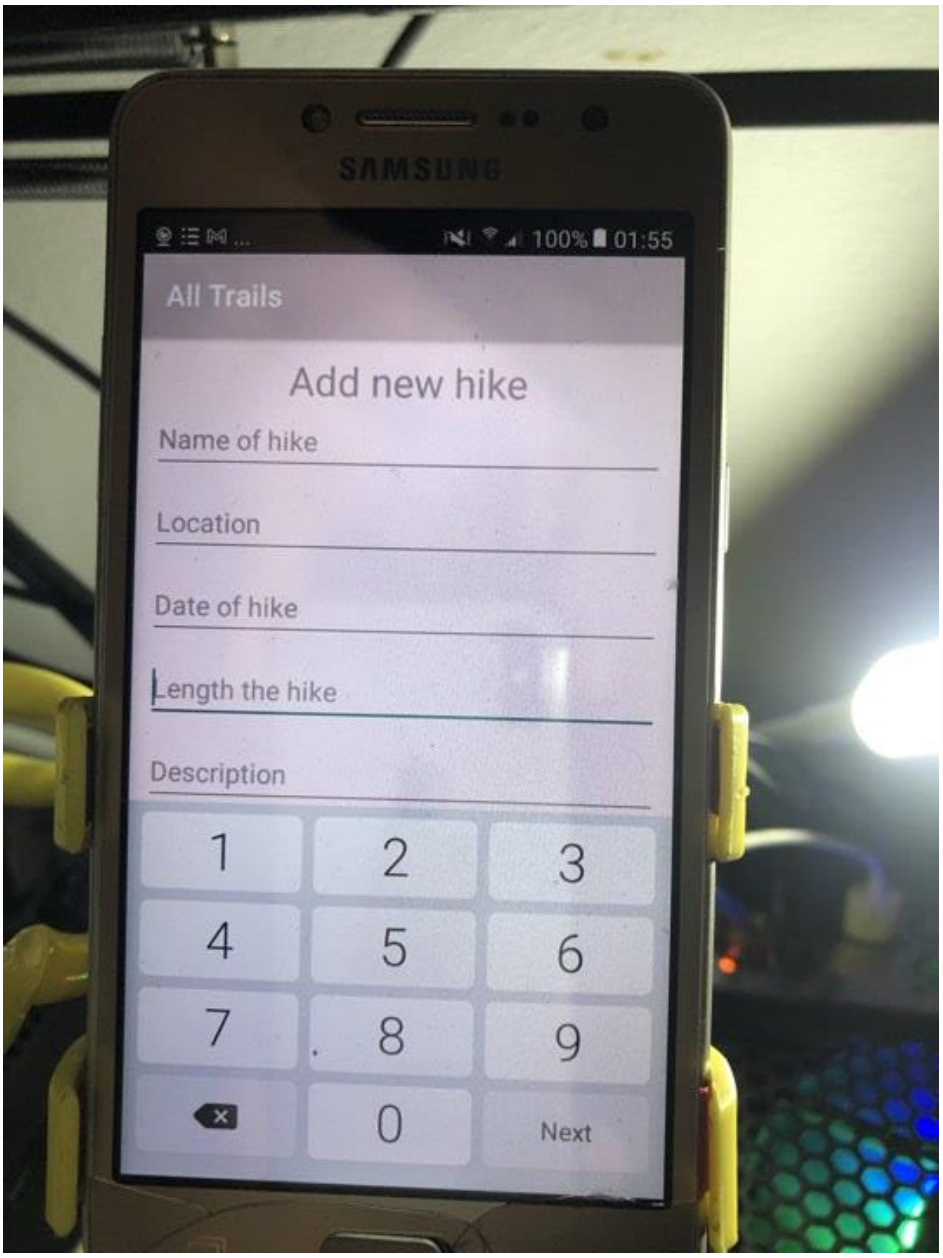

Figure 4: add length the hike

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

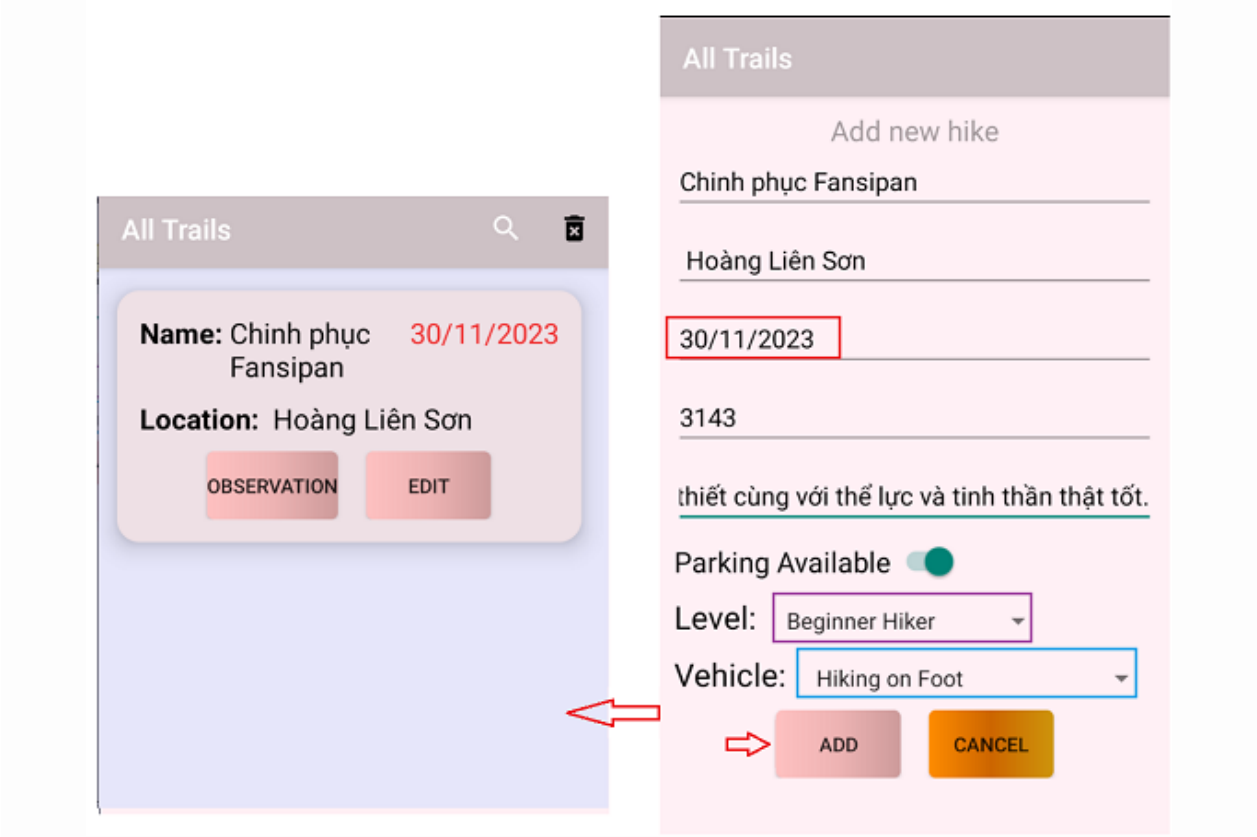

Figure 5: click button add

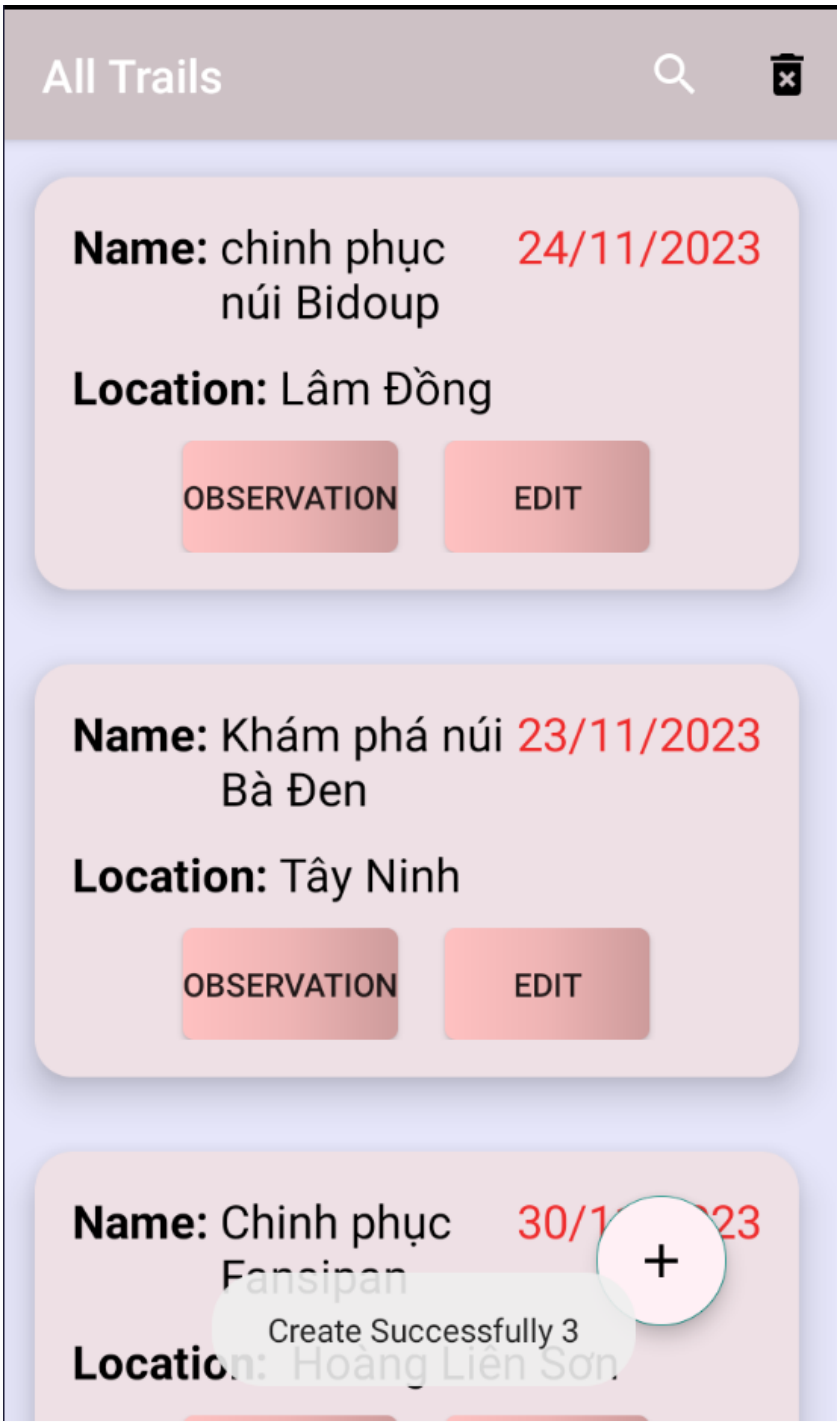

Figure 6: add more

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.

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).

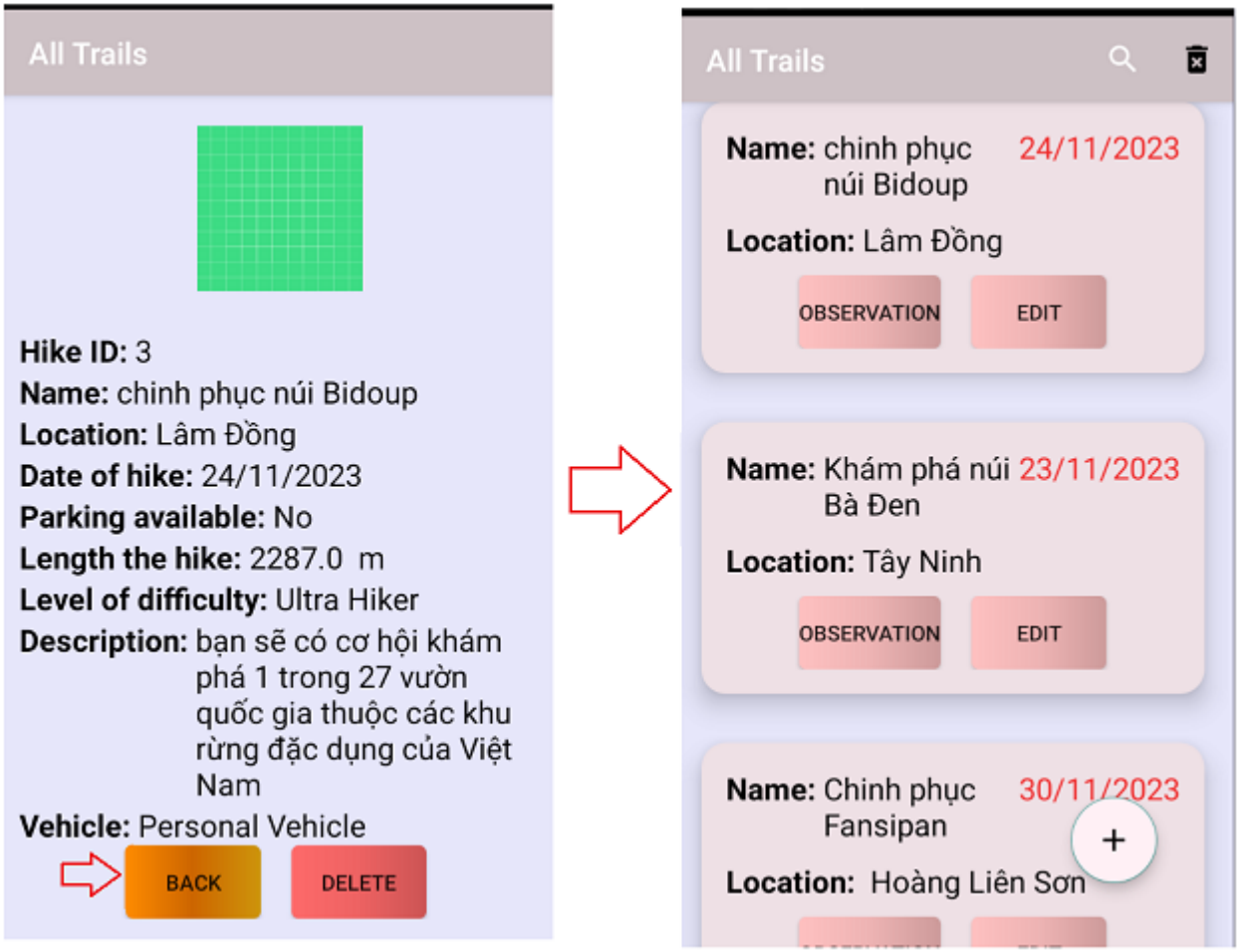

Figure 8: click button back

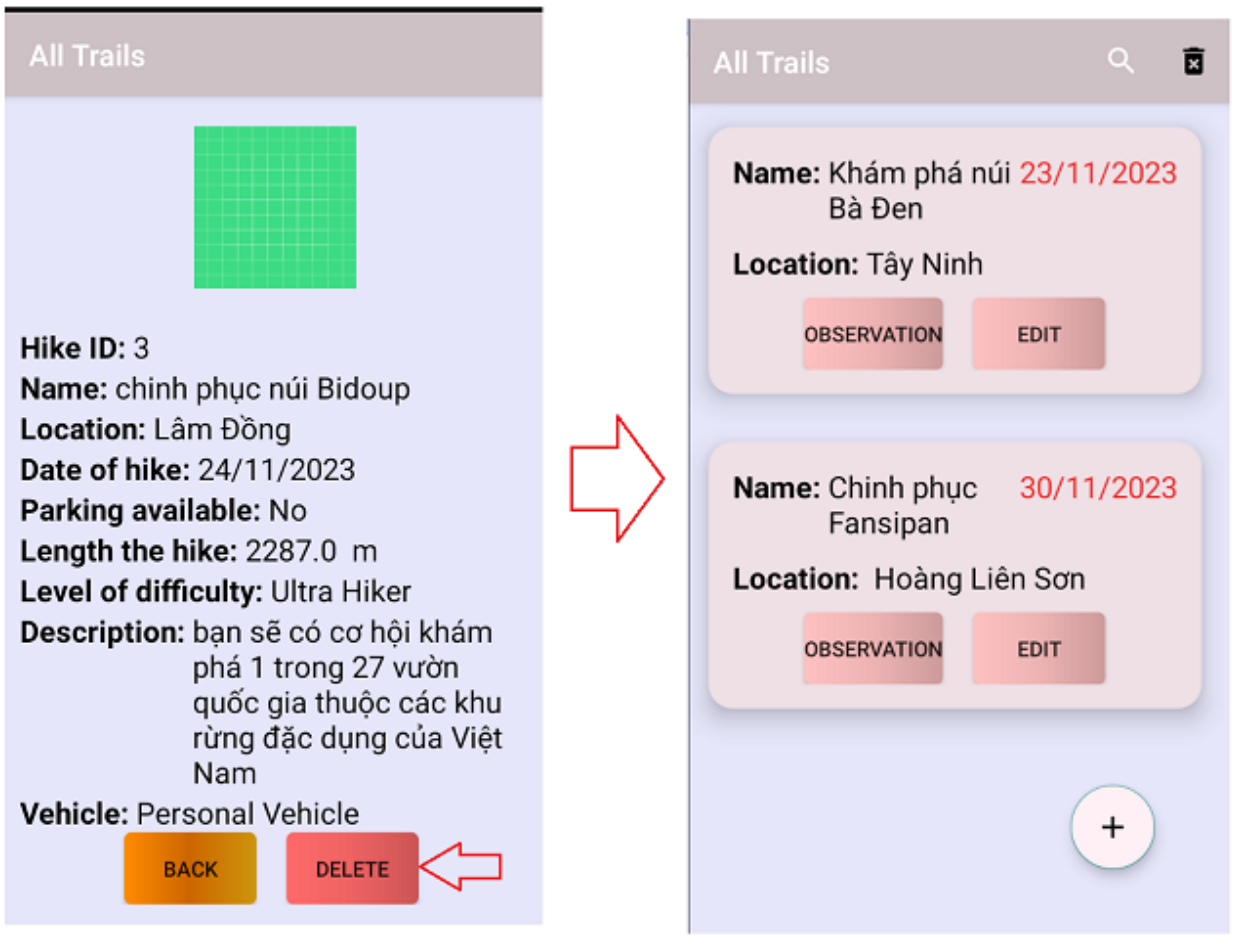

Figure 9: delete hike

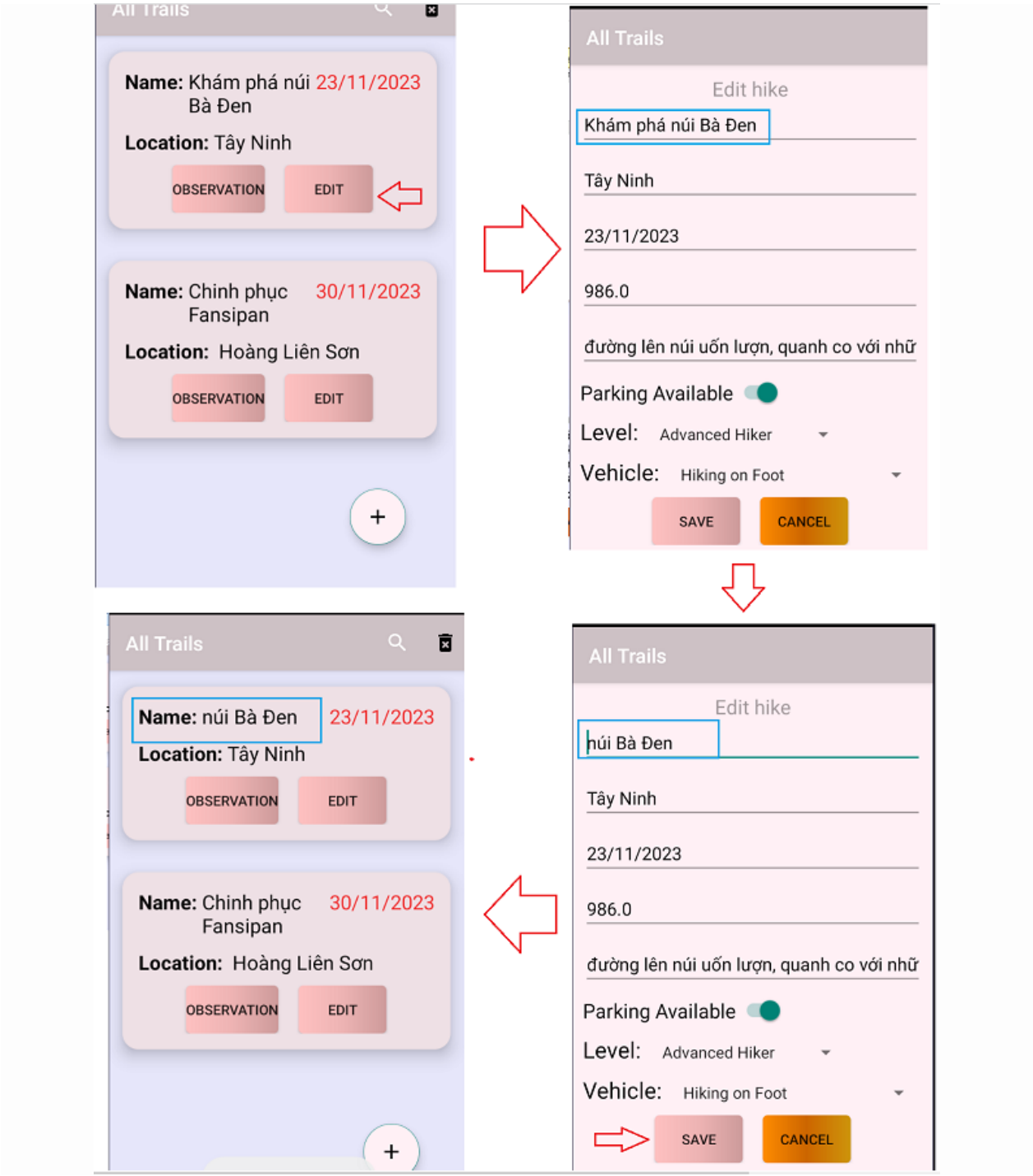

Figure 10: edit hike

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.

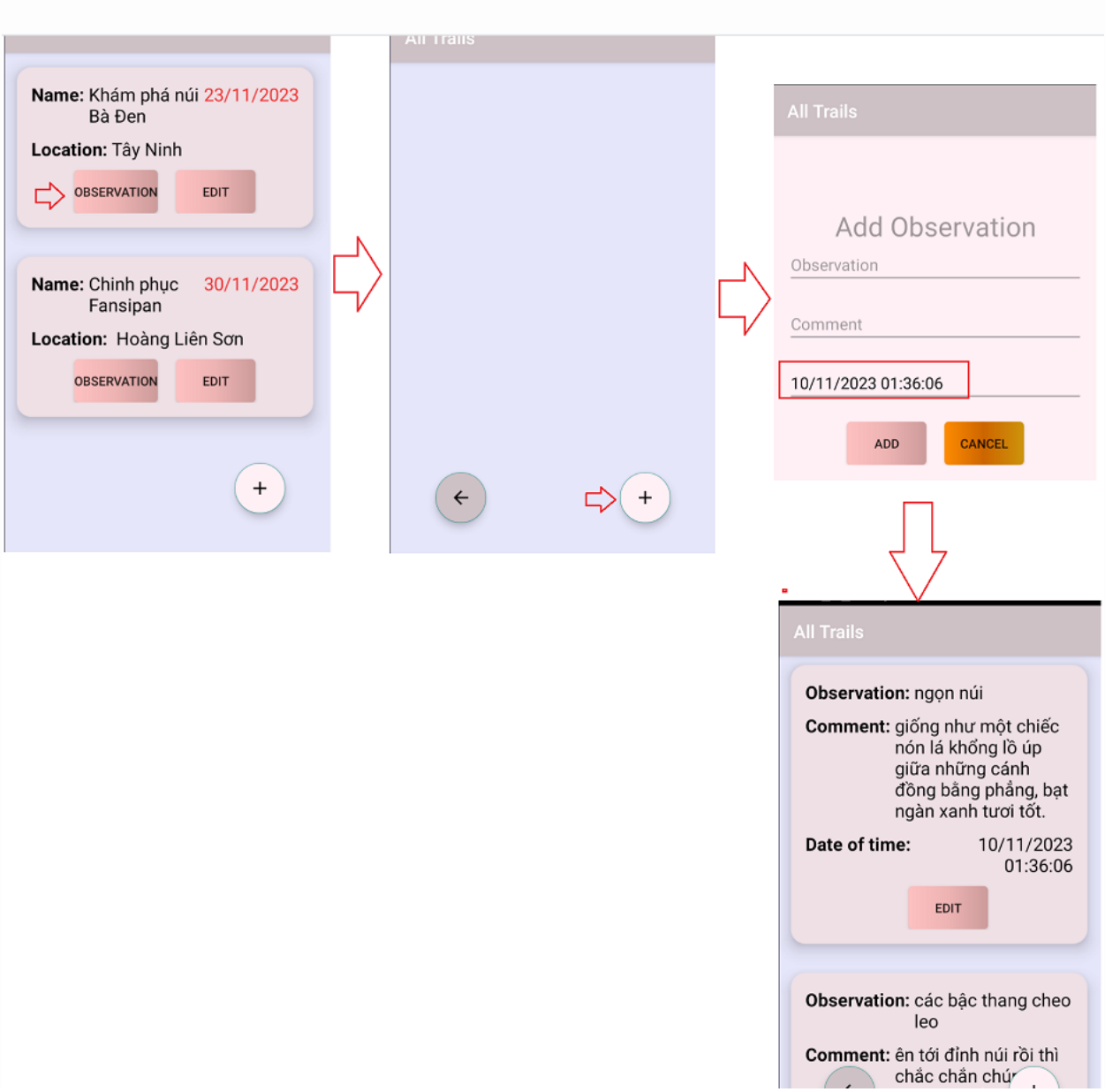

Figure 11: add observation

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.

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).

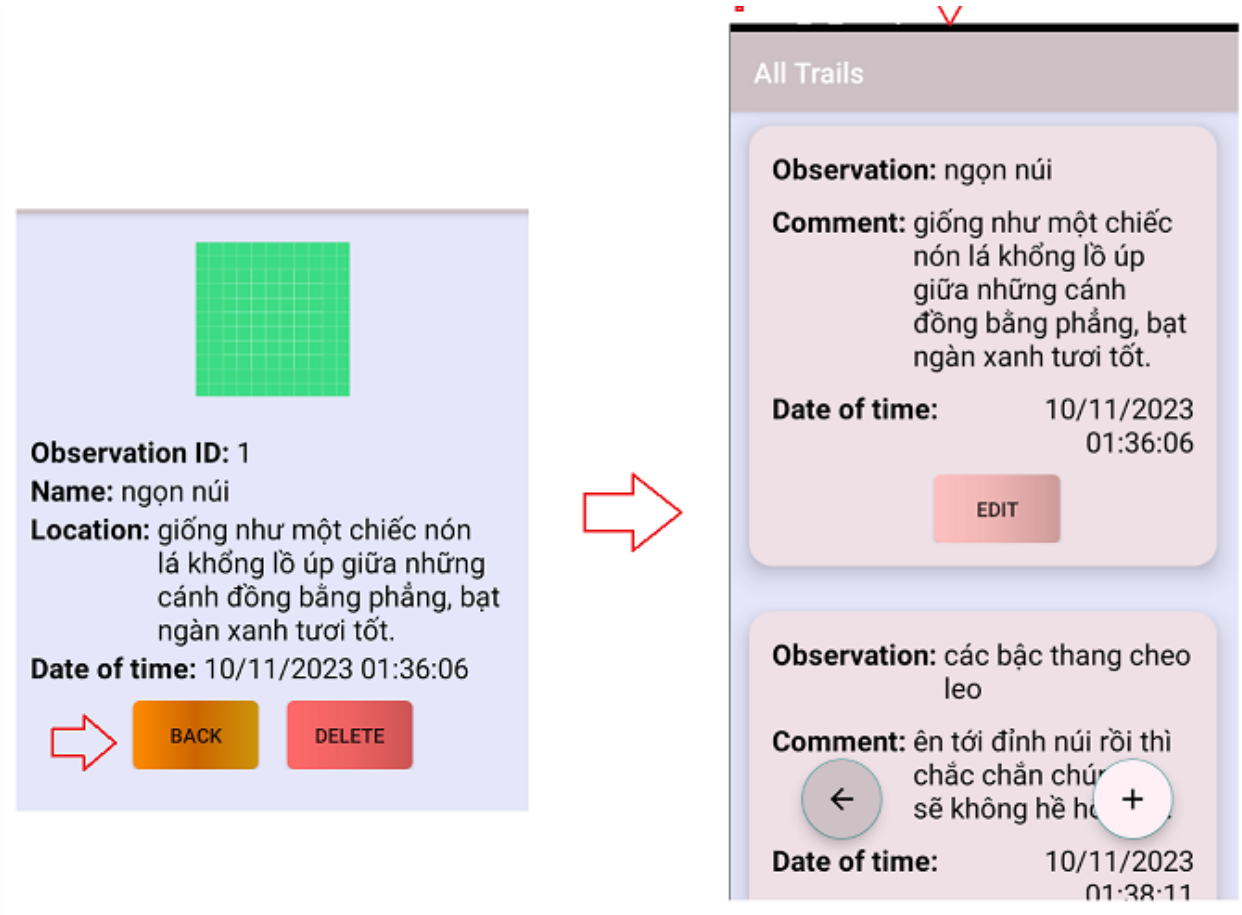

Figure 13: button back in observation

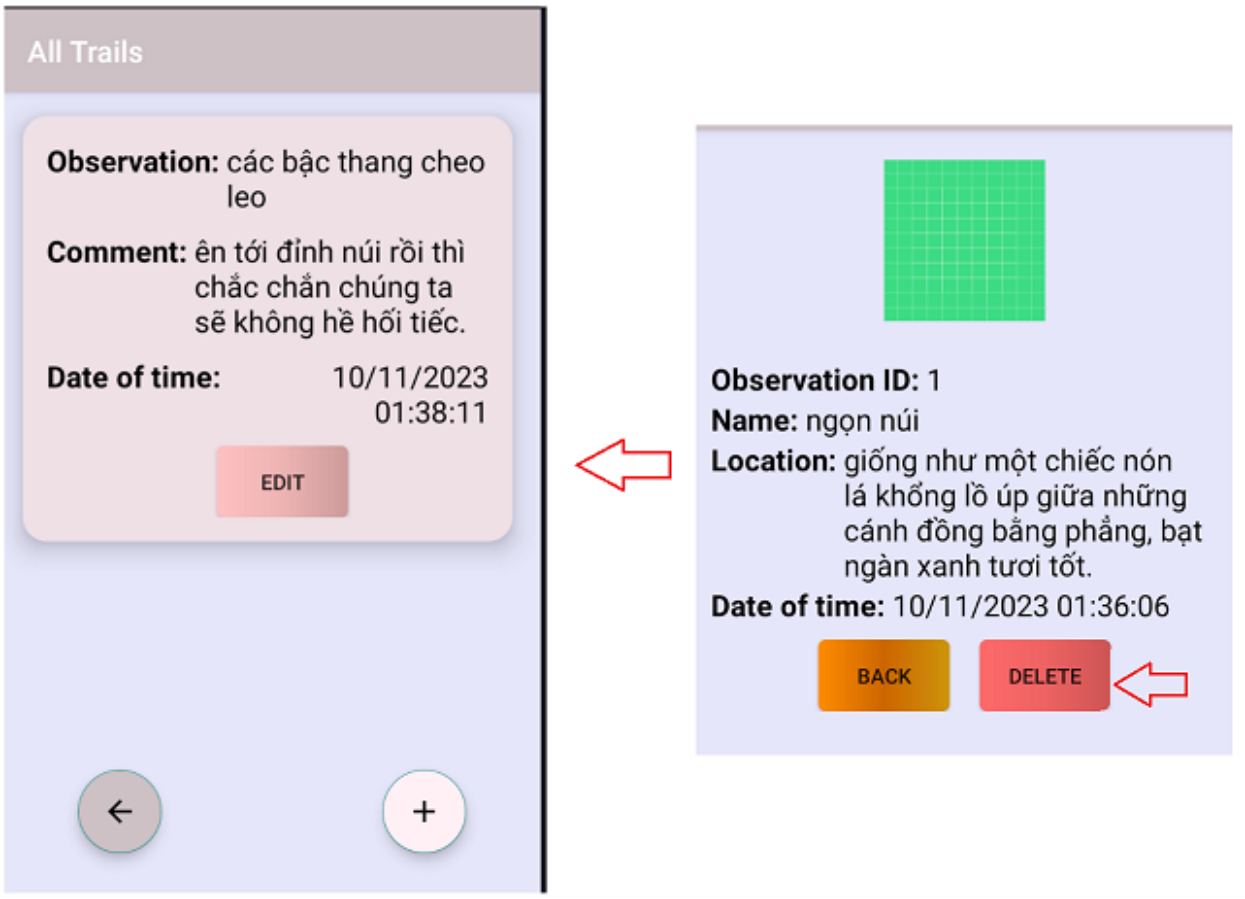

Figure 14: delete observation

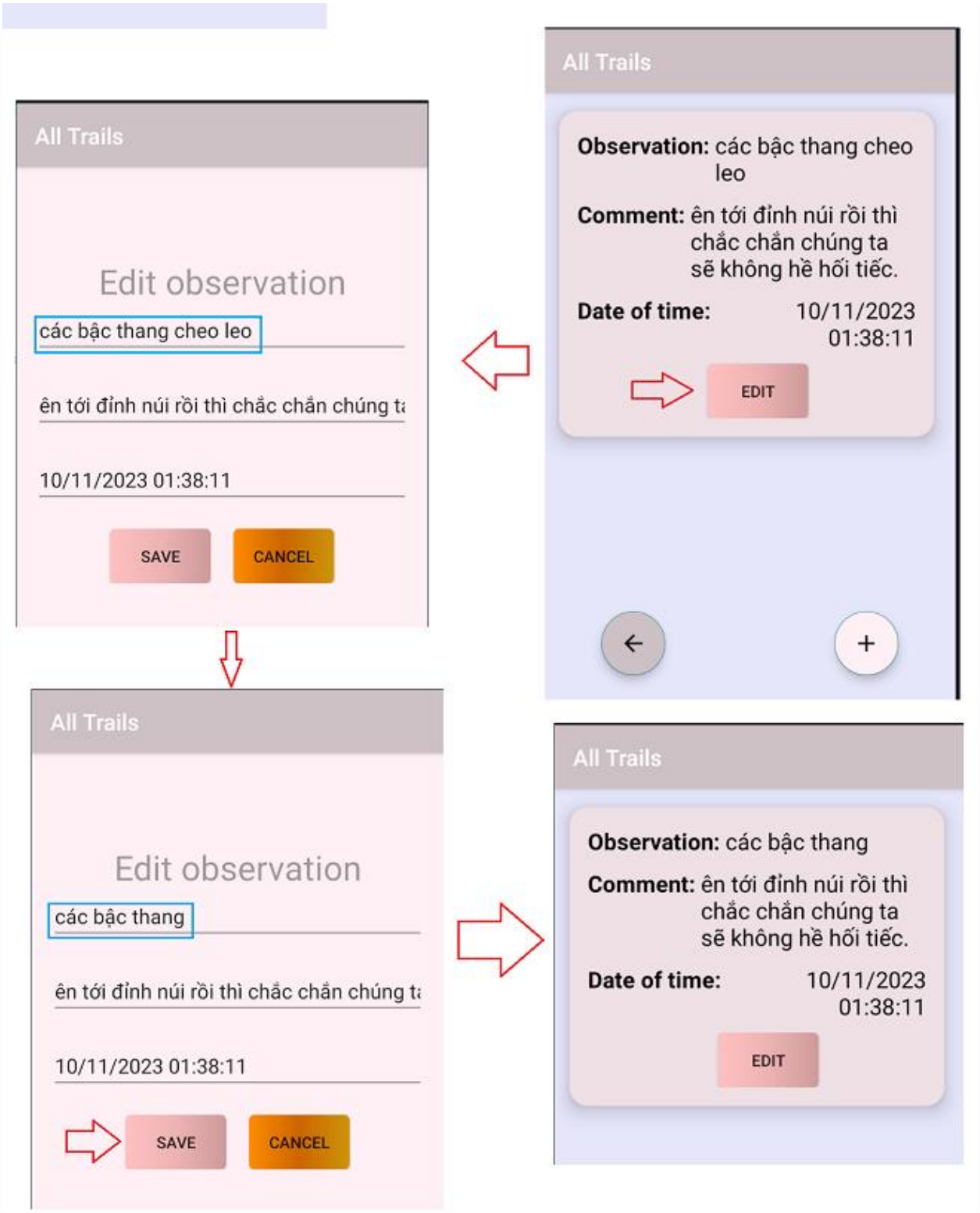

Figure 15: edit observation

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.

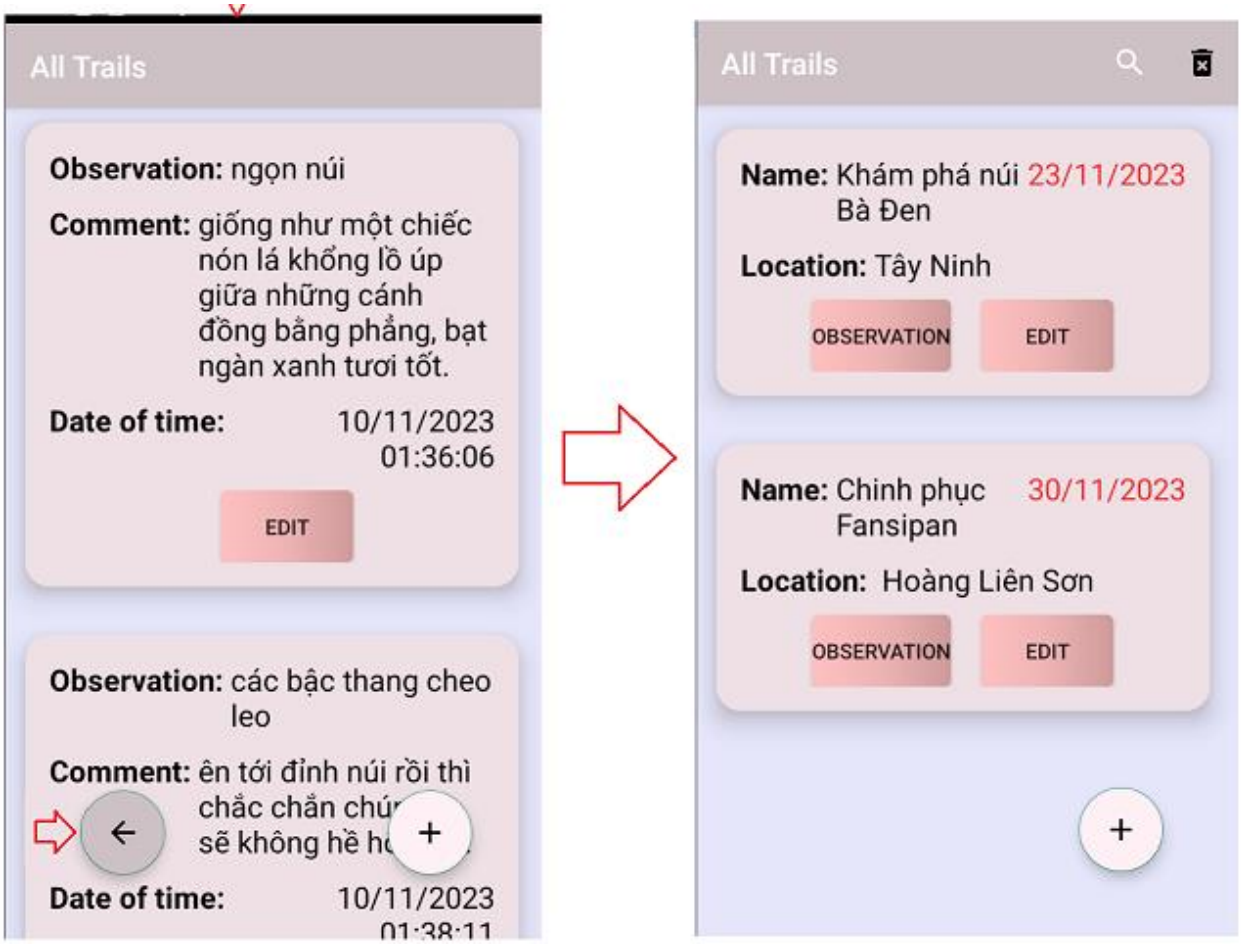

Figure 16: back to home page

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

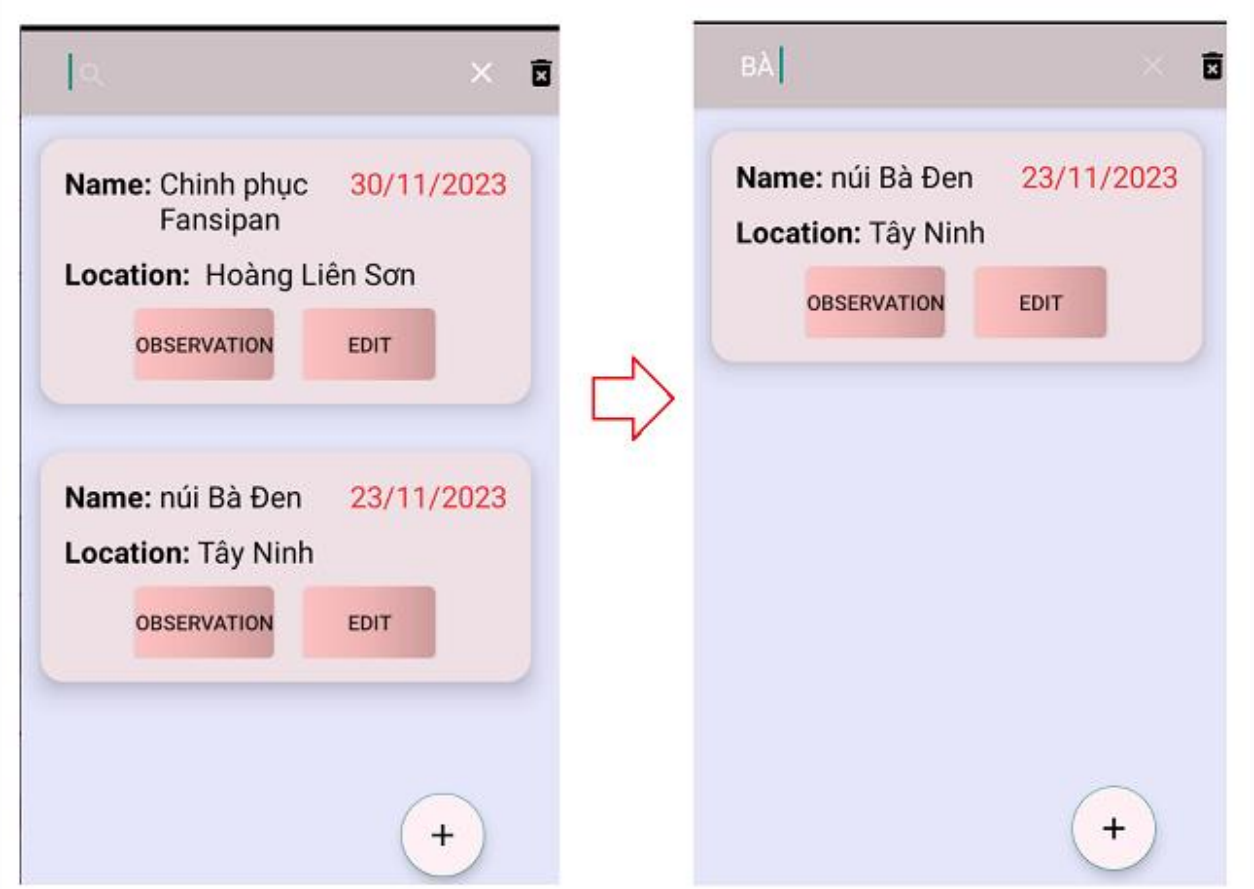

Figure 17: search hike

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.

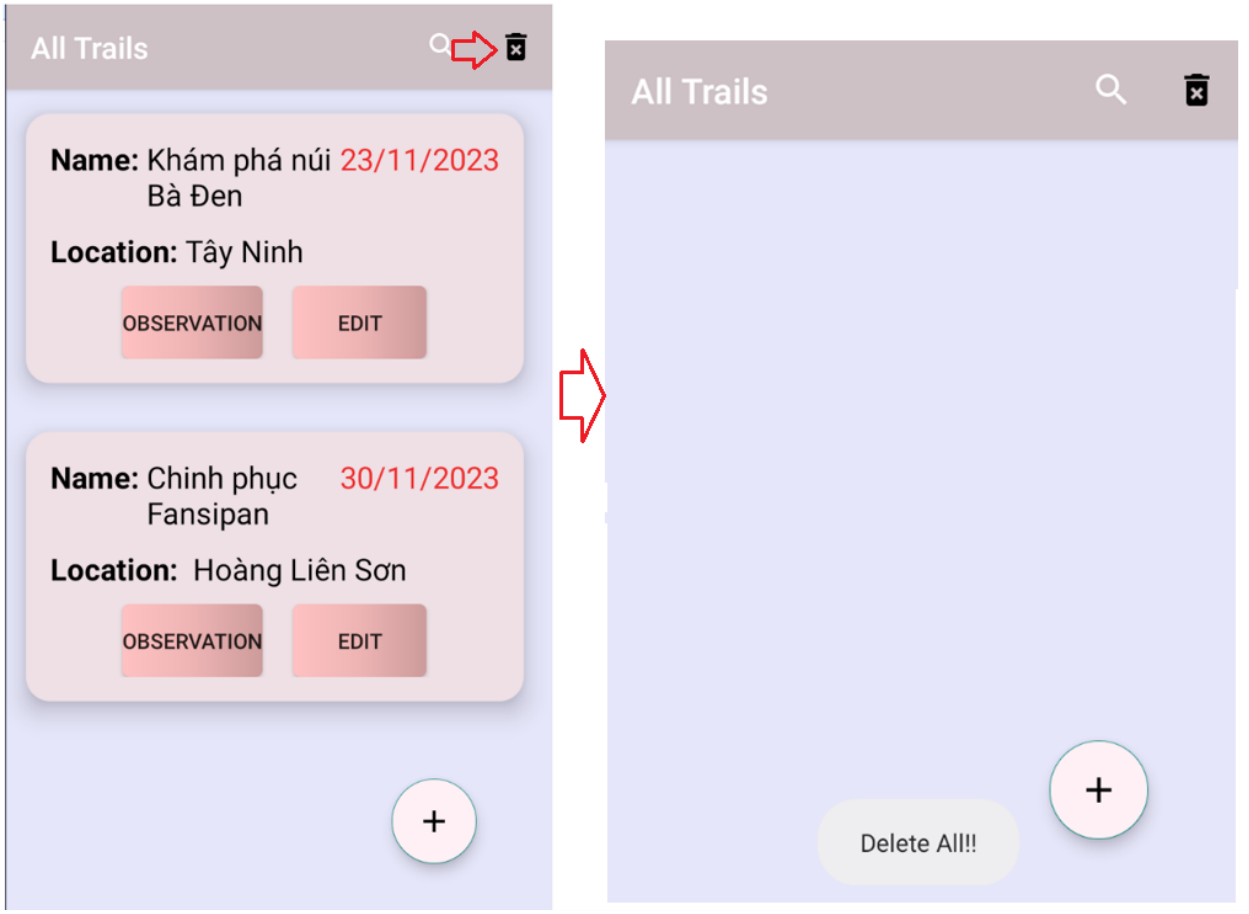

Figure 18: delete all

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

Section 3: REFLECTION

I. 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.

II. 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.

III. 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.

IV. 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

I. 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.

II. 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

III. 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.

IV. 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_DESCRIPTI
ON)));
```

```
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.COL
UMN_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_DESCRIPTI
ON)));

            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.COL
UMN_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.COL
UMN_NAME)));

observation.setDateOfTime(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COL
UMN_DATE)));

observation.setComment(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLU
MN_COMMENT)));

observation.setHikeID(cursor.getInt(cursor.getColumnIndexOrThrow(Observation.COLUMN_FO
REIGN_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.COL
UMN_NAME)));

observation.setDateOfTime(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COL
UMN_DATE)));

observation.setComment(cursor.getString(cursor.getColumnIndexOrThrow(Observation.COLU
MN_COMMENT)));

observation.setHikeID(cursor.getInt(cursor.getColumnIndexOrThrow(Observation.COLUMN_FO
REIGN_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>
```