

Safer Hike Project Report

COMP 8047 – Major Project

Geoffrey Browning – A01053901
9-20-2024

Table of Contents

1.	Introduction	3
1.1.	Student Background	3
Experience.....	3	
Education	3	
1.2.	Acknowledgements	3
1.3.	Project Description	4
1.3.1.	Essential Problems.....	4
1.3.2.	Goals and Objectives.....	4
2.	Body	5
2.1.	Background.....	5
2.2.	Project Statement.....	5
2.3.	Possible Alternative Solutions	5
2.4.	Chosen Solution.....	5
2.5.	Details of Design and Development	6
System design diagram	6	
Class Diagram.....	7	
State Diagrams.....	8	
Encryption	9	
Sequence Diagrams.....	13	
2.5.1	Api Documentation	19
Design Patterns.....	26	
Deliverables.....	27	
2.6.	Testing Details and Results.....	28
Flask Unit Testing	28	
Manual Testing of App	29	
2.7.	Implications of Implementation	42
Technical Implications.....	42	
Practical Implications	43	
2.8.	Innovation.....	43
2.9.	Complexity.....	44
2.10.	Research in New Technologies.....	45
2.11.	Future Enhancements	46

2.12.	Timeline and Milestones	47
3.	Conclusion.....	47
3.1.	Lessons Learned	48
3.2.	Closing Remarks.....	48
4.	Appendix	48
4.1.	Approved Proposal	48
4.2.	Project Supervisor Approvals	48
5.	References	49

1. Introduction

1.1. Student Background

Geoffrey Browning is currently a student of the Bachelor of Technology program at BCIT, majoring in network security and applications development. He previously graduated from the Computer Systems Technology program at BCIT in which his specialized option was artificial intelligence and machine learning.

Experience

COMP-4800 Fortinet Project Practicum: Completed a practicum project for Fortinet as part of a group of four. The project consisted of adding a custom office building tool to Fortinet's virtual training space within the Godot Engine. I was personally responsible for creating a build menu that used raycasting to allow the player to place and manipulate objects within the office space. I added functionality for the player to preview and place a selected object in the build menu as well as to modify its dimensions in real-time.

COMP-3800 ISSP Vanderhoof Chamber Application: Completed a project for the manager of the Vanderhoof Chamber of Commerce which would aid in providing accessibility to local businesses within Vanderhoof throughout the covid pandemic. Through the utilization of dart and flutter within android studio my team created an application that would serve as a local business directory and hiking guide for the citizens of Vanderhoof. While the project itself utilized the google maps API to display local businesses on the directory, I was mainly responsible for the front-end implementation and didn't get any experience using the maps API.

Education

Computer Systems Technology – Diploma | BCIT January 2020 – December 2021

Specialization: Artificial Intelligence and Machine Learning

Computer Systems Technology – Bachelor of Technology | BCIT January 2022 – Current

Specialization: Network Security Applications Development

1.2. Acknowledgements

I would like to give special thanks to my family for their support throughout this process. They took me in when I lost my living arrangements and provided me with the necessary environment to complete this project.

1.3. Project Description

This project involved the development of a mobile application that enhances the safety of individuals going on hikes. The app allows users to create detailed hike plans that include information about the supplies they brought and custom markers detailing waypoints. It tracks users on their hikes and records their traveled path. It also allows the user to register a list of emergency contacts who are to be notified of the hike's details, and the users traveled path, if the user fails to complete a hike within their specified duration. A flask server paired with a PostgreSQL database was built alongside the app to handle secure data storage and emergency alerts. The server allows for alerts to be sent regardless of the status of a user's device.

1.3.1. Essential Problems

The primary problems addressed by this project are:

- The difficulty of communicating with emergency contacts while on a hike
- Limited tools for creating hike plans that could assist in search and rescue efforts
- Lack of reliable tools for hikers to communicate their location during hikes
- The need for automated alerts that notify people if a hiker goes missing

1.3.2. Goals and Objectives

The goals and objectives of this project are:

- To create a mobile app that allows hikers to plan and track their hikes with real-time updates
- To implement emergency contact alert features that automatically send notifications if the user fails to complete a hike
- To offer a user-friendly interface that integrates seamlessly with backend services like Flask and Firebase

2. Body

2.1. Background

The idea for the Safer Hike app emerged from the growing concerns regarding hiker safety, especially in remote areas. Many hiking incidents go unnoticed until it is too late, and the lack of an automated alert system to notify emergency contacts further exacerbates the issue. I frequently embark on solo trips into the Whistler backcountry, and extensive hikes in the Golden Ears Provincial Park. While I do inform family members of my general plans, this information is limited. In the event of a search and rescue operation, the search area can be vast and challenging to pinpoint. This app leverages modern technologies like GPS and cloud services to provide a comprehensive solution to this problem.

2.2. Project Statement

In the context of improving outdoor safety, the Safer Hike app acts as a critical tool for tracking hiker locations, verifying hike completions, and enabling emergency responses if hikers go missing. The project focuses on automating key aspects of safety and simplifying communication between the hiker and their emergency contacts.

2.3. Possible Alternative Solutions

Before proposing the Safer Hike app, alternative solutions considered included:

- A web-based platform that could be used on any device with internet access, though this would not leverage mobile-specific features like GPS and background location tracking.
- A piece of hardware tailored to the problem at hand that would last weeks on a single charge and act as a beacon handling the hike tracking. This hardware would sync to the user's device for hike details submitted via an application.

2.4. Chosen Solution

The chosen solution was to develop a native Android app using Jetpack Compose and the Google Maps API paired with a Flask server and PostgreSQL database. Jetpack Compose was selected for its modern declarative UI framework, which simplified the development of dynamic user interfaces, such as the hike planning and tracking screens. The Google Maps API allowed users to visualize their planned hikes, place custom markers, and track their location.

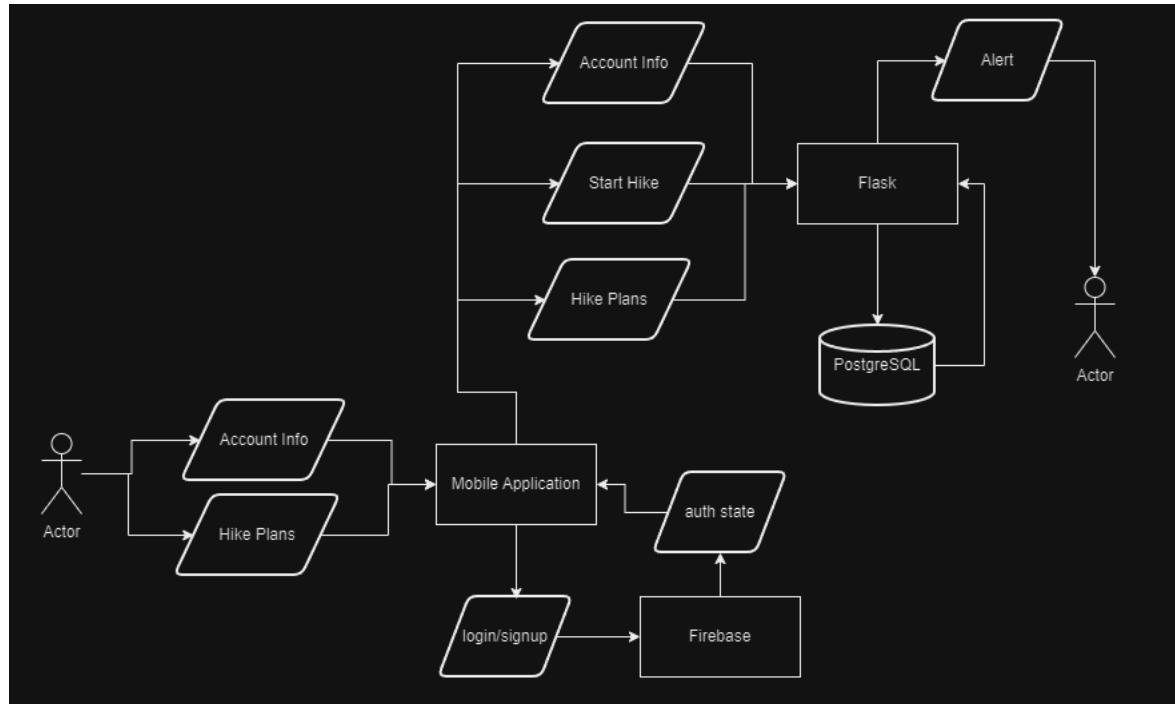
The flask server was used to handle key backend operations like storing journey plans, processing active and completed hikes, and managing emergency alerts. PostgreSQL was chosen as the database for its robustness and ability to handle complex datatypes and queries, particularly for storing location data, emergency contacts, and other hike statistics.

By combining these technologies, the app delivers a seamless user experience for planning, tracking, and managing hikes, while ensuring that the server can handle emergency alerts and location updates efficiently. This solution was ideal because it offered real-time functionality, a flexible and scalable backend, and an intuitive user experience tailored to safety in the outdoors.

2.5. Details of Design and Development

The design and development process for the Safer Hike app followed a structured approach, leveraging modern technologies to address the primary goals of enhancing hiker safety through real-time tracking and automated emergency alerts. This section outlines the deliverables and key artifacts created during the design, analysis, and implementation phases of the project.

System design diagram

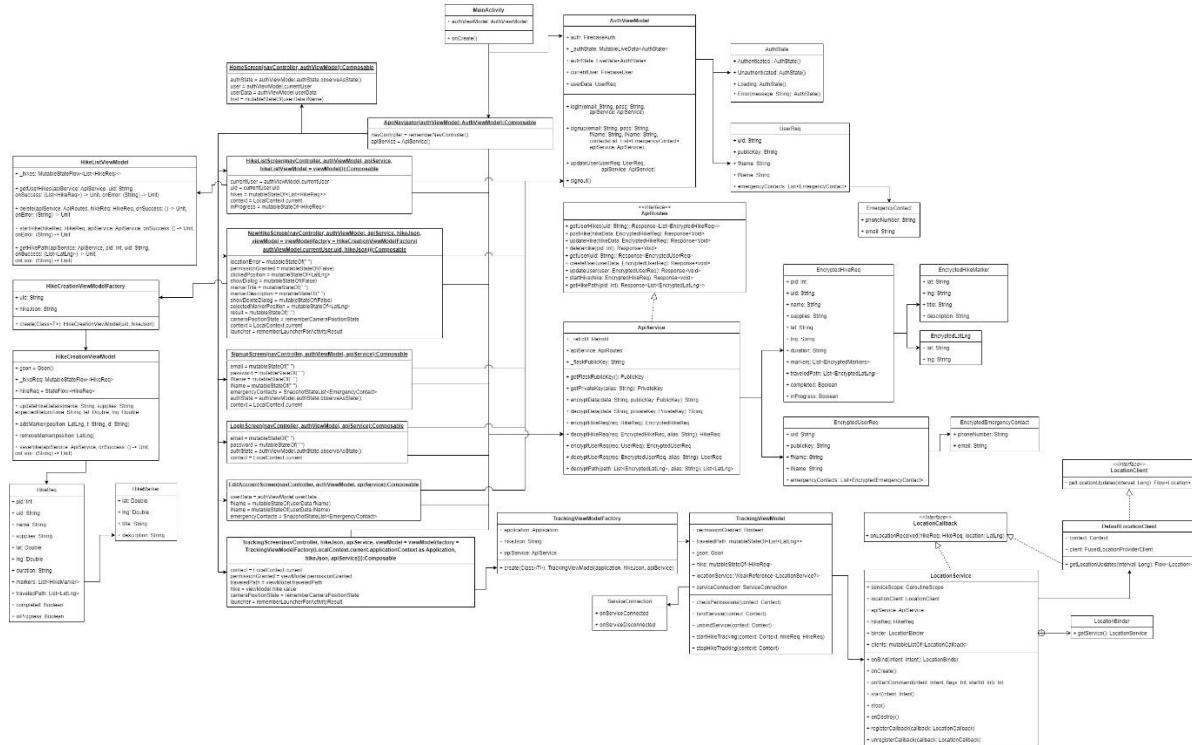


Class Diagram

A comprehensive class diagram was developed to model the core entities within the app, including:

- UserReq: Stores user information
- HikeReq: Stores hike information
- LocationService: Manages GPS tracking, retrieving real-time location data during the hike.
- DefaultLocationClient: Provides locations to LocationService.
- ApiService: Api implementation handling calls to flask and encryption/decryption of data
- ApiRoutes: Defines the routes to be implemented in the Api Service
- AuthViewModel: Handles authentication and user account details
- HikeCreationViewModel: Handles data relating to hike creation
- TrackingViewModel: Handles data relating to hike tracking and manages the location service
- ViewModelFactories: Required to instantiate view models that intake starting parameters while maintaining their access to the android lifecycle
- AppNavigator: Handles the navigation between each screen

Preview of class diagram:

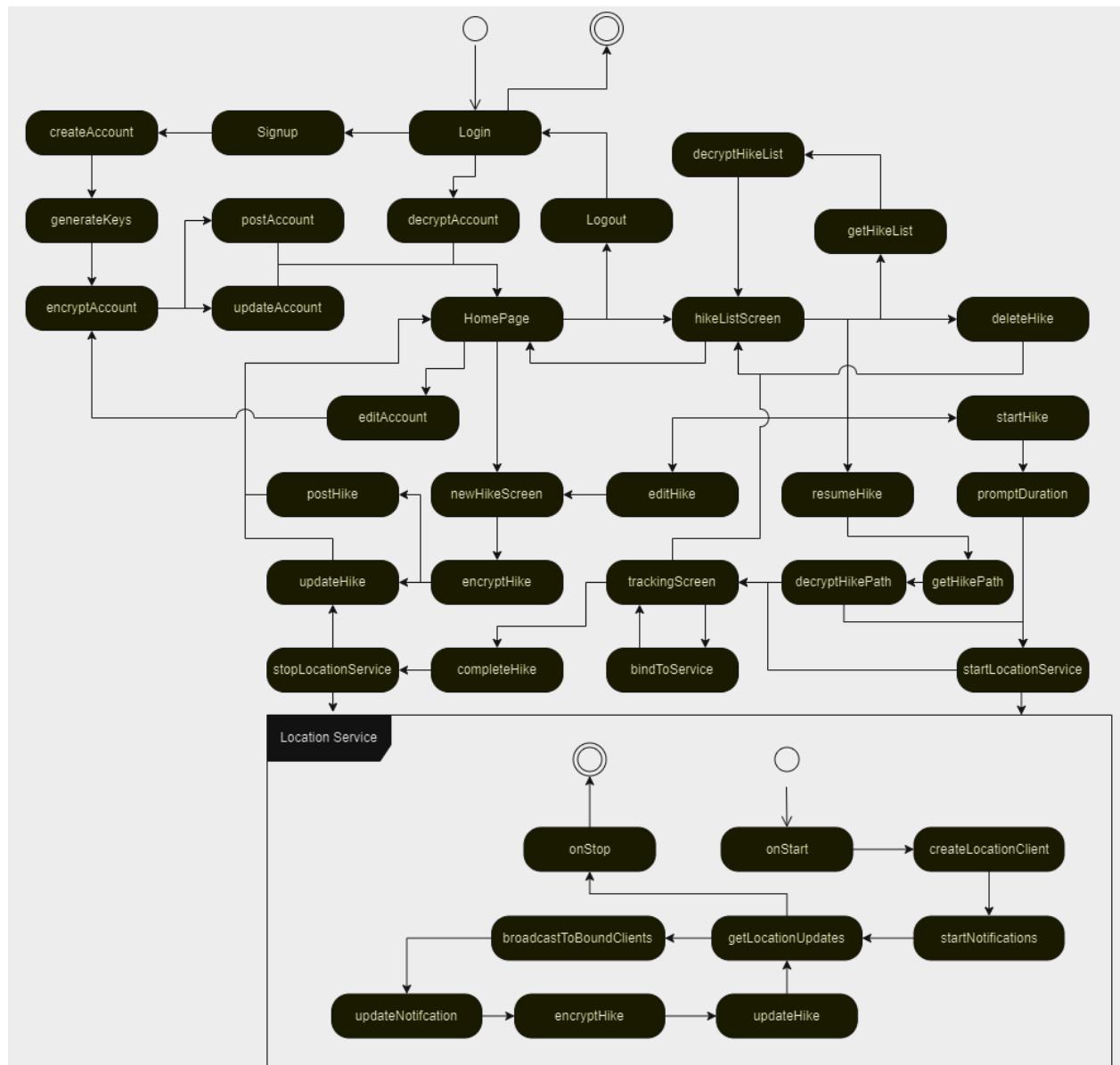


Link to class diagram:

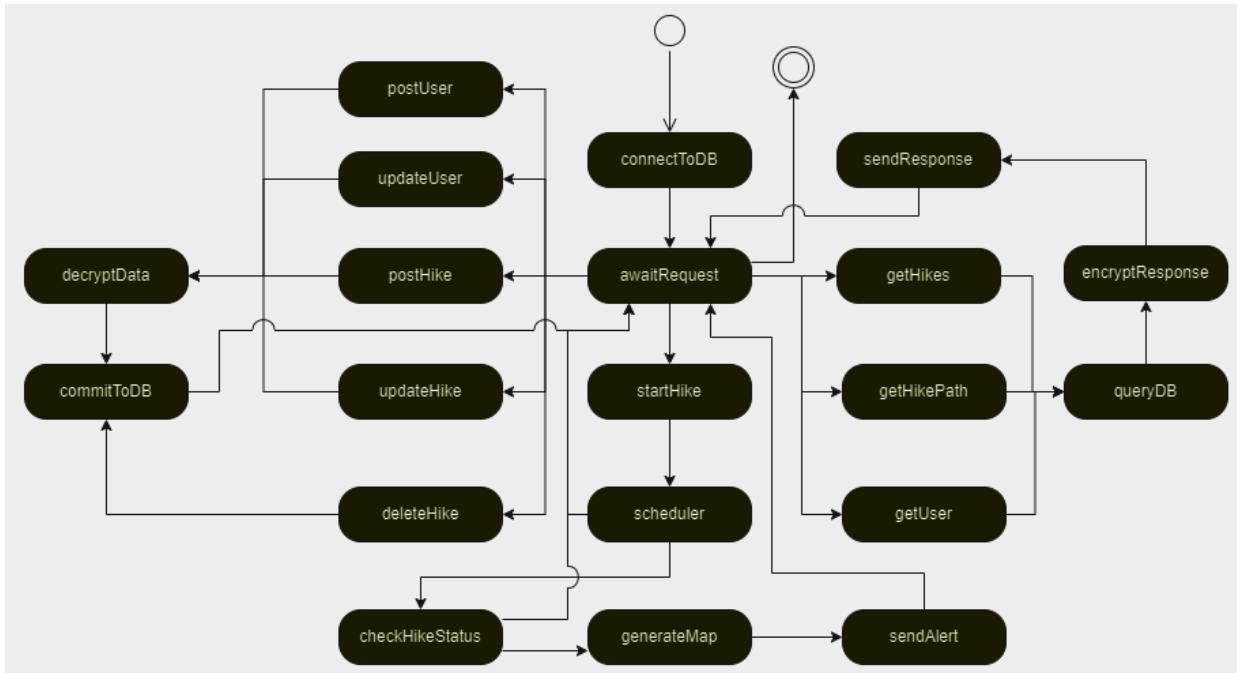
<https://drive.google.com/file/d/1ooVLnFC1e542N8iNHjiBgZK5nKEL08GP/view?usp=sharing>

State Diagrams

Android app state diagram:



Flask server state diagram:



Encryption

The Safer Hike app implements encryption to ensure secure communication between the Android application and the Flask server. The system is based on a public-private key pair mechanism, as well as SSL for database communication. Below is a detailed breakdown of how encryption is handled in the app.

Key Pair Generation and Storage (Client-Side)

When a user registers for the app, an RSA key pair is generated locally on their device. This keypair is stored in Android's KeyStore, ensuring the private key is secure and never leaves the device. A public key is also generated and stored locally. The public key is then sent to the Flask server during registration in string format, where it is stored in the PostgreSQL database associated with the user.

Here is the process for generating and storing the key pair:

```
val keyGen = KeyPairGenerator.getInstance(  
    KeyProperties.KEY_ALGORITHM_RSA,  
    "AndroidKeyStore"  
)  
val keyGenParameterSpec = KeyGenParameterSpec.Builder(  
    currentUser?.uid ?: "SaferHikeKey",  
    KeyProperties.PURPOSE_ENCRYPT or KeyProperties.PURPOSE_DECRYPT  
)  
.setDigests(KeyProperties.DIGEST_SHA256, KeyProperties.DIGEST_SHA512)  
.setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDING_RSA_PKCS1)  
.build()  
keyGen.initialize(keyGenParameterSpec)  
val keyPair = keyGen.genKeyPair()
```

Encrypting Data on the Client-Side

The app comes bundled with the Flask server's public key. This key is used by the app to encrypt data before sending it to the server, ensuring that only the server can decrypt the data using its private key.

The process of encrypting data using Flask's public key is as follows:

```
private fun getFlaskPublicKey(): PublicKey {
    val decodedKey = Base64.decode(_flaskPublicKey, Base64.DEFAULT)
    val keySpec = X509EncodedKeySpec(decodedKey)
    val keyFactory = KeyFactory.getInstance(algorithm: "RSA")
    return keyFactory.generatePublic(keySpec)
}

private fun encryptData(data: String, publicKey: PublicKey): String {
    val cipher = Cipher.getInstance(transformation: "RSA/ECB/PKCS1Padding")
    cipher.init(Cipher.ENCRYPT_MODE, publicKey)
    val encryptedBytes = cipher.doFinal(data.toByteArray())
    return Base64.encodeToString(encryptedBytes, Base64.DEFAULT)
}
```

Decrypting Data on the Client-Side

When the Flask server sends encrypted data back to the user the app decrypts the data using the private key stored securely in the Android KeyStore

```
private fun getPrivateKey(alias: String): PrivateKey {
    Log.d(tag: " ApiService", msg: "Attempting to getPrivateKey")
    val keyStore = KeyStore.getInstance(type: "AndroidKeyStore")
    keyStore.load(param: null)
    return keyStore.getKey(alias, password: null) as PrivateKey
}

private fun decryptData(encryptedData: String, privateKey: PrivateKey): String {
    Log.d(tag: " ApiService", msg: "Decrypting data: $encryptedData")
    val cipher = Cipher.getInstance(transformation: "RSA/ECB/PKCS1Padding")
    Log.d(tag: " ApiService", msg: "Initializing cipher")
    cipher.init(Cipher.DECRYPT_MODE, privateKey)
    Log.d(tag: " ApiService", msg: "Attempting to decode bytes and do final decryption")
    return try {
        val encryptedBytes = Base64.decode(encryptedData, Base64.DEFAULT)
        String(cipher.doFinal(encryptedBytes))
    } catch (e: Exception) {
        ""
    }
}
```

The private key is accessed using the user's UID as the alias. The data is decrypted using the private key, ensuring that only the intended client can read the data.

Encrypting Data on the Server

When the Flask server sends data to the user, it retrieves the user's public key from the database and uses it to encrypt the data. This ensures the data can only be decrypted by the intended recipient.

```
def get_user_public_key(uid):
    print("Attempting to retrieve user public key")
    user = db.session.query(User).filter_by(uid=uid).first()
    print("User Received")
    if user and user.public_key:
        print("Attempting to create public key")
        base64_key = user.public_key.strip()
        pem_key = (
            f"-----BEGIN PUBLIC KEY-----\n{base64_key}\n-----END PUBLIC KEY-----"
        ).encode("utf-8")
        public_key = serialization.load_pem_public_key(
            pem_key, default_backend()
        )
        print("Public key made")
        return public_key
    else:
        return None

17 usages new *
def encrypt_data(data, public_key):
    print("Attempting to encrypt data: {}".format(data))
    if not isinstance(data, bytes):
        data = data.encode('utf-8')
    encrypted_data = public_key.encrypt(
        data,
        padding.PKCS1v15()
    )
    print("Encrypted: {}".format(encrypted_data))
    return base64.b64encode(encrypted_data).decode('utf-8')
```

Decrypting Data on the Server

Similarly, the Flask server decrypts data sent by the user using the server's private key. This ensures that the data sent from the client remains confidential during transmission.

```
5 usages new *
def load_private_key():
    with open("flask_private_key.pem", "rb") as key_file:
        private_key = serialization.load_pem_private_key(
            key_file.read(),
            password=None,
            backend=default_backend()
        )
    return private_key

33 usages new *
def decrypt_data(encrypted_data, private_key):
    encrypted_data = base64.b64decode(encrypted_data)
    print("Inside decrypt_data")
    decrypted_data = None
    print("Encrypted data: {}".format(encrypted_data))
    try:
        decrypted_data = private_key.decrypt(
            encrypted_data,
            padding.PKCS1v15()
        )
        print("Decrypted data: {}".format(decrypted_data))
    except Exception as e:
        print("Decryption Failed:", str(e))
    print("Decrypted Data now returning")
    return decrypted_data.decode("utf-8")
```

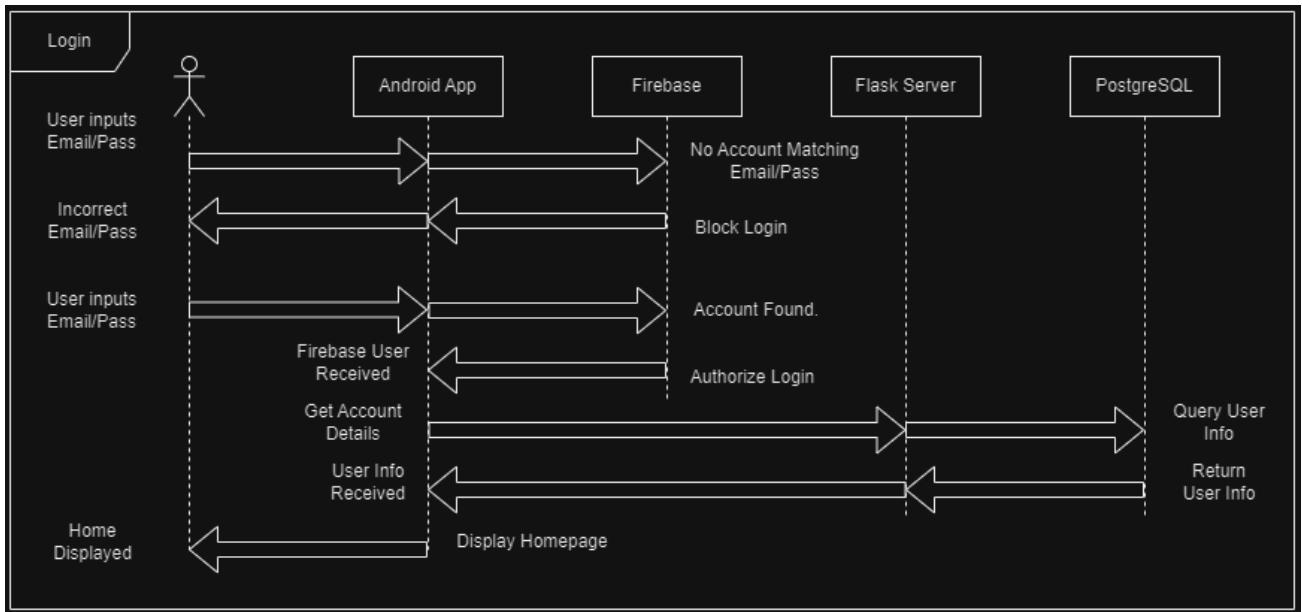
SSL for Database Communication

In addition to public-private key cryptography, communication between the Flask server and the PostgreSQL database is secured using SSL. This ensures that even during internal communication between the server and the database, all data remains encrypted, adding an extra layer of security to the system.

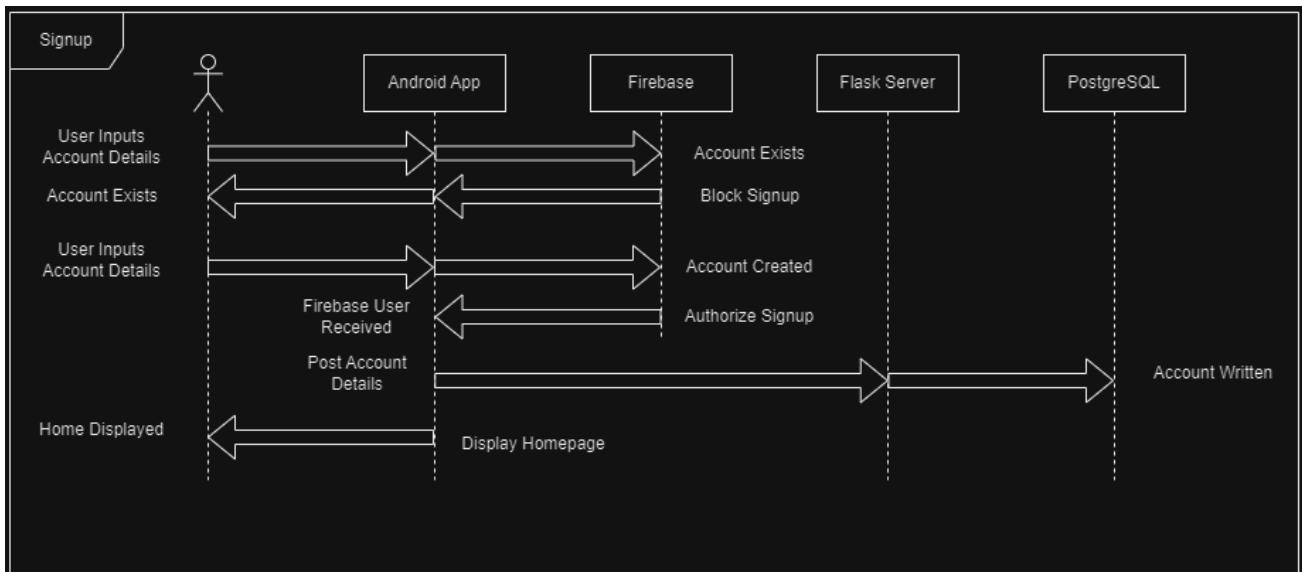
This combination ensures that all sensitive information is being encrypted during transmission and securely stored, minimizing the risk of unauthorized access or data breaches.

Sequence Diagrams

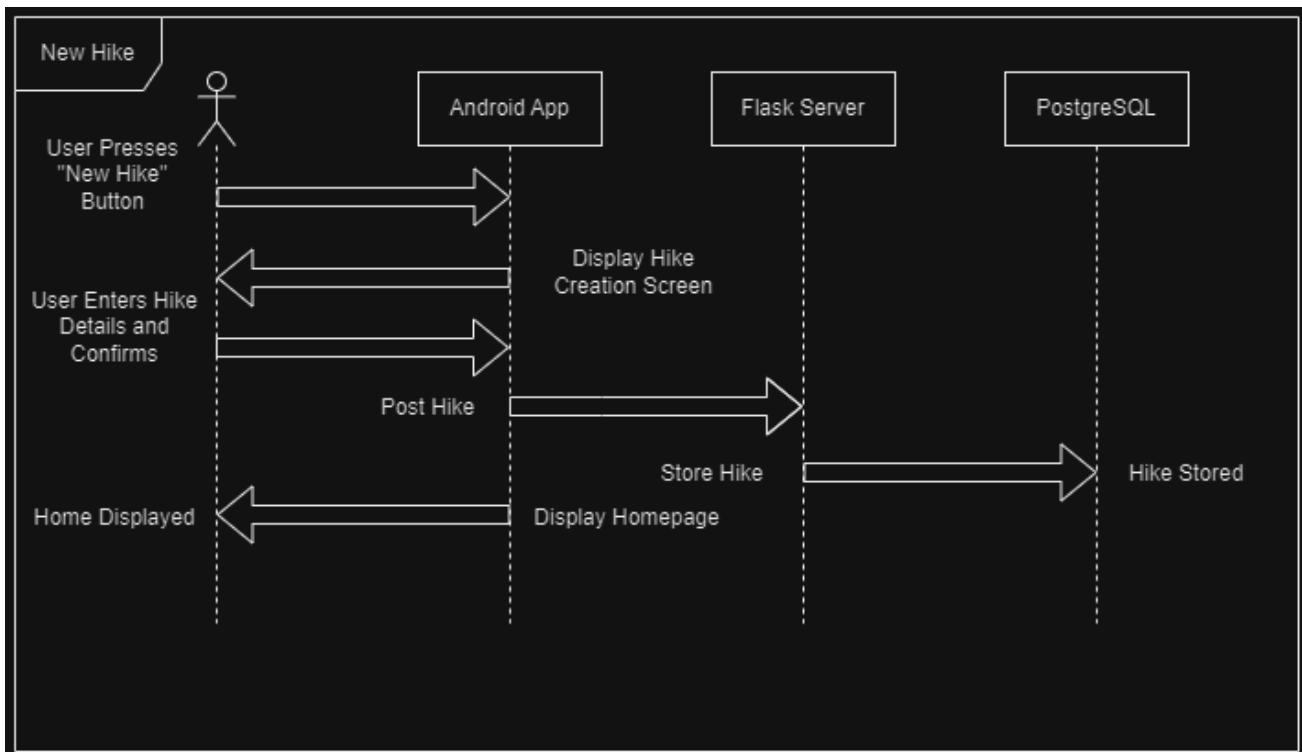
Login:



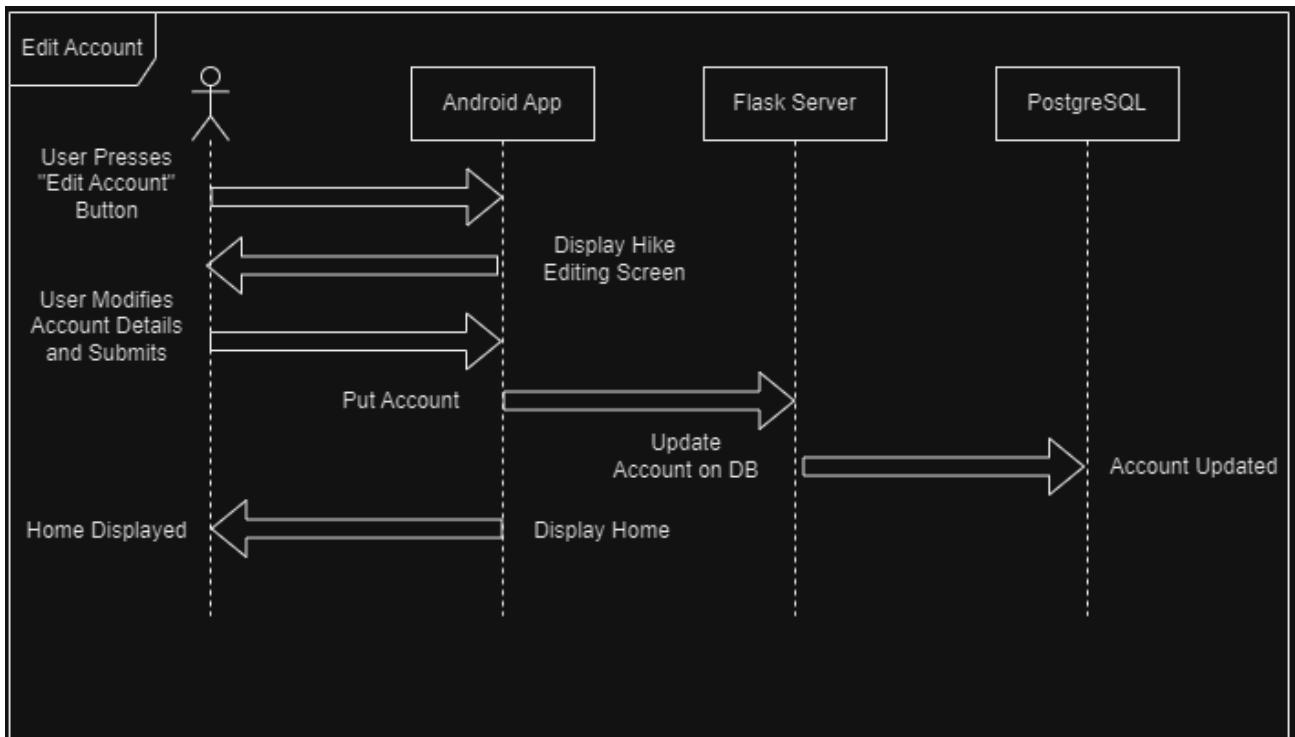
Signup:



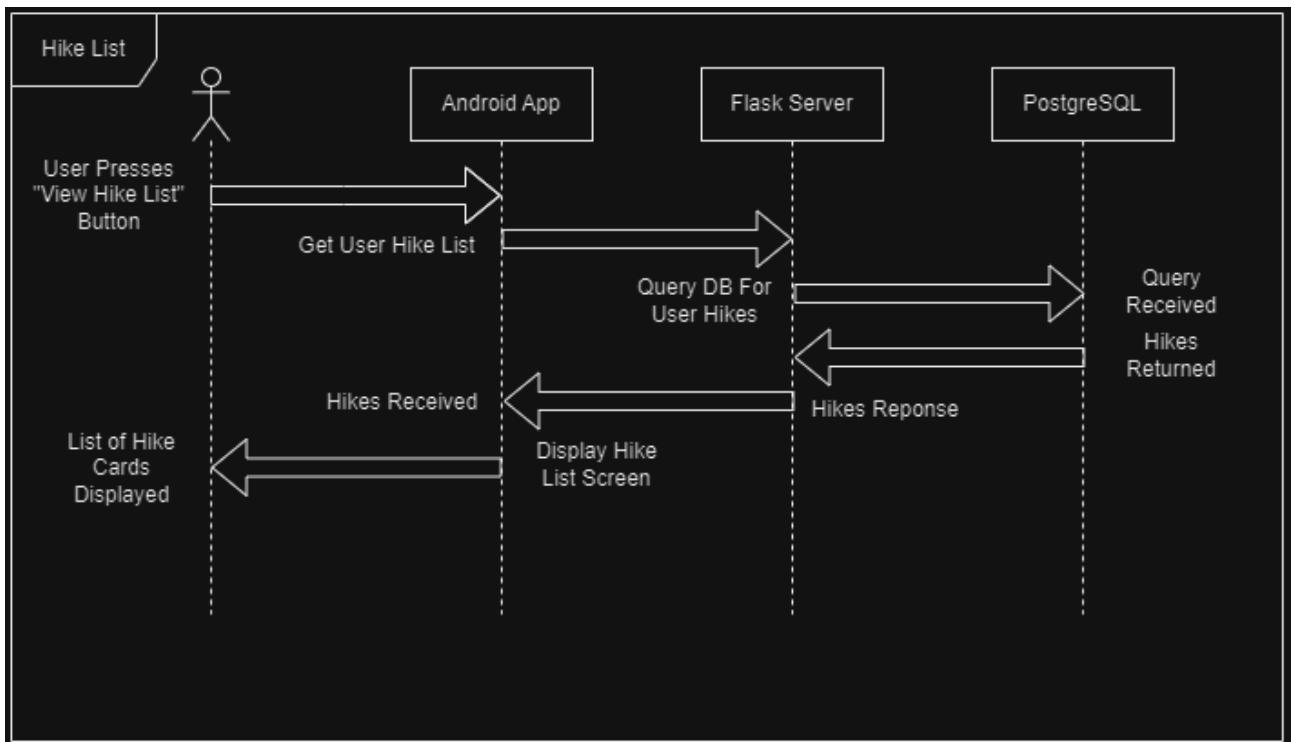
New Hike:



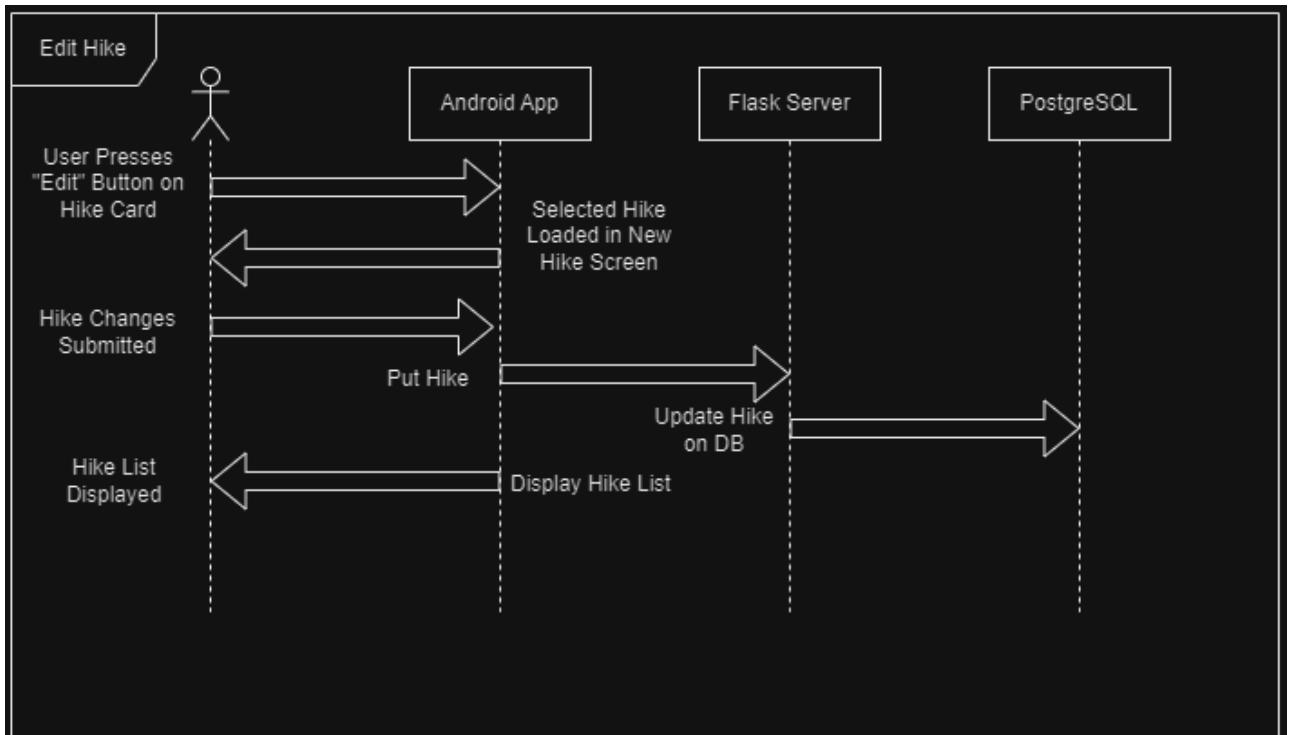
Edit Account:



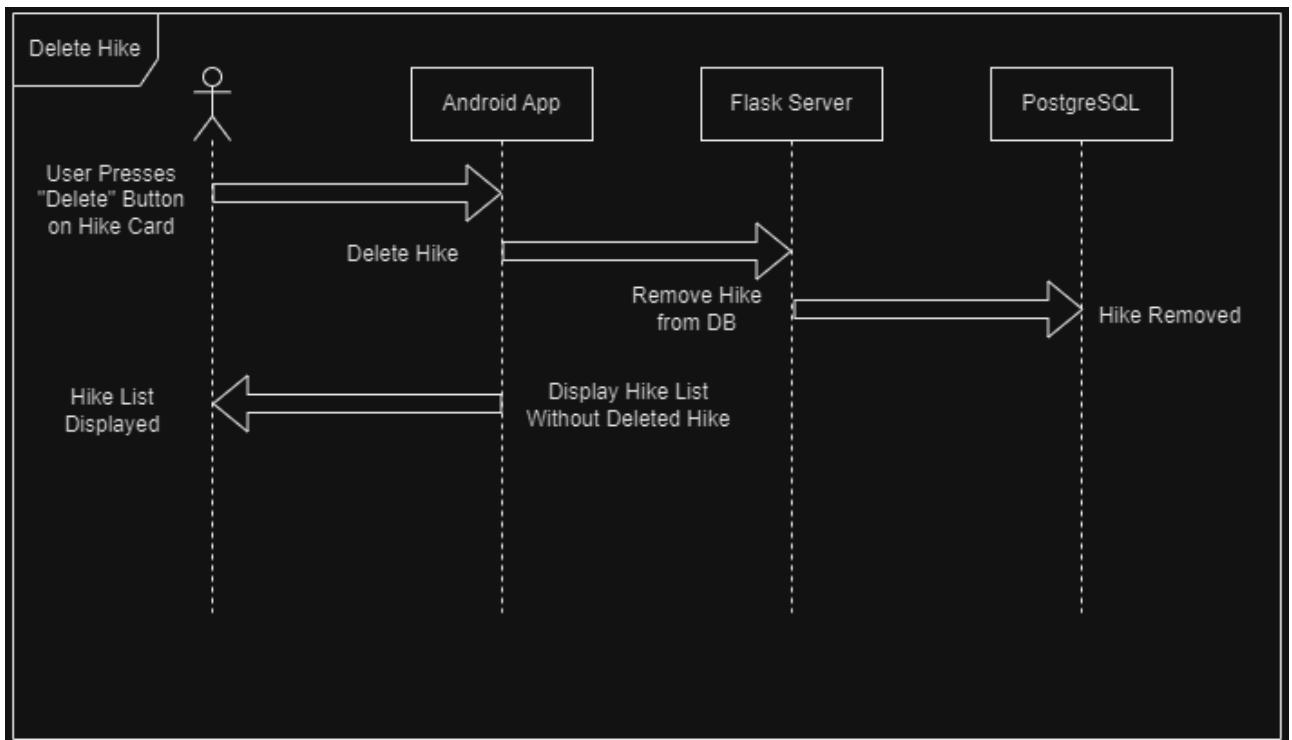
Hike List:



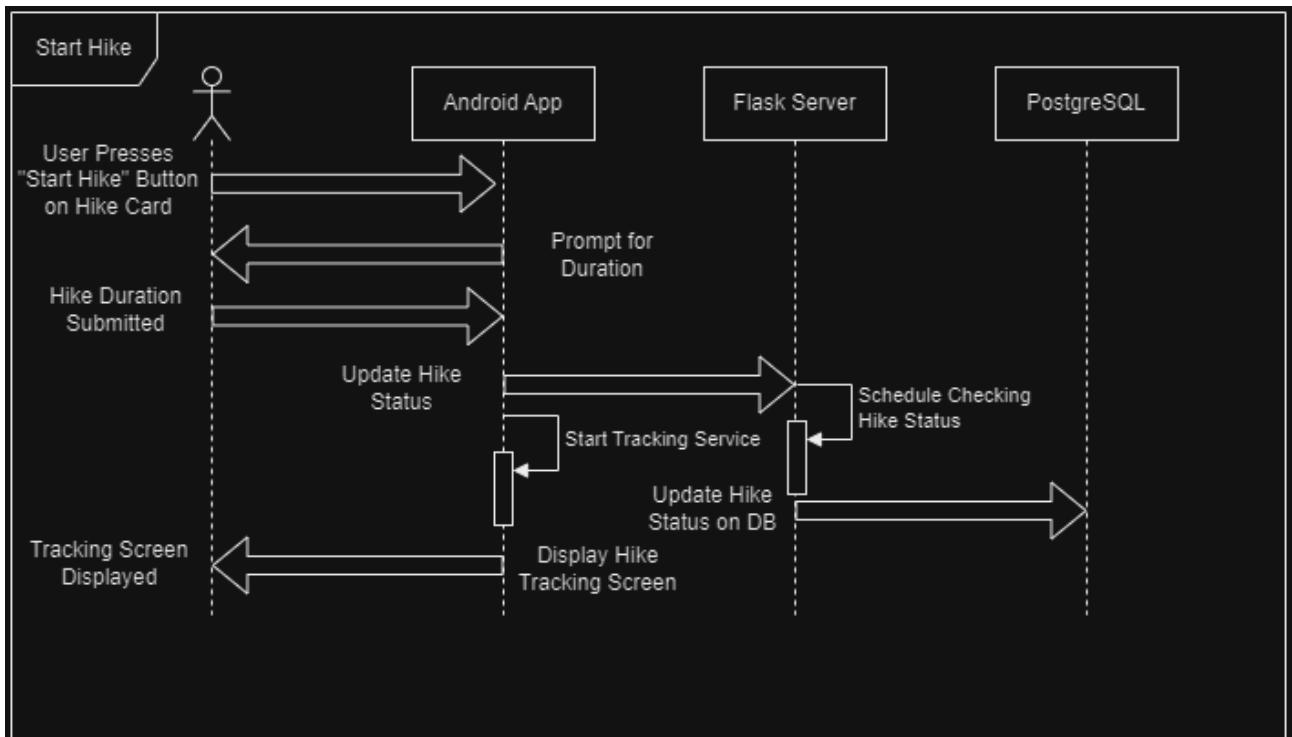
Edit Hike:



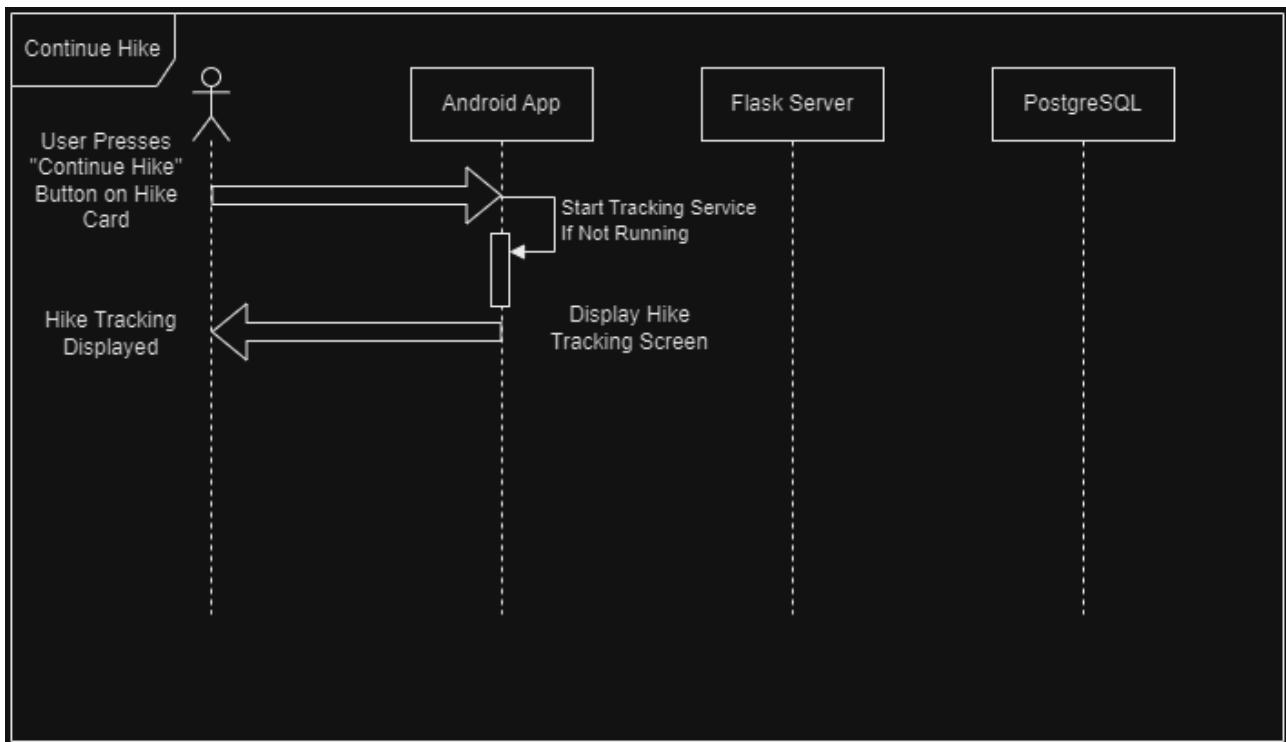
Delete Hike:



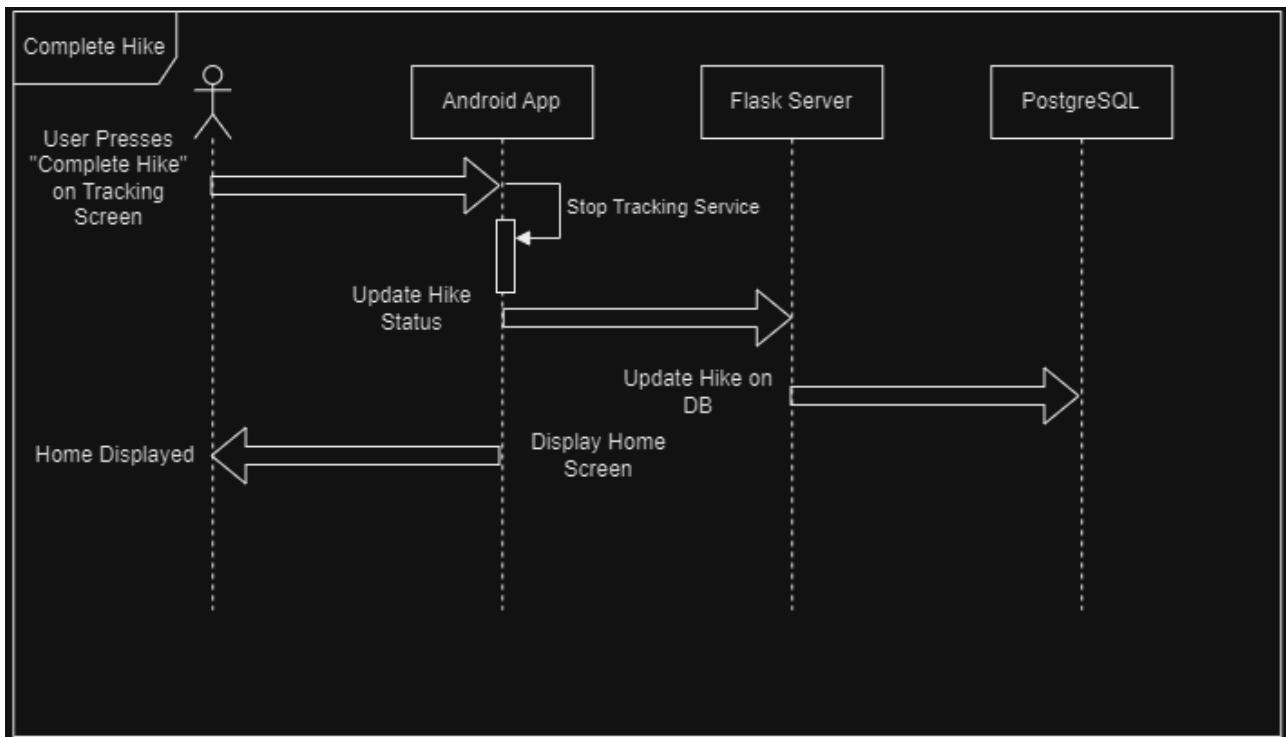
Start Hike:



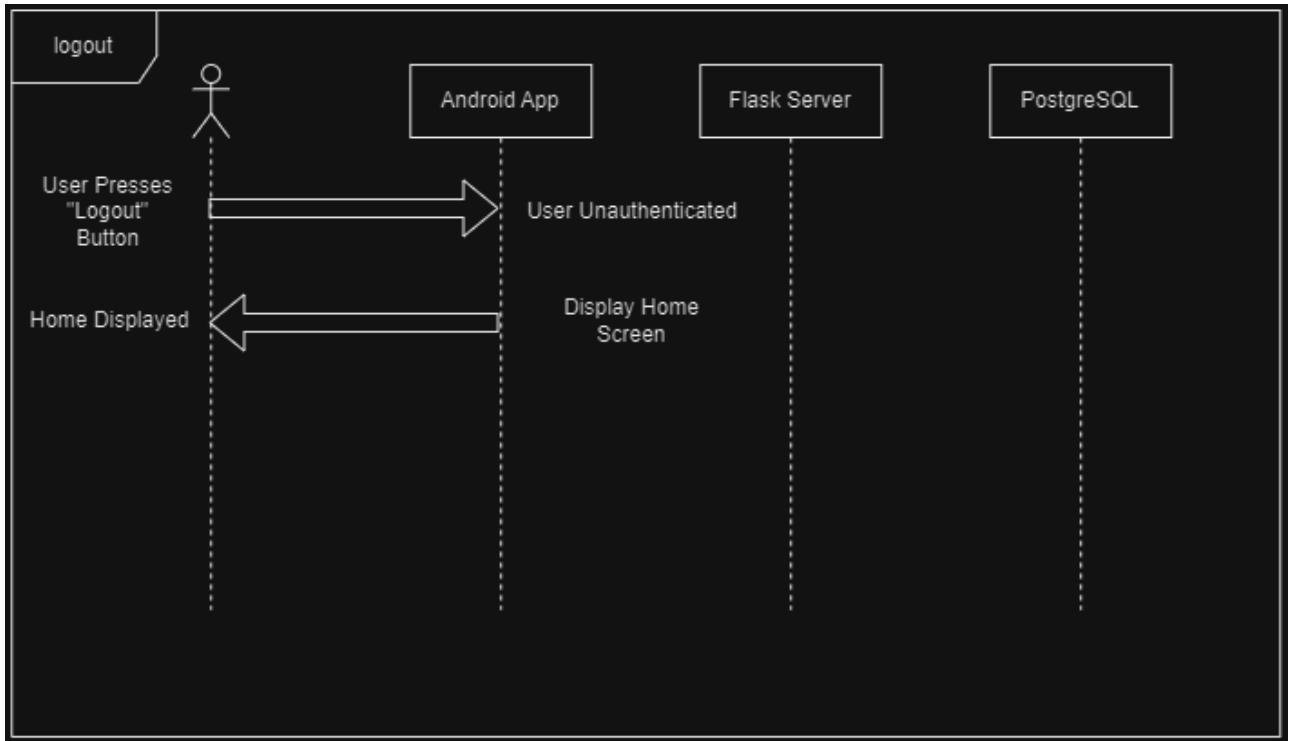
Continue Hike:



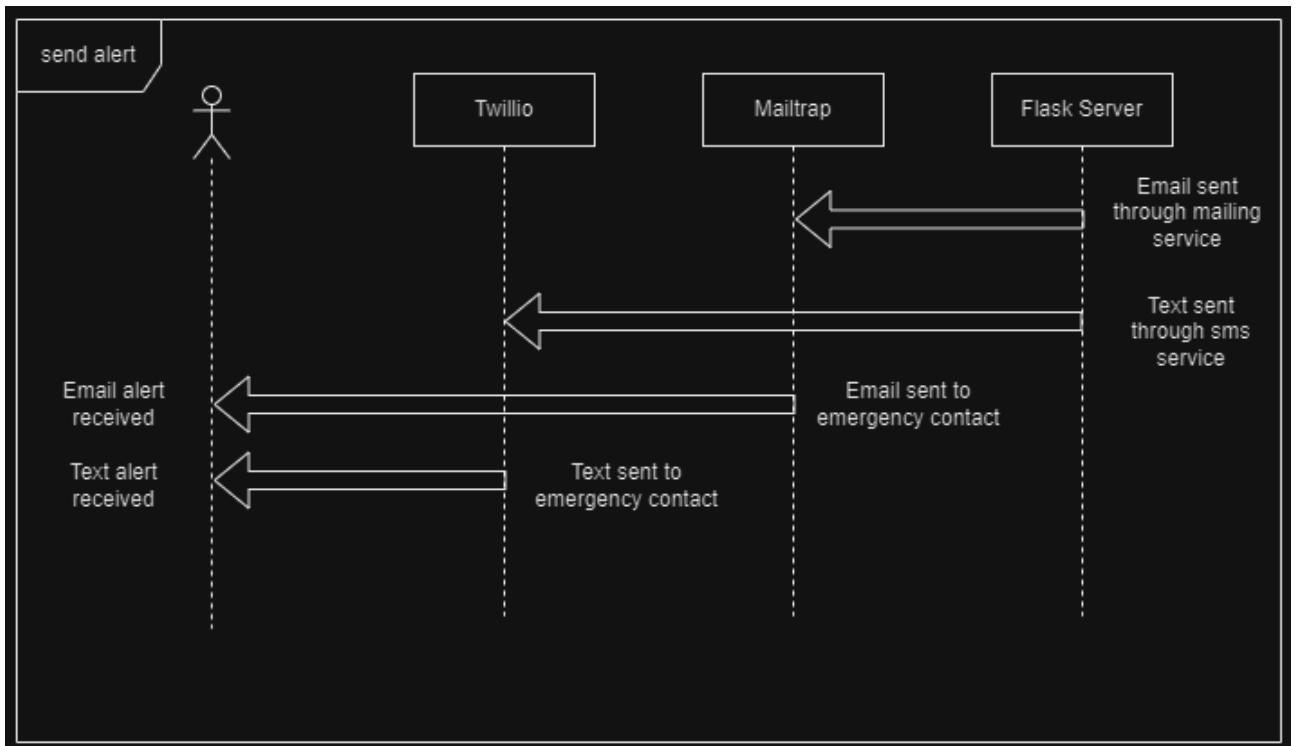
Complete Hike:



Logout:



Send Alert:



2.5.1 Api Documentation

User Model

Fields

- **uid**: (String) Unique identifier for the user.
- **public_key**: (String) User's public key for encryption.
- **f_name**: (String) User's first name.
- **l_name**: (String) User's last name.
- **emergency_contacts**: (JSON) List of user's emergency contacts.
- **registration_date**: (DateTime) Date of user registration.

Hike Model

Fields

- **pid**: (Integer) Unique identifier for the hike plan.
- **uid**: (String) Unique identifier for the user associated with the hike.
- **hname**: (String) Name of the hike.
- **end_time**: (String) Expected end time of the hike.
- **supplies**: (Text) Supplies needed for the hike.
- **markers**: (JSON) List of markers for points of interest along the hike.
- **traveled_path**: (JSON) List of locations traveled during the hike.
- **creation_date**: (DateTime) Date of hike creation.
- **lat**: (Float) Latitude of the hike's starting point.
- **lng**: (Float) Longitude of the hike's starting point.
- **completed**: (Boolean) Indicates if the hike has been completed.
- **in_progress**: (Boolean) Indicates if the hike is currently in progress.

Utility Functions

get_user_public_key(uid)

Retrieves the public key for a user identified by their unique UID.

- **Parameters:**

- uid: (String) Unique identifier for the user.

- **Returns:** (Public Key) User's public key if found; otherwise, None.

encrypt_data(data, public_key)

Encrypts the provided data using the specified public key.

- **Parameters:**

- data: (String or bytes) Data to be encrypted.
 - public_key: (Public Key) Public key to be used for encryption.

- **Returns:** (String) Base64-encoded encrypted data.

load_private_key()

Loads the private key from the file system.

- **Returns:** (Private Key) Loaded private key object.

decrypt_data(encrypted_data, private_key)

Decrypts the provided encrypted data using the specified private key.

- **Parameters:**

- encrypted_data: (String) Base64-encoded encrypted data to be decrypted.
 - private_key: (Private Key) Private key to be used for decryption.

- **Returns:** (String) Decrypted data as plain text.

generate_map_url(hike)

Generates a public URL for a map showing the hike's traveled path and markers.

- **Parameters:**

- hike: (Hike) Hike object containing traveled path and markers.

check_hike_status(hike_id)

Checks the status of a hike by its ID and sends notifications if the hike is incomplete.

- **Parameters:**

- hike_id: (Integer) Unique identifier for the hike.

- **Returns:** None

send_sms(to_number, message)

Sends an SMS notification to a specified phone number.

- **Parameters:**

- to_number: (String) Recipient's phone number.
 - message: (String) Message to be sent via SMS.

- **Returns:** None

send_email(to_email, subject, message)

Sends an email notification to a specified email address.

- **Parameters:**

- to_email: (String) Recipient's email address.
 - subject: (String) Subject line of the email.
 - message: (String) Message content of the email.

- **Returns:** None

Routes

1. Serve Map File

- **Endpoint:** /maps/<path:filename>
- **Method:** GET
- **Description:** Serves a map file from the specified directory.
- **Parameters:**
 - filename (path): The name of the file to be served.
- **Responses:**
 - 200 OK: Returns the requested map file.
 - 404 Not Found: If the specified file does not exist.

2. Start Hike

- **Endpoint:** /hikes/start
- **Method:** PUT
- **Description:** Starts a hike by updating its status and scheduling a status check.
- **Request Body (JSON):**
 - pid (integer): The ID of the hike.
 - duration (string): Expected return time in H:M format.
 - traveledPath (array): An array of traveled path coordinates, each with latitude and longitude.
- **Responses:**
 - 200 OK: Hike started successfully.
 - 400 Bad Request: If pid or duration is missing or invalid.
 - 404 Not Found: If the hike with the specified ID does not exist.

3. Get Hike Path

- **Endpoint:** /hikes/<int:hike_id>/paths
- **Method:** GET
- **Description:** Retrieves the traveled path for a specific hike.
- **Parameters:**
 - hike_id (integer): The ID of the hike.
- **Responses:**
 - 200 OK: Returns the encrypted traveled path of the hike.
 - 404 Not Found: If the hike with the specified ID does not exist.

4. Hike Request

- **Endpoint:** /hikes
- **Method:** GET, POST, PUT
- **Description:** Handles hike-related requests.
- **GET:** Retrieve all hikes for a user.
 - **Parameters:**
 - uid (string): User ID to filter hikes.
 - **Responses:**
 - 200 OK: Returns a list of hikes.
 - 404 Not Found: If no hikes are found for the user.

- **POST:** Create a new hike.
 - **Request Body (JSON):**
 - uid (string): User ID.
 - supplies (string): Supplies for the hike.
 - markers (array): Array of markers, each containing lat, lng, title, and description.
 - name (string): Name of the hike.
 - pid (integer): ID for the hike (0 for new).
 - **Responses:**
 - 201 Created: Hike created successfully.
 - 400 Bad Request: If required fields are missing or invalid.
- **PUT:** Update an existing hike.
 - **Request Body (JSON):**
 - pid (integer): ID of the hike to update.
 - Additional fields similar to the POST request.
 - **Responses:**
 - 200 OK: Hike updated successfully.
 - 404 Not Found: If no hike with the specified ID exists.

5. Delete Hike

- **Endpoint:** /hikes/<int:pid>
- **Method:** DELETE
- **Description:** Deletes a specific hike.
- **Parameters:**
 - pid (integer): ID of the hike to delete.
- **Responses:**
 - 200 OK: Hike deleted successfully.
 - 404 Not Found: If the hike with the specified ID does not exist.

6. Handle User

- **Endpoint:** /users
- **Method:** POST, PUT, GET
- **Description:** Manages user-related requests.
- **POST:** Create a new user.
 - **Request Body (JSON):**
 - uid (string): User ID.
 - publicKey (string): User's public key.
 - fName (string): First name.
 - lName (string): Last name.
 - emergencyContacts (array): List of emergency contacts, each with phoneNumber and email.
 - **Responses:**
 - 201 Created: User created successfully.
 - 500 Internal Server Error: If there is a problem with user creation.
- **PUT:** Update an existing user.
 - **Request Body (JSON):** Same structure as POST, but user updates.
 - **Responses:**
 - 200 OK: User updated successfully.
 - 404 Not Found: If the user does not exist.
- **GET:** Retrieve user details.
 - **Parameters:**
 - uid (string): User ID.
 - **Responses:**
 - 200 OK: Returns encrypted user data.
 - 404 Not Found: If the user with the specified ID does not exist.

Design Patterns

Throughout development of the Safer Hike app, several design patterns were used to ensure code modularity and maintainability:

MVVM (Model-View-ViewModel) Pattern

The MVVM pattern was implemented to separate concerns and allow better state management for the user interface in the app:

- Model: This layer handles the data-related logic of the app, including data retrieval from the Flask API
- View: The view layer, primarily built using Jetpack Compose, is responsible for rendering the UI based on the data provided by the ViewModel. Key views include the hike creation screen, tracking screen, and hike list
- ViewModel: The ViewModel is responsible for managing the app's state and business logic. It observes data changes from the Model and updates the View accordingly. For example, the TrackingViewModel receives location updates from LocationService and uses those to manage the overall hike state

The use of MVVM provided a clear separation of concerns, allowing for easier management of the UI state, especially during location tracking.

Singleton Pattern

The singleton pattern was applied to certain components like the LocationService and ApiService to ensure only one of these services runs during the app's lifecycle:

The LocationService manages GPS location updates and is active in the background even when the app is minimized.

The ApiService, implemented using Retrofit, handles all API communication between the app and the Flask server. Ensuring only one instance of ApiService reduced the risk of making duplicate API calls.

Observer Pattern

The observer pattern was used primarily in the ViewModel for handling live data updates. The tracking screen, for example, observes location changes via LiveData from the LocationService, updating the map in real time as the user moves. Observers in the ViewModel respond to location updates, triggering necessary state transitions and notifications. This pattern was crucial in handling asynchronous data flows, such as the locations provided by LocationService.

Factory Pattern

The factory pattern was used in the creation of HikeTrackingViewModels and HikeCreationViewModels because these ViewModels have input parameters. ViewModels in Android typically require a no-argument constructor, but in this case, both the HikeTrackingViewModels and HikeCreationViewModels needed input parameters, such as the hike ID or user information. The factory pattern allowed the app to encapsulate the creation logic of these ViewModels while ensuring that necessary parameters were passed correctly. This helped maintain separation of concerns and kept the code for the ViewModel instantiation clean and manageable. By using this pattern, the app was able to inject dependencies into ViewModels while adhering to Android's ViewModel architecture.

Deliverables

Deliverable	Status
User Application	Completed as Expected
Live Location Tracking	Completed as Expected
Flask Server	Completed as Expected
End-to-End Encryption	Completed as Expected
Automated Alert System	Completed as Expected
Final Report	Completed as Expected

2.6. Testing Details and Results

Testing was performed using a combination of unit tests for flask, and manual app testing using an android emulator capable of simulating location changes.

Flask Unit Testing

Unit testing for flask was conducted with pytest using encrypted user and hike data. Every route on the server has a unit test.

```
===== test session starts =====
collecting ... collected 10 items

testing.py::FlaskTestCase::test_create_user
testing.py::FlaskTestCase::test_create_hike
testing.py::FlaskTestCase::test_get_hikes_for_user
testing.py::FlaskTestCase::test_update_hike
testing.py::FlaskTestCase::test_start_hike
testing.py::FlaskTestCase::test_delete_hike
testing.py::FlaskTestCase::test_get_user
testing.py::FlaskTestCase::test_update_user
testing.py::FlaskTestCase::test_serve_map_file
testing.py::FlaskTestCase::test_serve_nonexistent_map_file

===== 10 passed, 2 warnings in 1.14s =====
```

Example test data:

```
def setUp(self):
    # Setup test data
    self.user_data = {
        "uid": "YfvAomj1p/aKWeo7k7IKfrcJ92KXiInw4l0Bmhz6hBz1WN60SNH+1B6M6evMQVvtgvdw/ok1R2STq8ZhJtecABVaiLPEs5rylEf3hQKvR0J+yMNku2ELN10TGYoF0gg0TzaVixNR+woR4Ud098UeUh2
        "publicKey": "MIBIEjRciRzQD0qkgh16w#0BAQFA0ACQAMII8CgkCQAC0NchS18vXk89sgowdw/ok1R2STq8ZhJtecABVaiLPEs5rylEf3hQKvR0J+yMNku2ELN10TGYoF0gg0TzaVixNR+woR4Ud098UeUh2
        "fName": "J0sJFLNu1qgChimh6k01+vrtzpzrg5di9xEnh/MnzbR6107PxPxFx2CLCqz+7LyaozP.Kadrn/[w]IcwtfwSxloU2Fvt3pzbicuZlkPgsWFepDzS0qu58v2cQ055yc9kev4f7J9/ow7ex10
        "lName": "FrIksece07zBn06TC/gjrBgvUYy4lPnzSTq1q0n8go8si+y/xjmcQUnJxki1zXvNob0hFaKij/*657hdjVWOp0iZBom+0KkBu1VhDMARX2vKp/gnP4ik0461BnVax9ThUBjJrVJKhCr2M
        "emergencyContacts": [
            {
                "phoneNumber": "+1U1znJQJWPSONSYKEJPSLW19JegoHQv/yrawxXuv5PK40b7KDjUiwt0CvwFrSER04xCdZjdh90xFxcA61bAj+j9BbRE7brEiv6mf1h39161VxZ6rQZ44Cw5jJuJ11Yvb2dAb
                "email": "aQvxSaS0i1vLvlNUDECMp8owEdh+aNc8VjijpYwMF1AVFJA1o4+kOen17n921yD15baeCTwE/dQRtgBhZE/41ObvJlmJ4N6T1fx+2efd0mD137x3kd2fYewka9NokkvC4zyRmonD
            }
        ]
    }

    self.hike_data = {
        "pid": 1,
        "uid": "Go0dececmHPutzgRYYDLoqJW/inI9zUwD0dvm+m89Ax4mWcJGv2c9fwzJZNz0zDBw9sH0NReBHG51HTuAYQQ7EzCwxCu4tWAK6K+KVARsDjTHHvglhXnFHfvLHNLYVqzzzemX/pvP7FP1cf0vhB3d
        "name": "Bcn0vRciRzQD0qkgh16w#0BAQFA0ACQAMII8CgkCQAC0NchS18vXk89sgowdw/ok1R2STq8ZhJtecABVaiLPEs5rylEf3hQKvR0J+yMNku2ELN10TGYoF0gg0TzaVixNR+woR4Ud098UeUh2
        "supplies": "X4eu2XzbqyJuisd0U0V8B#noCfGmDPsa13fV6gV0281yVduLqgnWcbbj3M7/gIm+*YH+BzHIsfGe1Um3Vf1zVtAg06Vne1J5ozcjoZ0wB1yTqvlfNWTFJ7d0x+P1IB0Ph/cb0oCjP8
        "lat": "HzR5Q0gQ+aZtALKKn68+bjh9LrySeemLdR3h02mT46SYT0ADJrycvQyjfs+V0c4nLleU/Jk66e160c420+EvgluLA0os3Tz0n82VrfToRUW1VkmK9803JQg4vH62KruadoC1cQ8NS/bglh/3q+aZM
        "lng": "W1ecmQBkPAcia3QhBaLW66t800t2zWdrqzT50517Mf0ty1uPaJwQuFuWC0+oxVlsdJtw9nJ97fForATmrw9cNqNyTofu1LESxv0d0K0tLNgcpc2EP1r4p1xLk3es1Jlymmiuao0tK2ag5b4f
        "duration": "hz250TRzg9ZNmW/VgWQLo0apuHE0jQTxW05E1xIwcb7k7LbtF0xH+QxOpYTTbyuABsho0avardzCPZj40daCzbdq+NIUlwfbtoXj0sk51iX5ZgANReUy0w3jg/vrlnR9/g
        "markers": [
            {
                "lat": "LogTgAfFnVlbQyyQyQy5ehIcx7+AeN/Qz10Ip5U0ln97exunAIxKrk6g8Y8l2aTHx6/MMx/C91vSPV47xfj0aUHJwEtjWcUhurAwY1lJMt7d0evL8K3JYr8TcTLwszfVhSbM0QVqQyY/eR9
                "lng": "izubK0yppWkK6Qk+xGIyB2Q0pVEnoS+SsPb+NCJnZu1IESDCu1+K0e/+kHEW2rnVNIXP0nfKg82ZFc1awVhn+9qd1896fa3F1bs37zI6awH0Ce0qt+*6sJ0pd3gYzLPr1t5zUxL40
                "title": "HhL+tJhK8cwojts7VdpS7zV+jcWHBFuXtbhP0v6RZ5H2Yz4wf0t5dj+jQEsKUBe3npCwya2RvePM7YiXdr5ME2cyFchc6Gcaotr8D0j1cmc05WnlmsMMSL+g/37mdhcrHycqMuL+TSWf
                "description": "id09yvvQqSh0raiFBs4q1leBgRyrAN108sfw#P1PjE9xfNjS3KLwEry1E00VBN1J6959nvAHAZL0JH61b4MCHXYO4vAYC2W1uokW52K1cyYLbqdbDtosMzorpngh5puv1pig
            },
            {
                "lat": "KjtC5SMUqyliwp0NL69KnR8+Pre/wWvTLiqaz4KL0B7RtjILcJB36uruBarcRJCYNeQNpPs8PKo0dbLt2nTzWdVc5uKKIMlMh76xthXo/YgS30d/TVdKqLnT1tdM1adpxoHgf+Xl7q/rk
                "lng": "Jg77ndg133ZBKuW0EUoRMrxR0aZWNf5iyMs1k1CtvB00DVsDHxoj5MswWt4Fx1NeYtf0lDisk0d6cmgqS0l/FVr/oyq+F+Kbm+dk/oFtACx9pC2fsqS00pxR/Gz1mt02nnVVK4frf
                "title": "g7VbSt89XIm0D0rZfLEni+CKTzDveBnUkpihv69D2de/tb0y1d1A+jH3SLH9H3ex9Afz2fB7Bh6p4FceJH660DSR24f4jsrCwEBs0uN0mvo9R5afv1/gu4RDrnxAOQPtuncC
                "description": "WLzJr1sb7XKEqr65t4au018Vst+B6xizvCjkqgqlbzF26SXzr7Ktp69ARQZIA8dvtfplphSw3Tw+Wrh51FHcwby0tf699+jc2sD6LJJSBYT29jgYg1B13eVXRxDj0x5LCM41
            }
        ],
        "traveledPath": [],
        "completed": False,
        "inProgress": False
    }
```

Manual Testing of App

1) User Login to Non-existent Account

Test	Attempt to login to non-existent account
Pass	Login blocked
Fail	Login Accepted

2) User Signup

Test	Attempt to sign up with new account
Pass	Account added to Firebase and PostgreSQL and app redirects home
Fail	Sign up blocked or account missing from Firebase or PostgreSQL

Post account & Establish connection between flask and PostgreSQL:

127.0.0.1	127.0.0.1	TCP	56 52947 → 5000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
127.0.0.1	127.0.0.1	TCP	56 5000 → 52947 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
127.0.0.1	127.0.0.1	TCP	44 52947 → 5000 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
127.0.0.1	127.0.0.1	TCP	1584 52947 → 5000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=1460 [TCP PDU reassembled in 6]
127.0.0.1	127.0.0.1	TCP	44 5000 → 52947 [ACK] Seq=1 Ack=1461 Win=2159872 Len=0
127.0.0.1	127.0.0.1	HTTP/1...	1045 POST /users HTTP/1.1 , JSON (application/json)
127.0.0.1	127.0.0.1	TCP	44 5000 → 52947 [ACK] Seq=1 Ack=2462 Win=2158848 Len=0
::1	::1	TCP	76 52948 → 5432 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
::1	::1	TCP	76 5432 → 52948 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
::1	::1	TCP	64 52948 → 5432 [ACK] Seq=1 Ack=1461 Win=2159872 Len=0
::1	::1	PGSQL	72 >?
::1	::1	TCP	64 5432 → 52948 [ACK] Seq=1 Ack=9 Win=2160640 Len=0
::1	::1	PGSQL	65 <
::1	::1	TCP	64 52948 → 5432 [ACK] Seq=9 Ack=2 Win=327168 Len=0
::1	::1	TLSv1.3	371 Client Hello (SNI=localhost)
::1	::1	TCP	64 5432 → 52948 [ACK] Seq=2 Ack=316 Win=2160128 Len=0
::1	::1	TLSv1.3	163 Hello Retry Request, Change Cipher Spec
::1	::1	TCP	64 52948 → 5432 [ACK] Seq=316 Ack=101 Win=327168 Len=0
::1	::1	TLSv1.3	410 Change Cipher Spec, Client Hello (SNI=localhost)
::1	::1	TCP	64 5432 → 52948 [ACK] Seq=101 Ack=662 Win=2159872 Len=0
::1	::1	TLSv1.3	2617 Server Hello, Application Data, Application Data, Application Data, Application Data, Application Data
::1	::1	TCP	64 52948 → 5432 [ACK] Seq=662 Ack=2654 Win=324608 Len=0
::1	::1	TLSv1.3	2233 Application Data, Application Data, Application Data
::1	::1	TCP	64 5432 → 52948 [ACK] Seq=2654 Ack=2831 Win=2157824 Len=0
::1	::1	TLSv1.3	128 Application Data
::1	::1	TCP	64 5432 → 52948 [ACK] Seq=2654 Ack=2895 Win=2157568 Len=0
::1	::1	TLSv1.3	129 Application Data

Encrypted fields in post request:

▼ Member: emergencyContacts
▼ Array
▼ Object
▼ Member: email
[Path with value [...]: /emergencyContacts/[]/email:KXkzeXAdKXAhN51bH8qtyd7e]
[Member with value [...]: email:KXkzeXAdKXAhN51bH8qtyd7esyVdZxc80ArUsQzeJFjc4E2oLphcpb2v]
String value [...]: KXkzeXAdKXAhN51bH8qtyd7esyVdZxc80ArUsQzeJFjc4E2oLphcpb2v
Key: email
[Path: /emergencyContacts/[]/email]
▼ Member: phoneNumber
[Path with value [...]: /emergencyContacts/[]/phoneNumber:IEcCTB4htWgeQxKn609RM9Px+uCGfsiddpSS/hZYdjXtrJ/sf0dxPNr]
[Member with value [...]: phoneNumber:IEcCTB4htWgeQxKn609RM9Px+uCGfsiddpSS/hZYdjXtrJ/sf0dxPNr]
String value [...]: IEcCTB4htWgeQxKn609RM9Px+uCGfsiddpSS/hZYdjXtrJ/sf0dxPNr
Key: phoneNumber
[Path: /emergencyContacts/[]/phoneNumber]
Key: emergencyContacts
[Path: /emergencyContacts]
▼ Member: fName
[Path with value [...]: /fName:Ys8QDdw1XvpHj1uDpLdQ00Mgt//iF9n4e7y0+vYKCydI5IU18DduQ]
[Member with value [...]: fName:Ys8QDdw1XvpHj1uDpLdQ00Mgt//iF9n4e7y0+vYKCydI5IU18DduQ]
String value [...]: Ys8QDdw1XvpHj1uDpLdQ00Mgt//iF9n4e7y0+vYKCydI5IU18DduQJfSVWSs4Tdg
Key: fName
[Path: /fName]

3) User Login

Test	Attempt to login to previously made account
Pass	Login accepted redirected home. Fetch encrypted account and decrypt successfully.
Fail	Login blocked

User login request:

```

127.0.0.1      127.0.0.1      HTTP      195 GET /users?uid=0KHOPd1UU2hE2ZAGQNFxBpHDmld2 HTTP/1.1
127.0.0.1      127.0.0.1      TCP       44 5000 → 53028 [ACK] Seq=1 Ack=152 Win=2161152 Len=0
::1             ::1           TLSv1.3   97 Application Data
::1             ::1           TCP       64 5432 → 52948 [ACK] Seq=4254 Ack=4514 Win=2156032 Len=0
::1             ::1           TLSv1.3   103 Application Data
::1             ::1           TCP       64 52948 → 5432 [ACK] Seq=4514 Ack=4293 Win=323072 Len=0
::1             ::1           TLSv1.3   397 Application Data
::1             ::1           TCP       64 5432 → 52948 [ACK] Seq=4293 Ack=4847 Win=2155776 Len=0
::1             ::1           TLSv1.3   888 Application Data
::1             ::1           TCP       64 52948 → 5432 [ACK] Seq=4847 Ack=5117 Win=322048 Len=0
::1             ::1           TLSv1.3   397 Application Data
::1             ::1           TCP       64 5432 → 52948 [ACK] Seq=5117 Ack=5180 Win=2155264 Len=0
::1             ::1           TLSv1.3   888 Application Data
::1             ::1           TCP       64 52948 → 5432 [ACK] Seq=5180 Ack=5941 Win=321280 Len=0
::1             ::1           TLSv1.3   100 Application Data
::1             ::1           TCP       64 5432 → 52948 [ACK] Seq=5941 Ack=5216 Win=2155264 Len=0
::1             ::1           TLSv1.3   106 Application Data
::1             ::1           TCP       64 52948 → 5432 [ACK] Seq=5216 Ack=5983 Win=321280 Len=0
127.0.0.1      127.0.0.1      TCP       211 5000 → 53028 [PSH, ACK] Seq=1 Ack=152 Win=2161152 Len=167
127.0.0.1      127.0.0.1      TCP       44 53028 → 5000 [ACK] Seq=152 Ack=168 Win=2161152 Len=0
127.0.0.1      127.0.0.1      HTTP/J... 2268 HTTP/1.1 200 OK , JSON (application/json)

```

Encrypted fields:

```

▼ Object
  ▼ Member: emergencyContacts
    ▼ Array
      ▼ Object
        ▼ Member: email
          [Path with value [...]: /emergencyContacts/[]/email:G7hgfluTiV03ABp7nbYSo/XnFw1VPUWFDU2d
          [Member with value [...]: email:G7hgfluTiV03ABp7nbYSo/XnFw1VPUWFDU2dQNm4SQqbhKkEsdeW1Vb
          String value [...]: G7hgfluTiV03ABp7nbYSo/XnFw1VPUWFDU2dQNm4SQqbhKkEsdeW1VbkxBvKnx9js0c
          Key: email
          [Path: /emergencyContacts/[]/email]
        ▼ Member: phoneNumber
          [Path with value [...]: /emergencyContacts/[]/phoneNumber:j/wM2Daisnu1JVa6G+tLtMB2Wy9xaj
          [Member with value [...]: phoneNumber:j/wM2Daisnu1JVa6G+tLtMB2Wy9xajGlu+LFrLLrINXTqfk/GwDRz3gdXgwrwdmOlwAu0
          String value [...]: j/wM2Daisnu1JVa6G+tLtMB2Wy9xajGlu+LFrLLrINXTqfk/GwDRz3gdXgwrwdmOlwAu0
          Key: phoneNumber
          [Path: /emergencyContacts/[]/phoneNumber]
        Key: emergencyContacts
        [Path: /emergencyContacts]
      ▼ Member: fName
        [Path with value [...]: /fName:lLzkLcsVuzHnURN4bbQ1cTTqvqbS00q7Ryp3IXhDb2Wj0Xpu7yj9e2pqQPHdKtSVo
        [Member with value [...]: fName:lLzkLcsVuzHnURN4bbQ1cTTqvqbS00q7Ryp3IXhDb2Wj0Xpu7yj9e2pqQPHdKtSVo
        String value [...]: lLzkLcsVuzHnURN4bbQ1cTTqvqbS00q7Ryp3IXhDb2Wj0Xpu7yj9e2pqQPHdKtSVo
        Key: fName
        [Path: /fName]
      ▼ Member: lName
        [Path with value [...]: /lName:dLlKaZO/egN4L3ut46xQaktGVPeTvh+BWCBlY5TNIVqsSPFdS/m+LXu+eYyy15hPh
        [Member with value [...]: lName:dLlKaZO/egN4L3ut46xQaktGVPeTvh+BWCBlY5TNIVqsSPFdS/m+LXu+eYyy15hPh
        String value [...]: dLlKaZO/egN4L3ut46xQaktGVPeTvh+BWCBlY5TNIVqsSPFdS/m+LXu+eYyy15hPh/F14mIrZd6L2
        Key: lName
        [Path: /lName]
      ▼ Member: publicKey
        [Path with value [...]: /publicKey:MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEAndxxFGnm8n+81TKeq
        [Member with value [...]: publicKey:MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEAndxxFGnm8n+81TKeq
        String value [...]: MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEAndxxFGnm8n+81TKeqM8QvyoAKf8Jh0r7Y
        Key: publicKey
        [Path: /publicKey]
      ▼ Member: uid
        [Path with value [...]: /uid:gMx1HHDOJZnpVFN3gBCvdDy/5X8Em0BrpbE0xdKUNKPyiNHCHNLD4gqyRhIcA0ccCS9
        [Member with value [...]: uid:gMx1HHDOJZnpVFN3gBCvdDy/5X8Em0BrpbE0xdKUNKPyiNHCHNLD4gqyRhIcA0ccCS9
        String value [...]: gMx1HHDOJZnpVFN3gBCvdDy/5X8Em0BrpbE0xdKUNKPyiNHCHNLD4gqyRhIcA0ccCS9ZGV3LYLdfv
        Key: uid
        [Path: /uid]

```

4) View Hike List Without Saved Hikes

Test	View hike list on fresh user account
Pass	Popup message displayed "No Hikes Found"
Fail	Unhandled exception or app crash

Get hike list request:

```

HTTP    195 GET /hikes?uid=0KHOPd1UU2hE2ZAGQNFxBpHDmlD2 HTTP/1.1
TCP      44 5000 → 53058 [ACK] Seq=1 Ack=152 Win=2161152 Len=0
TLSv1.3  97 Application Data
TCP      64 5432 → 52948 [ACK] Seq=5983 Ack=5249 Win=2155264 Len=0
TLSv1.3  103 Application Data
TCP      64 52948 → 5432 [ACK] Seq=5249 Ack=6022 Win=321280 Len=0
TLSv1.3  677 Application Data
TCP      64 5432 → 52948 [ACK] Seq=6022 Ack=5862 Win=2154752 Len=0
TLSv1.3  559 Application Data
TCP      64 52948 → 5432 [ACK] Seq=5862 Ack=6517 Win=320768 Len=0
TLSv1.3  100 Application Data
TCP      64 5432 → 52948 [ACK] Seq=6517 Ack=5898 Win=2154752 Len=0
TLSv1.3  106 Application Data
TCP      64 52948 → 5432 [ACK] Seq=5898 Ack=6559 Win=320768 Len=0
TCP      216 5000 → 53058 [PSH, ACK] Seq=1 Ack=152 Win=2161152 Len=0
TCP      44 53058 → 5000 [ACK] Seq=152 Ack=173 Win=2161152 Len=0
HTTP/J...  85 HTTP/1.1 404 NOT FOUND , JSON (application/json)

```

5) Create New Hike

Test	Input hike details and submit
Pass	Hike encrypted and transmitted to flask which decrypts and stores hike on PostgreSQL
Fail	Hike fails to save, or encryption/decryption fails

Post hike request:

```

127.0.0.1      127.0.0.1      HTTP/J...  1049 POST /hikes HTTP/1.1 , JSON (application/json)
127.0.0.1      127.0.0.1      TCP      44 5000 → 53127 [ACK] Seq=1 Ack=6846 Win=2154240
::1            ::1           TLSv1.3  97 Application Data
::1            ::1           TCP      64 5432 → 52948 [ACK] Seq=6559 Ack=5931 Win=21544
::1            ::1           TLSv1.3  103 Application Data
::1            ::1           TCP      64 52948 → 5432 [ACK] Seq=5931 Ack=6598 Win=32076
::1            ::1           TLSv1.3  708 Application Data
::1            ::1           TCP      64 5432 → 52948 [ACK] Seq=6598 Ack=6575 Win=21539
::1            ::1           TLSv1.3  212 Application Data
::1            ::1           TCP      64 52948 → 5432 [ACK] Seq=6575 Ack=6746 Win=32051
::1            ::1           TLSv1.3  98 Application Data
::1            ::1           TCP      64 5432 → 52948 [ACK] Seq=6746 Ack=6609 Win=21539
::1            ::1           TLSv1.3  104 Application Data
::1            ::1           TCP      64 52948 → 5432 [ACK] Seq=6609 Ack=6786 Win=32051
127.0.0.1      127.0.0.1      TCP      214 5000 → 53127 [PSH, ACK] Seq=1 Ack=6846 Win=215
127.0.0.1      127.0.0.1      TCP      44 53127 → 5000 [ACK] Seq=6846 Ack=171 Win=216115
127.0.0.1      127.0.0.1      HTTP/J...  87 HTTP/1.1 201 CREATED , JSON (application/json)

```

Encrypted hike fields:

```
▼ Object
  ▼ Member: completed
    [Path with value: /completed:false]
    [Member with value: completed:false]
    False value
    Key: completed
    [Path: /completed]
  ▼ Member: duration
    [Path with value [...]: /duration:NAWNJwED2EqhglQFX0Pk+BZe8lrvnJCG/qADvaObR2fAn8rFJ]
    [Member with value [...]: duration:NAWNJwED2EqhglQFX0Pk+BZe8lrvnJCG/qADvaObR2fAn8rF]
    String value [...]: NAWNJwED2EqhglQFX0Pk+BZe8lrvnJCG/qADvaObR2fAn8rFJ/MMio4lZ8ZcSyX
    Key: duration
    [Path: /duration]
  ▼ Member: inProgress
    [Path with value: /inProgress:false]
    [Member with value: inProgress:false]
    False value
    Key: inProgress
    [Path: /inProgress]
  ▼ Member: lat
    [Path with value [...]: /lat:hFicr8N2KZGh0bSYfZjeyMDGJ/OYcvWh3YmKeKijz8u75r200eU2RZ]
    [Member with value [...]: lat:hFicr8N2KZGh0bSYfZjeyMDGJ/OYcvWh3YmKeKijz8u75r200eU2RZ]
    String value [...]: hFicr8N2KZGh0bSYfZjeyMDGJ/OYcvWh3YmKeKijz8u75r200eU2RZkk+yar44H
    Key: lat
    [Path: /lat]
  ▼ Member: lng
    [Path with value [...]: /lng:00q9mAFCx1YQ1HV3P0T5ItSwQzyN/6ADX7fe3USxRi/K01TFY8mWHiw]
    [Member with value [...]: lng:00q9mAFCx1YQ1HV3P0T5ItSwQzyN/6ADX7fe3USxRi/K01TFY8mWHiw]
    String value [...]: 00q9mAFCx1YQ1HV3P0T5ItSwQzyN/6ADX7fe3USxRi/K01TFY8mWHiwssmTAUI1P
    Key: lng
    [Path: /lng]
  ▼ Member: markers
    ▼ Array
      ▼ Object
        ▼ Member: description
          [Path with value [...]: /markers/[]/description:L64z1PpLNzJaX6yo05jp/HBpR8A]
          [Member with value [...]: description:L64z1PpLNzJaX6yo05jp/HBpR8AmQ+mhYAUCvGSn906Ngn1LZg7rk/K]
          String value [...]: L64z1PpLNzJaX6yo05jp/HBpR8AmQ+mhYAUCvGSn906Ngn1LZg7rk/K
          Key: description
          [Path: /markers/[]/description]
        ▼ Member: lat
          [Path with value [...]: /markers/[]/lat:1hhW2t+618vhAFGcxhmGUBLrKHAtdM4Bu6/]
          [Member with value [...]: lat:1hhW2t+618vhAFGcxhmGUBLrKHAtdM4Bu6/1uS6VNodXe]
          String value [...]: 1hhW2t+618vhAFGcxhmGUBLrKHAtdM4Bu6/1uS6VNodXeIAJ/UkkGN1
          Key: lat
          [Path: /markers/[]/lat]
        ▼ Member: lng
          [Path with value [...]: /markers/[]/lng:MMde2q4ZoEf5Q9rfuoXh8qarNz2nHh0xwSe]
          [Member with value [...]: lng:MMde2q4ZoEf5Q9rfuoXh8qarNz2nHh0xwSebU5hu9wJ8p]
          String value [...]: MMde2q4ZoEf5Q9rfuoXh8qarNz2nHh0xwSebU5hu9wJ8pGBoZ06PUFU
          Key: lng
          [Path: /markers/[]/lng]
```

6) View List of Hikes

Test	View hike list after adding hikes
Pass	Hike list decrypted and shown with hikes as expandable cards
Fail	Fails to retrieve hike list

Successful get hike list request:

```
HTTP    195 GET /hikes?uid=0KHO...Dm1D2 HTTP/1.1
TCP      44 5000 → 53208 [ACK] Seq=1 Ack=152 Win=2161152 Len=0
TLSv1.3  97 Application Data
TCP      64 5432 → 52948 [ACK] Seq=7240 Ack=7547 Win=2152960 Len=0
TLSv1.3  103 Application Data
TCP      64 52948 → 5432 [ACK] Seq=7547 Ack=7279 Win=320000 Len=0
TLSv1.3  677 Application Data
TCP      64 5432 → 52948 [ACK] Seq=7279 Ack=8160 Win=2152448 Len=0
TLSv1.3  1392 Application Data
TCP      64 52948 → 5432 [ACK] Seq=8160 Ack=8607 Win=318720 Len=0
TLSv1.3  397 Application Data
TCP      64 5432 → 52948 [ACK] Seq=8607 Ack=8493 Win=2151936 Len=0
TLSv1.3  888 Application Data
TCP      64 52948 → 5432 [ACK] Seq=8493 Ack=9431 Win=317696 Len=0
TLSv1.3  100 Application Data
TCP      64 5432 → 52948 [ACK] Seq=9431 Ack=8529 Win=2151936 Len=0
TLSv1.3  106 Application Data
TCP      64 52948 → 5432 [ACK] Seq=8529 Ack=9473 Win=317696 Len=0
TCP      212 5000 → 53208 [PSH, ACK] Seq=1 Ack=152 Win=2161152 Len=168 [TCP PDU reassembled in 283]
TCP      44 53208 → 5000 [ACK] Seq=152 Ack=169 Win=2161152 Len=0
HTTP/J... 11004 HTTP/1.1 200 OK , JSON (application/json)
```

Encrypted hike list fields:

```
▼ Array
  ▼ Object
    ▼ Member: completed
      [Path with value: /[]/completed:false]
      [Member with value: completed:false]
      False value
      Key: completed
      [Path: /[]/completed]
    ▼ Member: duration
      [Path with value: /[]/duration:]
      [Member with value: duration:]
      String value:
      Key: duration
      [Path: /[]/duration]
    ▼ Member: inProgress
      [Path with value: /[]/inProgress:false]
      [Member with value: inProgress:false]
      False value
      Key: inProgress
      [Path: /[]/inProgress]
    ▼ Member: lat
      [Path with value [...]: /[]/lat:N6/4OnTVCfpdYaIhwGBkrY3aZzDVZczIb+nnUiQxwYqojxbB3ycH4G+T04dqhC1w0ZB+kTGENNL]
      [Member with value [...]: lat:N6/4OnTVCfpdYaIhwGBkrY3aZzDVZczIb+nnUiQxwYqojxbB3ycH4G+T04dqhC1w0ZB+kTGENNL]
      String value [...]: N6/4OnTVCfpdYaIhwGBkrY3aZzDVZczIb+nnUiQxwYqojxbB3ycH4G+T04dqhC1w0ZB+kTGENNL
      Key: lat
      [Path: /[]/lat]
    ▼ Member: lng
      [Path with value [...]: /[]/lng:mVU4UsSmEWTPZiw/tI4vvm7CAS5XqeZ0PlbbOab3UWhaYdTo1FGRppU0Ag9iABeUddFlads29ef]
      [Member with value [...]: lng:mVU4UsSmEWTPZiw/tI4vvm7CAS5XqeZ0PlbbOab3UWhaYdTo1FGRppU0Ag9iABeUddFlads29ef]
      String value [...]: mVU4UsSmEWTPZiw/tI4vvm7CAS5XqeZ0PlbbOab3UWhaYdTo1FGRppU0Ag9iABeUddFlads29ef
      Key: lng
      [Path: /[]/lng]
    ▼ Member: markers
      ▼ Array
        ▼ Object
          ▼ Member: description
            [Path with value [...]: /[]/markers/[]/description:Ic70nPaZgSGV3dCxRztgFyAyBI0S8v5bjv8T4u6y4/RF9B6pS]
            [Member with value [...]: description:Ic70nPaZgSGV3dCxRztgFyAyBI0S8v5bjv8T4u6y4/RF9B6pS]
            String value [...]: Ic70nPaZgSGV3dCxRztgFyAyBI0S8v5bjv8T4u6y4/RF9B6pSIVJqrn4vtDtJVmxwvf
            Key: description
            [Path: /[]/markers/[]/description]
          ▼ Member: lat
            [Path with value [...]: /[]/markers/[]/lat:Kizd6UXi+ETy5TaigvzH640XQ42FjC0HHfMvDRn38MEWp1qdwJH1m97C]
            [Member with value [...]: lat:Kizd6UXi+ETy5TaigvzH640XQ42FjC0HHfMvDRn38MEWp1qdwJH1m97C]
            String value [...]: Kizd6UXi+ETy5TaigvzH640XQ42FjC0HHfMvDRn38MEWp1qdwJH1m97C
            Key: lat
            [Path: /[]/markers/[]/lat]
          ▼ Member: lng
            [Path with value [...]: /[]/markers/[]/lng:Jg00F3y06om5jar2wHauAQQYhE6WA8fi5pRX+mz+vdFg]
            [Member with value [...]: lng:Jg00F3y06om5jar2wHauAQQYhE6WA8fi5pRX+mz+vdFg]
            String value [...]: Jg00F3y06om5jar2wHauAQQYhE6WA8fi5pRX+mz+vdFg
            Key: lng
```

7) Start Hike

Test	Start hike from hike list
Pass	User prompted for duration of hike. Tracking screen displayed with hike information and location tracking service started. Check hike status scheduled on flask
Fail	Start hike button shows error and fails to start or navigate to tracking screen

Start hike request:

```
HTTP/J... 1055 PUT /hikes/start HTTP/1.1 , JSON (application/json)
TCP      44 5000 → 53228 [ACK] Seq=1 Ack=6852 Win=2154240 Len=0
TLSv1.3  97 Application Data
TCP      64 5432 → 52948 [ACK] Seq=9473 Ack=8562 Win=2151936 Len=0
TLSv1.3  103 Application Data
TCP      64 52948 → 5432 [ACK] Seq=8562 Ack=9512 Win=317696 Len=0
TLSv1.3  649 Application Data
TCP      64 5432 → 52948 [ACK] Seq=9512 Ack=9147 Win=2151424 Len=0
TLSv1.3  1009 Application Data
TCP      64 52948 → 5432 [ACK] Seq=9147 Ack=10457 Win=316672 Len=0
TLSv1.3  172 Application Data
TCP      64 5432 → 52948 [ACK] Seq=10457 Ack=9255 Win=2151168 Len=0
TLSv1.3  106 Application Data
TCP      64 52948 → 5432 [ACK] Seq=9255 Ack=10499 Win=316672 Len=0
TLSv1.3  98 Application Data
TCP      64 5432 → 52948 [ACK] Seq=10499 Ack=9289 Win=2151168 Len=0
TLSv1.3  104 Application Data
TCP      64 52948 → 5432 [ACK] Seq=9289 Ack=10539 Win=316672 Len=0
TCP      209 5000 → 53228 [PSH, ACK] Seq=1 Ack=6852 Win=2154240 Len=165 [TCP PDU reassembled in 320]
TCP      44 53228 → 5000 [ACK] Seq=6852 Ack=166 Win=2161152 Len=0
HTTP/J... 84 HTTP/1.1 200 OK , JSON (application/json)
```

Encrypted data:

```
JavaScript Object Notation: application/json
{
  "Object": {
    "Member: completed": [
      "[Path with value: /completed:false]",
      "[Member with value: completed:false]",
      "False value",
      "Key: completed",
      "[Path: /completed]"
    ],
    "Member: duration": [
      "[Path with value [...]: /duration:EIPpRZOS/Zzc4km7FBm3KMoM9sRDZaFbehVLUb5Cjqct+8GWUJ1TCyJVeyz+VONWc34c+DnD7p]", 
      "[Member with value [...]: duration:EIPpRZOS/Zzc4km7FBm3KMoM9sRDZaFbehVLUb5Cjqct+8GWUJ1TCyJVeyz+VONWc34c+DnD7p]", 
      "String value [...]: EIPpRZOS/Zzc4km7FBm3KMoM9sRDZaFbehVLUb5Cjqct+8GWUJ1TCyJVeyz+VONWc34c+DnD7piF\nnCLCptUpfyN", 
      "Key: duration",
      "[Path: /duration]"
    ],
    "Member: inProgress": [
      "[Path with value: /inProgress:false]",
      "[Member with value: inProgress:false]",
      "False value",
      "Key: inProgress",
      "[Path: /inProgress]"
    ],
    "Member: lat": [
      "[Path with value [...]: /lat:ONopKj0dq1FmmneSYPcXL2Jdwgr+2k38ArtrXjhXLFKogVGRxiv/0+Cm1qTTepx1BitUNiu3sAxr\n]", 
      "[Member with value [...]: lat:ONopKj0dq1FmmneSYPcXL2Jdwgr+2k38ArtrXjhXLFKogVGRxiv/0+Cm1qTTepx1BitUNiu3sAxr\n]", 
      "String value [...]: ONopKj0dq1FmmneSYPcXL2Jdwgr+2k38ArtrXjhXLFKogVGRxiv/0+Cm1qTTepx1BitUNiu3sAxr\n", 
      "Key: lat",
      "[Path: /lat]"
    ],
    "Member: lng": [
      "[Path with value [...]: /lng:VJlmmQJ70doPrgHQRMtas/dyn9Zki7QY/g383JBjF3wzd+06kMjC5W6s/7e1X2zZ3QRwMZpwhHqG\n]", 
      "[Member with value [...]: lng:VJlmmQJ70doPrgHQRMtas/dyn9Zki7QY/g383JBjF3wzd+06kMjC5W6s/7e1X2zZ3QRwMZpwhHqG\n]", 
      "String value [...]: VJlmmQJ70doPrgHQRMtas/dyn9Zki7QY/g383JBjF3wzd+06kMjC5W6s/7e1X2zZ3QRwMZpwhHqG\n", 
      "Key: lng",
      "[Path: /lng]"
    ],
    "Member: markers": [
      "Array",
      "Object",
      "Member: description": [
        "[Path with value [...]: /markers/[]/description:HDQsFLhEpok9m0W21zeCS2r1VCVG71G8ji/j6JJaPUF73NTzTLPh9SHBj5e/g]", 
        "[Member with value [...]: description:HDQsFLhEpok9m0W21zeCS2r1VCVG71G8ji/j6JJaPUF73NTzTLPh9SHBj5e/g]", 
        "String value [...]: HDQsFLhEpok9m0W21zeCS2r1VCVG71G8ji/j6JJaPUF73NTzTLPh9SHBj5e/gtubmSVQ56KRhcs\nBy"
      ],
      "Key: description",
      "[Path: /markers/[]/description]"
    ]
  }
}
```

8) Edit Hike

Test	Edit hike from hike list
Pass	Hike loaded into hike creation screen with existing details. Any changes are submitted to PostgreSQL upon confirmation of changes
Fail	Hike fails to update

Edit hike request:

```

HTTP/J... 326 PUT /hikes HTTP/1.1 , JSON (application/json)
TCP      44 5000 → 53230 [ACK] Seq=1 Ack=7583 Win=2153728 Len=0
TLSv1.3  97 Application Data
TCP      64 5432 → 52948 [ACK] Seq=10539 Ack=9322 Win=2151168 Len=0
TLSv1.3  103 Application Data
TCP      64 52948 → 5432 [ACK] Seq=9322 Ack=10578 Win=316672 Len=0
TLSv1.3  659 Application Data
TCP      64 5432 → 52948 [ACK] Seq=10578 Ack=9917 Win=2150656 Len=0
TLSv1.3  1012 Application Data
TCP      64 52948 → 5432 [ACK] Seq=9917 Ack=11526 Win=315648 Len=0
TLSv1.3  209 Application Data
TCP      64 5432 → 52948 [ACK] Seq=11526 Ack=10062 Win=2150400 Len=0
TLSv1.3  106 Application Data
TCP      64 52948 → 5432 [ACK] Seq=10062 Ack=11568 Win=315648 Len=0
TLSv1.3  98 Application Data
TCP      64 5432 → 52948 [ACK] Seq=11568 Ack=10096 Win=2150400 Len=0
TLSv1.3  104 Application Data
TCP      64 52948 → 5432 [ACK] Seq=10096 Ack=11608 Win=315648 Len=0
TCP      209 5000 → 53230 [PSH, ACK] Seq=1 Ack=7583 Win=2153728 Len=165
TCP      44 53230 → 5000 [ACK] Seq=7583 Ack=166 Win=2161152 Len=0
HTTP/J... 85 HTTP/1.1 200 OK , JSON (application/json)

```

Encrypted Data:

```

JavaScript Object Notation: application/json
{
  "Object": [
    {
      "Member": "completed",
      "Value": false,
      "Path": "/completed"
    },
    {
      "Member": "duration",
      "Value": "18gPfGGC5SRSuDUpfv7HoD",
      "Path": "/duration"
    },
    {
      "Member": "inProgress",
      "Value": true,
      "Path": "/inProgress"
    },
    {
      "Member": "lat",
      "Value": "40.7128° N, 74.0060° W",
      "Path": "/lat"
    },
    {
      "Member": "lng",
      "Value": "74.0060° W",
      "Path": "/lng"
    }
  ]
}

```

9) Delete Hike

Test	Delete hike from hike list
Pass	Deleted hike removed from list of hikes and from PostgreSQL
Fail	Hike remains in list or PostgreSQL

Delete request:

```

HTTP    168 DELETE /hikes/20 HTTP/1.1
TCP      44 5000 → 53375 [ACK] Seq=1 Ack=125 Win=2161152 Len=0
TLSv1.3  97 Application Data
TCP      64 5432 → 52948 [ACK] Seq=15343 Ack=12279 Win=2148352 Len=0
TLSv1.3  103 Application Data
TCP      64 52948 → 5432 [ACK] Seq=12279 Ack=15382 Win=311808 Len=0
TLSv1.3  649 Application Data
TCP      64 5432 → 52948 [ACK] Seq=15382 Ack=12864 Win=2147584 Len=0
TLSv1.3  703 Application Data
TCP      64 52948 → 5432 [ACK] Seq=12864 Ack=16021 Win=311296 Len=0
TLSv1.3  140 Application Data
TCP      64 5432 → 52948 [ACK] Seq=16021 Ack=12940 Win=2147584 Len=0
TLSv1.3  106 Application Data
TCP      64 52948 → 5432 [ACK] Seq=12940 Ack=16063 Win=311296 Len=0
TLSv1.3  98 Application Data
TCP      64 5432 → 52948 [ACK] Seq=16063 Ack=12974 Win=2147584 Len=0
TLSv1.3  104 Application Data
TCP      64 52948 → 5432 [ACK] Seq=12974 Ack=16103 Win=311040 Len=0
TCP      209 5000 → 53375 [PSH, ACK] Seq=1 Ack=125 Win=2161152 Len=165
TCP      44 53375 → 5000 [ACK] Seq=125 Ack=166 Win=2161152 Len=0
HTTP/J...  96 HTTP/1.1 200 OK , JSON (application/json)

```

10) Edit Account

Test	Change account details in edit account screen and submit changes
Pass	User account details updated on PostgreSQL
Fail	Account details fail to update

Edit account request:

```
HTTP/J... 1049 PUT /hikes HTTP/1.1 , JSON (application/json)
TCP      44 5000 → 53427 [ACK] Seq=1 Ack=6846 Win=2154240 Len=0
TLSv1.3  97 Application Data
TCP      64 5432 → 52948 [ACK] Seq=16103 Ack=13007 Win=2147584 Len=0
TLSv1.3  103 Application Data
TCP      64 52948 → 5432 [ACK] Seq=13007 Ack=16142 Win=311040 Len=0
TLSv1.3  659 Application Data
TCP      64 5432 → 52948 [ACK] Seq=16142 Ack=13602 Win=2147072 Len=0
TLSv1.3  798 Application Data
TCP      64 52948 → 5432 [ACK] Seq=13602 Ack=16876 Win=310272 Len=0
TLSv1.3  496 Application Data
TCP      64 5432 → 52948 [ACK] Seq=16876 Ack=14034 Win=2146560 Len=0
TLSv1.3  106 Application Data
TCP      64 52948 → 5432 [ACK] Seq=14034 Ack=16918 Win=310272 Len=0
TLSv1.3  98 Application Data
TCP      64 5432 → 52948 [ACK] Seq=16918 Ack=14068 Win=2146560 Len=0
TLSv1.3  104 Application Data
TCP      64 52948 → 5432 [ACK] Seq=14068 Ack=16958 Win=310272 Len=0
TCP      209 5000 → 53427 [PSH, ACK] Seq=1 Ack=6846 Win=2154240 Len=165
TCP      44 53427 → 5000 [ACK] Seq=6846 Ack=166 Win=2161152 Len=0
HTTP/J... 85 HTTP/1.1 200 OK , JSON (application/json)
```

Encrypted Data:

```
JavaScript Object Notation: application/json
▼ Object
  ▼ Member: completed
    [Path with value: /completed:false]
    [Member with value: completed:false]
    False value
    Key: completed
    [Path: /completed]
  ▼ Member: duration
    [Path with value [...]: /duration:SJ1Zr2DXeQA6Vh5mP220VVVPQVzqyq1JnSWLNWg6EYGWELHvAA]
    [Member with value [...]: duration:SJ1Zr2DXeQA6Vh5mP220VVVPQVzqyq1JnSWLNWg6EYGWELHvAA]
    String value [...]: SJ1Zr2DXeQA6Vh5mP220VVVPQVzqyq1JnSWLNWg6EYGWELHvAAxL9/hVUiSwiMbH9
    Key: duration
    [Path: /duration]
  ▼ Member: inProgress
    [Path with value: /inProgress:false]
    [Member with value: inProgress:false]
    False value
    Key: inProgress
    [Path: /inProgress]
  ▼ Member: lat
    [Path with value [...]: /lat:WPzFu/GelT/VY+6AihheMvBP0WkzLonIYXRY5PNcxkdXlp8paTz4Bjk]
    [Member with value [...]: lat:WPzFu/GelT/VY+6AihheMvBP0WkzLonIYXRY5PNcxkdXlp8paTz4Bjk]
    String value [...]: WPzFu/GelT/VY+6AihheMvBP0WkzLonIYXRY5PNcxkdXlp8paTz4BjkCdYLEZpuaz
    Key: lat
    [Path: /lat]
  ▼ Member: lng
    [Path with value [...]: /lng:HG/oE1+y91CFVe0JpfMXDdrzff8SV74oVDd2cGCsTebLA4T3YTejh]
    [Member with value [...]: lng:HG/oE1+y91CFVe0JpfMXDdrzff8SV74oVDd2cGCsTebLA4T3YTejhSyzy8bVZB]
    String value [...]: HG/oE1+y91CFVe0JpfMXDdrzff8SV74oVDd2cGCsTebLA4T3YTejhSyzy8bVZB
    Key: lng
    [Path: /lng]
  ▼ Member: markers
    ▼ Array
      ▼ Object
        ▼ Member: description
          [Path with value [...]: /markers/[]/description:dpD2HKWP8I0ckEFFf+1wgFWyW34W]
          [Member with value [...]: description:dpD2HKWP8I0ckEFFf+1wgFWyW34W0Y821wGIVUp/+mXxHj5JPiLZuu]
          String value [...]: dpD2HKWP8I0ckEFFf+1wgFWyW34W0Y821wGIVUp/+mXxHj5JPiLZuu
          Key: description
          [Path: /markers/[]/description]
```

11) Resume Hike

Test	Resume previously started but incomplete hike
Pass	Tracking screen loaded and showed appropriate traveled path
Fail	Fails to load tracking screen or previously traveled path missing

12) Resume Hike After Device Restart

Test	Resume hike after restart or location service dies
Pass	Restart's location service and continues hike from where it left off
Fail	Fails to re-start location service or display tracking screen

13) Complete Hike

Test	Complete hike from tracking screen
Pass	Hike marked as complete on PostgreSQL
Fail	Hike status fails to update

Complete hike request:

```
HTTP/J... 1045 PUT /hikes HTTP/1.1 , JSON (application/json)
TCP      44 5000 → 53441 [ACK] Seq=1 Ack=5382 Win=2155776 Len=0
TLSv1.3  97 Application Data
TCP      64 5432 → 52948 [ACK] Seq=19618 Ack=15116 Win=2145536 Len=0
TLSv1.3  103 Application Data
TCP      64 52948 → 5432 [ACK] Seq=15116 Ack=19657 Win=307712 Len=0
TLSv1.3  659 Application Data
TCP      64 5432 → 52948 [ACK] Seq=19657 Ack=15711 Win=2144768 Len=0
TLSv1.3  987 Application Data
TCP      64 52948 → 5432 [ACK] Seq=15711 Ack=20580 Win=306688 Len=0
TLSv1.3  364 Application Data
TCP      64 5432 → 52948 [ACK] Seq=20580 Ack=16011 Win=2144512 Len=0
TLSv1.3  106 Application Data
TCP      64 52948 → 5432 [ACK] Seq=16011 Ack=20622 Win=306688 Len=0
TLSv1.3  98 Application Data
TCP      64 5432 → 52948 [ACK] Seq=20622 Ack=16045 Win=2144512 Len=0
TLSv1.3  104 Application Data
TCP      64 52948 → 5432 [ACK] Seq=16045 Ack=20662 Win=306688 Len=0
TCP      209 5000 → 53441 [PSH, ACK] Seq=1 Ack=5382 Win=2155776 Len=165
TCP      44 53441 → 5000 [ACK] Seq=5382 Ack=166 Win=2161152 Len=0
HTTP/J... 85 HTTP/1.1 200 OK , JSON (application/json)
```

Encrypted data:

```
JavaScript Object Notation: application/json
▼ Object
  ▼ Member: completed
    [Path with value: /completed:false]
    [Member with value: completed:false]
    False value
    Key: completed
    [Path: /completed]
  ▼ Member: duration
    [Path with value [...]: /duration:igV09NmPz7zVAy0+QR1YGETkLOBoNY/]
    [Member with value [...]: duration:igV09NmPz7zVAy0+QR1YGETkLOBoNY]
    String value [...]: igV09NmPz7zVAy0+QR1YGETkLOBoNY/TpXCb/UxhuW1C
    Key: duration
    [Path: /duration]
  ▼ Member: inProgress
    [Path with value: /inProgress:false]
    [Member with value: inProgress:false]
    False value
    Key: inProgress
    [Path: /inProgress]
  ▼ Member: lat
    [Path with value [...]: /lat:B3YPcLoCT8a9IGCeeIoCn14j/RZZDra01e16]
    [Member with value [...]: lat:B3YPcLoCT8a9IGCeeIoCn14j/RZZDra01e16]
    String value [...]: B3YPcLoCT8a9IGCeeIoCn14j/RZZDra01e16emXsZoaut
    Key: lat
    [Path: /lat]
  ▼ Member: lng
    [Path with value [...]: /lng:klpZSxAgiP2rEQAqa0s1ztLw5P6wWLRALY2J]
    [Member with value [...]: lng:klpZSxAgiP2rEQAqa0s1ztLw5P6wWLRALY2J]
    String value [...]: klpZSxAgiP2rEQAqa0s1ztLw5P6wWLRALY2JCZrr68Jwd
    Key: lng
    [Path: /lng]
```

14) Alert sent for Uncompleted Hike

Test	Start hike with 1 minute duration and don't mark as completed
Pass	Alert sent with link to map detailing the hike information and the users starting and last known positions
Fail	Alert fails to send

api@demomailtrap.com

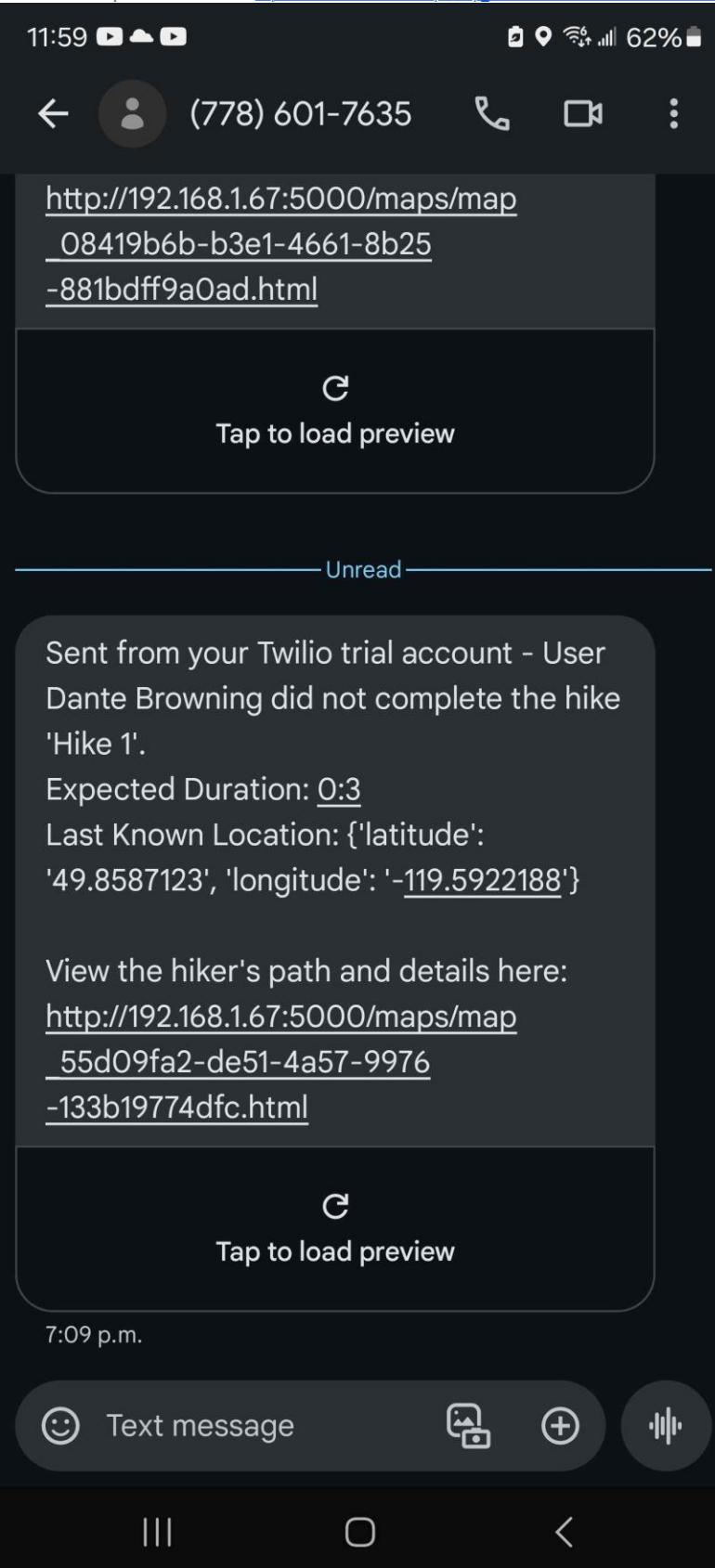
to me ▾

User Dante Browning did not complete the hike 'Hike 1'.

Expected Duration: 0:3

Last Known Location: {'latitude': '49.8587123', 'longitude': '-119.5922188'}

View the hiker's path and details here: http://192.168.1.67:5000/maps/map_55d09fa2-de51-4a57-9976-133b19774dfc.html



Full Manual Test Video:

https://drive.google.com/file/d/1IUGHkOXzFtljq_eQyb2Qob6D0xQ3iQg/view?usp=sharing

Link to Packet Capture File:

<https://drive.google.com/file/d/1RGZ0oPC4x8Hnj1OvLPntx9O9Qp2tlrLH/view?usp=sharing>

2.7. Implications of Implementation

The implementation of the Safer Hike App has several implications, both in technical and practical terms. These implications reflect how the app enhances user experience, contributes to safety, and poses challenges or opportunities for scalability.

Technical Implications

1. Security:

The application incorporates RSA keypair encryption to secure data during transmission between the app and the server. The public-private key system ensures that sensitive user data, such as location information and emergency contacts, is protected from potential breaches. The implementation of SSL/TLS encryption between the Flask server and the database further strengthens data privacy.

2. Integration of External Services:

The app makes use of Twilio and Flask-Mail for sending emergency alerts via SMS and email. This integration brings about the need for reliable network connectivity, service availability, and message deliverability. A failure in these services could hinder emergency notifications. Firebase authentication enables seamless user login and registration, which simplifies user management and improves security. However, reliance on third-party services like Firebase means the app is subject to their availability and policies.

3. Real-Time Tracking and Resource Usage:

Continuous location tracking and real-time updates of the user's position during a hike increase the demand on device resources such as battery life, data usage, and processing power. Optimizing location updates was crucial to ensure the app remains usable over extended periods of time. The Google Maps API integration for rendering maps and drawing polylines based on user data adds complexity. Updating the map in real-time required efficient state management, especially when dealing with background services and large sets of GPS data.

Practical Implications

1. User Safety and Emergency Response:

The app has a significant practical implication by potentially saving lives. Features like the automated alert system and real-time location sharing ensure that hikers who are in distress or missing can be located faster. This could reduce the time required for search and rescue operations. It also promotes responsible hiking practices by encouraging users to plan hikes, set markers, provide durations, and to stay connected with emergency contacts.

2. User Trust and Privacy

The implementation of encryption and secure data handling builds user trust. Hikers need assurance that their location data and emergency contacts are handled securely, especially in scenarios where this information is highly sensitive. Any breach or misuse of this data could damage user trust in the app, so maintaining strong security measures is critical.

3. User Experience

The design decisions around the user interface, especially for map integration and hike planning, have implications on user engagement. The ease of use, ability to place markers, and visualizing the hike route on a map all enhance the user experience. Any performance lag or confusing UI choices could reduce user satisfaction and disincline them to utilize the app.

2.8. Innovation

The Safer Hike App combines modern GPS tracking, personalized hike planning, and an automated emergency alert system into a single mobile platform designed specifically for outdoor enthusiasts. While existing apps offer individual elements of this system, none of them seamlessly integrate them into an all-in-one comprehensive package.

The key innovations in the app include:

- Automated Emergency Alerts:

The app provides a sophisticated system capable of automatically issuing emergency alerts to contacts if a user fails to complete a hike. Emergency contacts receive a message with a link to a map containing the user's last known location and associated hike details, allowing them to start a search and rescue effort if needed.

- Offline Data Handling:

Even in areas with no cellular service, the app queues data and syncs it when a connection becomes available, ensuring that the traveled route is appropriately recorded. This feature bridges the crucial gap between technology and the challenges of wilderness environments.

- Public-Private Key Encryption:

This security innovation ensures the safe transmission of sensitive user data, such as their emergency contacts or location data, safeguarding user's privacy and data integrity.

2.9. Complexity

The Safer Hike App involves several complex and technical design challenges, which justify its development as a BTech-level project. The complexity stems from multiple areas of the project:

1. Scalability and Data Security:

The app is designed to be scalable, handling data for many users while ensuring data security. Developing a secure server that stores sensitive information with public-private key encryption and SSL for database communication adds a layer of complexity

2. Emergency Alerts via SMS/Email:

Sending emergency alerts reliably using Twilio SMS and Flask-Mail for notifications requires integrating multiple services and ensuring that alerts are triggered accurately and without delay.

3. Learning Curve for New Technologies:

Implementing this project involved learning Jetpack Compose and Kotlin for android development, as well as working with APIs such as Google Maps and Twilio for the first time. Ensuring these technologies work together seamlessly increased the complexity of the project, especially in the context of managing asynchronous operations and background tasks within the app.

The project poses several complex challenges that demand advanced technical skills, extensive testing in field conditions, and a deep understanding of real-time systems and network security, making it suitable for a BTech-level project.

2.10. Research in New Technologies

Throughout the design and development of the Safer Hike App, several technologies and methodologies were researched and implemented:

1. Kotlin Programming:

As a modern programming language for Android development, Kotlin provides advantages such as null safety and concise syntax, which enhances code quality and maintainability. Learning Kotlin also improved my understanding of functional programming concepts and helped in leveraging Android's architectural components effectively.

2. GPS and Location Services:

Extensive research was conducted on using GPS tracking in mobile applications, particularly focused on implementing tracking through a foreground service to ensure reliable location updates, even when the app is not in the foreground. This approach allowed the app to maintain accurate tracking even when minimized or shutdown to conserve battery.

3. Jetpack Compose for UI Development:

Investigating Jetpack Compose enabled the creation of a responsive and intuitive user interface. The declarative UI approach simplifies UI development and enhances maintainability. Researching best practices in Compose helped in designing components that provide a seamless user experience while ensuring compatibility with various screen sizes and orientations.

4. Backend Development with Flask:

The use of Flask for developing the API allowed for efficient data handling. Researching Flask's extensions, such as Flask-Mail and Flask-SQLAlchemy, facilitated the integration of email notifications and database management securely and efficiently.

5. Twilio API for Messaging:

Exploring the Twilio API provided insights into reliable SMS notification systems. Understanding how to integrate third-party services for emergency alerts enhanced the app's responsiveness and reliability during critical situations.

2.11. Future Enhancements

The following future enhancements have been identified to further improve the Safer Hike App:

1. Enhanced User Interface:

Implementing a more dynamic user interface with additional customization options for hike plans and alerts can enhance the user experience. Incorporating user feedback to refine UI design could significantly improve the quality of the application.

2. Offline Maps Integration:

Allowing users to download maps for offline use could increase the app's functionality in areas with no cellular service. Integrating OpenStreetMap or similar services could provide users with essential navigation tools.

3. Machine Learning for Predictive Alerts:

Exploring machine learning algorithms to analyze user patterns and predict potential emergencies could lead to more proactive safety measures, providing alerts based on user behavior during hikes.

4. Integration of hardware devices:

Developing compatibility with other pieces of hardware such as smartwatches or tailor-made tracking devices could allow users to receive the benefits of the application without needing consistent access to their phones, providing greater convenience and safety during hikes.

5. Improved Emergency Features:

Enhancing the emergency alert system to include location sharing in real-time with emergency contacts or search and rescue teams could improve response times in emergency situations.

2.12. Timeline and Milestones

Task Name	Start	Finish	Hours
Setup Android Studio	7/4/2024	7/4/2024	2
Familiarize with Kotlin	7/4/2024	7/7/2024	22
Familiarize with Jetpack Compose	7/8/2024	7/10/2024	12
UI Design	7/10/2024	7/21/2024	40
Configure Flask Server	7/21/2024	7/21/2024	2
Configure PostgreSQL	7/22/2024	7/22/2024	6
Basic UI Implementation	7/23/2024	7/26/2024	12
App Communication to Flask	7/27/2024	7/29/2024	12
Flask Communication to PostgreSQL	7/30/2024	7/31/2024	12
Milestone 1: App Foundation	7/4/2024	7/31/2024	120
Configure Firebase Auth	8/1/2024	8/1/2024	2
Implement Signup/Login	8/1/2024	8/6/2024	26
Implement Hike Creation	8/7/2024	8/13/2024	24
Implement Hike List View	8/14/2024	8/16/2024	12
Implement Account Editing	8/17/2024	8/19/2024	12
Implement Hike Edits/Deletion	8/20/2024	8/23/2024	16
Implement LocationService	8/24/2024	8/29/2024	28
Implement HikeTracking	8/30/2024	9/7/2024	40
Implement Scheduled Alerts	9/8/2024	9/10/2024	10
Add Encryption	9/11/2024	9/13/2024	16
Add SSL between Flask and PostgreSQL	9/14/2024	9/15/2024	8
Add SMS/Email sending to alerts	9/16/2024	9/16/2024	8
Milestone 2: All Features Complete	8/1/2024	9/16/2024	202
Flask Unit Testing	9/17/2024	9/18/2024	12
Manual App Testing	9/19/2024	9/25/2024	30
Bug Fixes	9/17/2024	9/25/2024	40
Milestone 3: Testing Complete	9/17/2024	9/25/2024	82
Making Diagrams	9/26/2024	9/27/2024	16
Documenting Testing	9/27/2024	9/29/2024	16
Writing Report	9/29/2024	10/2/2024	22
Milestone 4: Report Done		10/2/2024	54
Total Time Spent	7/4/2024	10/2/2024	458 Hours

3. Conclusion

The Safer Hike App represents a significant step forward in wilderness safety applications, merging modern technology with practical safety measures. The project has not only allowed me to develop valuable technical skills but has also reinforced the importance of creating solutions that prioritize user safety in remote environments. As I move forward in my career, I am excited to continue exploring innovative solutions that address real-world challenges through technology.

3.1. Lessons Learned

Developing the Safer Hike App provided me with valuable insights beyond classroom knowledge. I learned the importance of balancing functionality and user experience, especially in emergency scenarios where ease of use is paramount. The integration of Jetpack Compose, Kotlin, and foreground services challenged me to adapt to new technologies. Additionally, I gained a deeper understanding of data security through the implementation of my encryption techniques. This project underscored the significance of securing user data in today's digital landscape and has equipped me with innovative problem-solving skills applicable to real-world challenges.

3.2. Closing Remarks

The development of this project has been a tremendously rewarding and insightful experience. Building the Safer Hike App has not only tested my technical skills, but has shown me the importance of creativity, innovation, and problem-solving in real-world applications. This app, designed with a heart for safety and practicality, is a testament to the power of technology in assisting people in life-threatening situations.

What started as a simple idea to enhance wilderness safety evolved into a multifaceted solution with the potential to make a tangible difference in emergency response times to missing hikers. The project pushed me to explore uncharted territories – both in terms of technology and user-centered design – and I feel a sense of deep satisfaction in what I have accomplished.

This project marks the beginning of many more future explorations into safety focused technology, and it has solidified my belief that, through innovation, we can tackle even the most complex challenges and make the world a safer, more connected place.

4. Appendix

4.1. Approved Proposal

<https://drive.google.com/file/d/1rq84LwPW1VH2MG2q1vpjYSAJdF48Mv5C/view?usp=sharing>

4.2. Project Supervisor Approvals

[Include written approvals from the project supervisor indicating that they've approved both the proposal and report. Also include any changes that have been approved by the project supervisor. Please note that without written approvals from the project supervisor, the committee may not review the final report.]

5. References

JetPack Compose UI App Development Toolkit - Android Developers. (n.d.). Android Developers.
<https://developer.android.com/compose>

Android Developers. (n.d.). *Android Mobile App Developer Tools – Android Developers.*
<https://developer.android.com/develop/sensors-and-location/location>

Kotlin Docs | Kotlin. (n.d.). Kotlin Help. <https://kotlinlang.org/docs/home.html>

Philipp Lackner. (2023, February 1). *The JetPack Compose Beginner Crash Course for 2023*  (Android Studio Tutorial) [Video]. YouTube. https://www.youtube.com/watch?v=6_wK_Ud8--0

Philipp Lackner. (2023b, June 18). *ViewModels & Configuration Changes - Android Basics 2023* [Video]. YouTube. <https://www.youtube.com/watch?v=9sqvBydNJSg>

Philipp Lackner. (2021, January 13). *The ultimate Retrofit crash course* [Video]. YouTube.
<https://www.youtube.com/watch?v=t6Sql3WMAnk>

Philipp Lackner. (2022, September 28). *How to track your users location in the background in Android - Android Studio tutorial* [Video]. YouTube. <https://www.youtube.com/watch?v=Jj14sw4Yxk0>

Docs: API reference, tutorials, and integration | Twilio. (n.d.). <https://www.twilio.com/docs>

Flask-Mail. (2024, May 23). PyPI. <https://pypi.org/project/Flask-Mail/>

SQLAlchemy Documentation — SQLAlchemy 2.0 documentation. (n.d.).
<https://docs.sqlalchemy.org/en/20/>