Here's the PDF-styled string you requested:

Fitbit to Roboflow Universe: Healthy Route Tracking

Use Fitbit's API to track user routes and then use Roboflow Universe's API to analyze the route for potential hazards or obstacles.

[*Fitbit* (https://dev.fitbit.com/), *Roboflow Universe* (https://universe.roboflow.com)]

Now, let's create a base structure for the project in Java. We'll need the following files:

1. `Main.java` - the entry point of our application

2. `FitbitAPI.java` - to interact with Fitbit's API

3. `RoboflowUniverseAPI.java` - to interact with Roboflow Universe's API

4. `RouteAnalyzer.java` - to analyze the route for potential hazards or obstacles

5. `User.java` - to represent a user and their tracked routes

Here's a starting point for each file:

Main.java

```java
public class Main {
    public static void main(String[] args) {
        // Initialize Fitbit API client
        FitbitAPI fitbitAPI = new FitbitAPI();
        // Initialize Roboflow Universe API client
        RoboflowUniverseAPI roboflowUniverseAPI = new RoboflowUniverseAPI();

        // Get user's tracked routes from Fitbit API
        List<Route> routes = fitbitAPI.getRoutes();

        // Analyze each route for potential hazards or obstacles using Roboflow Universe API
        for (Route route : routes) {
            RouteAnalyzer routeAnalyzer = new RouteAnalyzer(route);
            routeAnalyzer.analyze();
        }
    }
}
```

FitbitAPI.java

```java
public class FitbitAPI {
    public List<Route> getRoutes() {
        // Implement logic to retrieve user's tracked routes from Fitbit API
        // For now, just return a sample list of routes
        List<Route> routes = new ArrayList<>();
        Route route1 = new Route("Route 1", "2022-01-01T12:00:00Z", "2022-01-01T13:00:00Z");

        Route route2 = new Route("Route 2", "2022-01-02T14:00:00Z", "2022-01-02T15:00:00Z");

        routes.add(route1);

        routes.add(route2);

        return routes;

    }

}
```

RoboflowUniverseAPI.java

```java

public class RoboflowUniverseAPI {

    public List<Hazard> analyzeRoute(Route route) {

        // Implement logic to analyze the route for potential hazards or obstacles using Roboflow Universe API

        // For now, just return a sample list of hazards

        List<Hazard> hazards = new ArrayList<>();

        Hazard hazard1 = new Hazard("Hazard 1", "2022-01-01T12:00:00Z");

        Hazard hazard2 = new Hazard("Hazard 2", "2022-01-02T14:00:00Z");

        hazards.add(hazard1);

        hazards.add(hazard2);

        return hazards;
    }
}
```

RouteAnalyzer.java

```java
public class RouteAnalyzer {
    private Route route;

    public RouteAnalyzer(Route route) {
        this.route = route;
    }

    public void analyze() {
        // Call Roboflow Universe API to analyze the route for potential hazards or obstacles
        List<Hazard> hazards = new RoboflowUniverseAPI().analyzeRoute(route);

        // Process the hazards found in the route
        // For now, just print them out
        System.out.println("Hazards found on this route:");
        for (Hazard hazard : hazards) {
            System.out.println(hazard.getName());
        }
    }
}
```

User.java

```java
public class User {
    private String username;
    private List<Route> routes;

    public User(String username) {
        this.username = username;
        this.routes = new ArrayList<>();
    }

    public void addRoute(Route route) {
        routes.add(route);
    }

    public List<Route> getRoutes() {
        return routes;
    }
}
```

```

This is a starting point for your project. You'll need to implement the logic in each file to interact with Fitbit's API, Roboflow Universe's API, and analyze the route for potential hazards or obstacles. Good luck!
```