```java

/*****
 File Name: MainActivity.java
 This will be your Android activity where the user (rental company staff) can
 set speed limits for customers and check their speeds.

 */

package com.example.mycarrental;

import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;
import com.example.mycarrental.R;



public class MainActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";
    private RentalCar rentalCar;
    private String communicationChannel;// = "Firebase";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        String[] communicationChannels = {"Firebase", "AWS"};

        // Initialize Notification Service (Firebase-based)
        NotificationService notificationService = new NotificationService(this);

        // Initialize RentalCar instance
        rentalCar = new RentalCar(notificationService);


        // Get input fields and buttons
        EditText customerIdInput = findViewById(R.id.customerIdInput);
        EditText speedLimitInput = findViewById(R.id.speedLimitInput);
        Button setLimitButton = findViewById(R.id.setLimitButton);
        Button checkSpeedButton = findViewById(R.id.checkSpeedButton);
        Spinner communicationChannelSpinner  = findViewById(R.id.channelTypeSpinner);

        // Create an adapter to bind the data to the Spinner
        ArrayAdapter<String> adapter = new ArrayAdapter<>(
                this,
                android.R.layout.simple_spinner_item,
                communicationChannels
        );

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

        // Attach the adapter to the Spinner
        communicationChannelSpinner.setAdapter(adapter);

        communicationChannelSpinner.setOnItemSelectedListener(new AdapterView.
        OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position,
            long id) {
                communicationChannel = parent.getItemAtPosition(position).toString();
                Toast.makeText(getApplicationContext(), "Selected: " +
                    communicationChannel, Toast.LENGTH_SHORT).show();
                }
                @Override
                public void onNothingSelected(AdapterView<?> parent) {
                    communicationChannel = communicationChannels[0]; //usually Spinner
                    always selects an item, in case if it doesn't, i am setting it to
                    firebase
                    Toast.makeText(getApplicationContext(), "No communication channel
                    selected, defaulting to Firebase", Toast.LENGTH_SHORT).show();
                }
            });


            // Set speed limit for a customer
            setLimitButton.setOnClickListener(v -> {
                String customerId = customerIdInput.getText().toString().trim();
                String speedLimitText = speedLimitInput.getText().toString().trim();

                if (customerId.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter a customer ID", Toast.
                    LENGTH_SHORT).show();
                    return;
                }

                if (speedLimitText.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter a speed limit", Toast.
                    LENGTH_SHORT).show();
                    return;
                }

                try {
                    double speedLimit = Double.parseDouble(speedLimitText);
                    if (speedLimit <= 0) {
                        Toast.makeText(MainActivity.this, "Speed limit must be positive",
                        Toast.LENGTH_SHORT).show();
                        return;
                    }
                    Log.d(TAG, "Setting speed limit for customer: " + customerId + " to " +
                    speedLimit);
                    rentalCar.setSpeedLimitForCustomer(customerId, speedLimit,
                    communicationChannel);
                    Toast.makeText(MainActivity.this, "Speed limit set successfully", Toast.
                    LENGTH_SHORT).show();
                } catch (NumberFormatException e) {
                    Toast.makeText(MainActivity.this, "Invalid speed limit entered", Toast.
                    LENGTH_SHORT).show();
                }
            });

            // Check if the current speed exceeds the speed limit
            checkSpeedButton.setOnClickListener(v -> {
                String customerId = customerIdInput.getText().toString().trim();
                if (customerId.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter a customer ID", Toast.
                    LENGTH_SHORT).show();
                    return;
                }
                rentalCar.checkSpeed(customerId);
            });

            }
    }
```

```java
/*****
        File Name: RentalCar.java
        This class handles the main logic of setting speed limits for different customers,
        checking their speed, and notifying the rental company when limits are exceeded.

*/

package com.example.mycarrental;
import android.content.Context;
import android.util.Log;

import java.util.HashMap;
import java.util.Map;

public class RentalCar {


    private final Map<String, Customer> customers = new HashMap<>();
    private final NotificationService notificationService;
    private final CarPropertyHandler propertyHandler = new CarPropertyHandler();

    private static final String TAG = "RentalCar";
    public RentalCar(NotificationService notificationService) {
        this.notificationService = notificationService;
    }

    // Set speed limit for a customer
    public void setSpeedLimitForCustomer(String customerId, double limit, String communicationChannel) {
        customers.put(customerId, new Customer(customerId, limit, communicationChannel));
    }

    // Check if the current speed exceeds the speed limit
    public void checkSpeed(String customerId) {
        // currentSpeed can be read from CarPropertyManager
        if (customers.containsKey(customerId)) {
            Customer customer = customers.get(customerId);
            double currentSpeed = propertyHandler.getCurrentVehicleSpeed();
            if (customer != null && currentSpeed > customer.getSpeedLimit()) {
                Log.d(TAG, "Speed exceeded for customer " + customerId + ": " + currentSpeed + " > " + customer.getSpeedLimit());
                sendSpeedNotification(customerId, currentSpeed, customer.getSpeedLimit(), customer.getCommunicationChannel());
            } else {
                Log.d(TAG, "Speed within limit for customer " + customerId);
            }
        }    else {
                Log.d(TAG, "Customer ID not found!" + customerId);
            }
    }

    // Send a notification if speed exceeds limit
    private void sendSpeedNotification(String customerId, double currentSpeed, double speedLimit, String channel) {
        String message = "ALERT: Customer " + customerId + " exceeded the speed limit of " +
                speedLimit + " km/h. Current speed is: " + currentSpeed + " km/h.";
        notificationService.sendCustomerNotification(message, channel);
    }

}
```

```java
/********
        File Name: Customer.java
*****   This class manages a customer details such as customer ID, speed limit and
Communication Channel.
 *      from this class we can retrive the customer speed Limit, channel details
*/

package com.example.mycarrental;

import android.util.Log;

public class Customer {
    private String customerId;
    private double speedLimit;
    private String communicationChannel;  // "Firebase" or "AWS"

    private static final String TAG = "Customer";
    public Customer(String customerId, double speedLimit, String notificationChannel) {
        this.customerId = customerId;
        this.speedLimit = speedLimit;
        this.communicationChannel = notificationChannel;
        Log.d(TAG, "Customer, Speed Limit, channel details are set successfully");
    }

    public String getCustomerId() {
        return customerId;
    }

    public double getSpeedLimit() {
        //Log.d(TAG, "Returning Customer Speed Limit details");
        return speedLimit;
    }

    public String getCommunicationChannel() {
        Log.d(TAG, "Returning Customer NotificationChannel details");
        return communicationChannel;
    }

    public void setSpeedLimit(double speedLimit) {
        this.speedLimit = speedLimit;
    }

    public void setNotificationChannel(String notificationChannel) { this.
    communicationChannel = notificationChannel; }
}


```

```java
/******
     *
     File Name: CarPropertyHandler.java
     This class register's for the vehicle speed changes.
     everytime there is a change in the speed of the vehicle there is a CB trigger to
     property handler.
     *
 */

package com.example.mycarrental;

import android.car.Car;
import android.car.hardware.CarPropertyManager;
import android.content.Context;
import android.os.Bundle;
import android.util.Log;

public class CarPropertyHandler {

    private static final String TAG = "CarPropertyHandler";
    private CarPropertyManager mCarPropertyManager;
    private CarPropertyManager.OnPropertyChangedListener mSpeedChangedListener;
    private volatile float mCurrentSpeed = 0f; // Holds the latest known speed

    public CarPropertyHandler() {
         Car car = Car.createCar(context);
        mCarPropertyManager = (CarPropertyManager) car.getCarManager(Car.PROPERTY_SERVICE
        );
        Log.d(TAG, "Car and CarPropertyManager initialized.");

        // Safe to register callbacks now
        startListeningForSpeedChanges(); // automatically start listening to speed
        changes
    }


    public void startListeningForSpeedChanges() {
        // guard against multiple registerCallback() calls
        if (mSpeedChangedListener != null) return;

        // Create a listener for property changes
        mSpeedChangedListener = new CarPropertyManager.OnPropertyChangedListener() {
            @Override
            public void onPropertyChanged(CarPropertyManager.CarPropertyValue
            propertyValue) {
                if (propertyValue != null &&
                        propertyValue.getValue() != null &&
                        propertyValue.getPropertyId() == CarPropertyManager.
                        PROPERTY_SPEED) {
                    // Check if the property is the speed property
                    float speed = (Float) propertyValue.getValue();
                    mCurrentSpeed = speed; // Store latest speed
                    Log.d(TAG, "Car speed changed: " + speed + " km/h");
                }
            }
        };

        // Register the listener for speed changes
        try {
            mCarPropertyManager.registerCallback(mSpeedChangedListener,
                                                    CarPropertyManager.PROPERTY_SPEED,
                                                    CarPropertyManager.
                                                    SENSOR_RATE_ONCHANGE);
        } catch (SecurityException e) {
            Log.e(TAG, "Permission issue when registering listener", e);
        }

    }

    public void stopListeningForSpeedChanges() {
        // Unregister the listener when work is done
        if (mSpeedChangedListener != null) {
            mCarPropertyManager.unregisterCallback(mSpeedChangedListener);
            mSpeedChangedListener = null;
        }

    }

    /**
     * Returns the most recently received speed value.
     *
     * @return speed in km/h (or as provided by the vehicle)
     */
    public float getCurrentVehicleSpeed() {
        Log.d(TAG, "Returning current car speed in km/h");
        return mCurrentSpeed;
    }
}

```

```java
/******
        File Name: NotificationService.java
        This class handles sending notifications.
        In this case, we are use
            1. Firebase Cloud Messaging to notify the rental company if the speed limit
            is exceeded.
            2. AWS Notification
            3. Audio Alert to user
            4. Toast msg to user
*/

package com.example.mycarrental;

import android.content.Context;
import android.util.Log;
import android.widget.Toast;
import android.media.MediaPlayer;

public class NotificationService {

    private final Context context;

    UserNotification userMsgNotification = new UserNotification();
    private static final String TAG = "NotificationService";

    public NotificationService(Context context) {
        this.context = context;
    }

    public void sendCustomerNotification(String message, String channel) {
        Log.d(TAG, "Sending notifications to user and rental Car Owner");

        INotificationService notificationService = getNotificationService(channel);
        if (notificationService != null) {
            notificationService.sendNotification(message);
        } else {
            Log.e(TAG, "No notification service available for channel: " + channel);
        }

        userMsgNotification.playAudioAlert(context);
        userMsgNotification.sendToastNotification(context, message);
    }

    private INotificationService getNotificationService(String channel) {
        if ("Firebase".equalsIgnoreCase(channel)) {
            return new FcmNotification();
        } else if ("AWS".equalsIgnoreCase(channel)) {
            return new AwsNotification();
        } else {
            return null;
        }
    }
}
```

```java

/******
        File Name: INotificationService.java
        This interface defines the contract for a notification service within the Car
        Rental application.
        Implementing classes will provide specific method definition for sending
        notifications,
*/

package com.example.mycarrental;

public interface INotificationService {
    void sendNotification(String message);
}

```

```java
/******
     *
     File Name: FcmNotification.java
     This class handles sending Firebase Cloud Message notifications.
     *
 */


package com.example.mycarrental;
import com.google.firebase.messaging.FirebaseMessaging;
import com.google.firebase.messaging.Message;

import android.util.Log;

public class FcmNotification implements INotificationService {

    private static final String TAG = "FcmNotification";
    @Override
    public void sendNotification(String message) {
        Message message = Message.builder()
                .putData("Alert", "Your rental car speed is exceeding the speedLimit!")
                .setTopic("car_rentals")
                .build();
        Log.d(TAG, "Successfully sent Firebase message:.");

        try {
            String response = FirebaseMessaging.getInstance().send(message);
            Log.d(TAG,"Successfully sent Firebase message: " + response);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```java
/******
         File Name: AwsNotification.java
         This class handles sending AWS notifications.

*/

package com.example.mycarrental;

import android.util.Log;

public class AwsNotification implements INotificationService{

    private static final String TAG = "AwsNotification";

    @Override
    public void sendNotification(String message) {
        try {
            PublishRequest publishRequest = new PublishRequest();
            publishRequest.setTopicArn("car_rentals");
            publishRequest.setMessage("Your rental car speed is exceeding the
            speedLimit!");
            publishRequest.setSubject("Car Rental Alert");

            AmazonSNSClient snsClient = new AmazonSNSClient(credentialsProvider);
            PublishResult result = snsClient.publish(publishRequest);

            Log.d(TAG, "Message sent. MessageId: " + result.getMessageId());
        } catch (Exception e) {
            Log.e(TAG, "Error sending SNS notification", e);
        }
        Log.d(TAG, "Successfully sent AWS's SNS(Simple Notification Service) message:.");
    }
}
```

```java
/******
     *
     File Name: UserNotification.java
     This class handles sending User notifications.
     such as audio and visual notifications to the user
         1. Audio Alert to user
         2. Toast msg to user
     *
 */

package com.example.mycarrental;

import android.media.MediaPlayer;
import android.util.Log;
import android.widget.Toast;
import android.content.Context;
public class UserNotification {

    private static final String TAG = "UserNotification";
    public void playAudioAlert(Context context) {
        // Place your audio file (e.g., alert_sound.mp3) in the res/raw/alert_sound.mp3. Use MediaPlayer to play the audio alert.
        MediaPlayer mediaPlayer = MediaPlayer.create(context, R.raw.alert_sound);
        Log.d(TAG, "Playing Audio Alerts to user");
        mediaPlayer.start();
    }
    public void sendToastNotification(Context context, String message) {
        Log.d(TAG, "Displaying Toast Msg Alerts to user");
        Toast.makeText(context, message, Toast.LENGTH_LONG).show();

    }

}
```