Cuvva Mobile Developer Tech Test

Introduction

The point of this task is to demonstrate your thinking process, deductive reasoning, problem solving and how well you can interpret and implement a brief.

The way you approach this test is completely up to you as long as you achieve as much as possible of the following tasks within a reasonable time frame. Document any assumptions made in your project README.md along with any comments or feedback on the instructions given. Also highlighted in the readMe anything you would like us to consider or pay particular attention to along with a list of "Things I would include / do differently if I had more time"

The use of third party Code/SDKs is allowed, but you should be able to explain why you have chosen the third party.

In this pack you will find

- 3 images that show the desired screen layout for the app
  - HomePage.png
  - PolicyPage.png
  - ReciptPage.png
- Three car manufacturer logos
- Two API endpoints
  - Policy Event Data Stream API
  - Standard Text API
- Car logos
  - xml files for Android
  - png files for iOS
- Fonts:
  - use Roboto for Android
  - San Francisco for iOS

The Main Tasks

Retrieve the policy event stream from the end point : https://cuvva.herokuapp.com

- Process the event data into a format that can be used to support the screen designs.
- Implement a navigation from the home screen to the policy screen and then to the receipt screen.
- The app only needs to work in portrait mode
- The app needs to work on the following OS versions (relevant to the role applied for):
  - iOS : v11.0 to latest
  - Android : v5.0.1 to latest

- Once the API data has been successfully retrieved the app should be able to continue working even when not connected to the internet and after a forced close of the app. (e.g. data should be persisted)

Processing the Event Stream

The event stream data contains multiple events for multiple policies.
Each event has an event type, a timestamp, a unique key and a variable content payload.
The description and processing instructions for each type are as follows:

1. Policy Created

This occurs when a new insurance policy has been created. It may be a brand-new policy, or it may be an extension to an existing policy. If the values of the “policy_id” and “original_policy_id” within the payload are different then this is an extension policy and should be considered a child of the original policy.

2. Policy Financial Transaction

This occurs when there has been a financial transaction on the policy. There may be multiple transactions for a single policy. The details in the payload are values in pence and could be negative as well as positive. All policy financial transaction objects should be displayed in the receipt screen in date order.

3. Policy Cancelled

This occurs when a policy has been cancelled. There will only be one of these events per policy or extension policy.

Important Notes about Policies

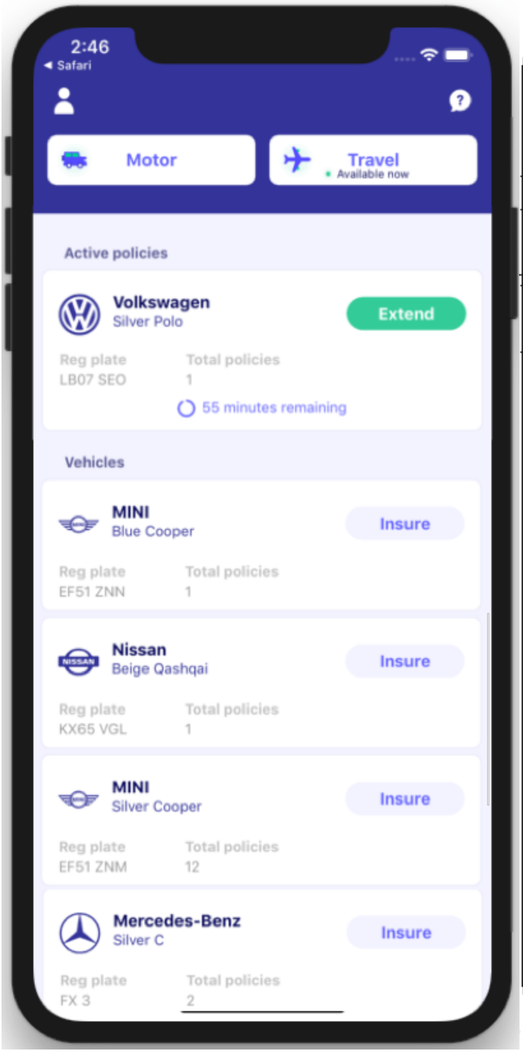
- A **Policy Start Time** is derived from the “start_date” of the “policy_created” payload.
- A **Policy End Time** is derived from the “end_date” of the “policy_created” payload OR the maximum “end_date” of any non-void extension policies.
- An “Active” policy is one where the current real-world time is between the **Policy Start Time** and **Policy End Time**
- For display purposes all policies should be grouped into the car that they cover

Optional Tasks

Implement the “Standard Text” API endpoint
http://www.mocky.io/v2/5c699176370000a90a07fd6f
Use the content of this endpoint to control the relevant elements of the design. We will change the content of the data on this endpoint to determine if the app displays the updated content.

Conclusion

Once you've completed the test, please upload the project to GitHub and send the link to Chris / Matt / Oli. Feel free to add any comments or notes to your submission that you would like us to consider. We would like to know roughly how long the task took to complete. After we have considered your submission you may be asked to discuss your solution either face to face or over a video / screen sharing call.



Active policies

Volkswagen
Silver Polo

Reg plate	Total policies
LB07 SEO	1

55 minutes remaining

Vehicles

MINI
Blue Cooper

Reg plate	Total policies
EF51 ZNN	1

Nissan
Beige Qashqai

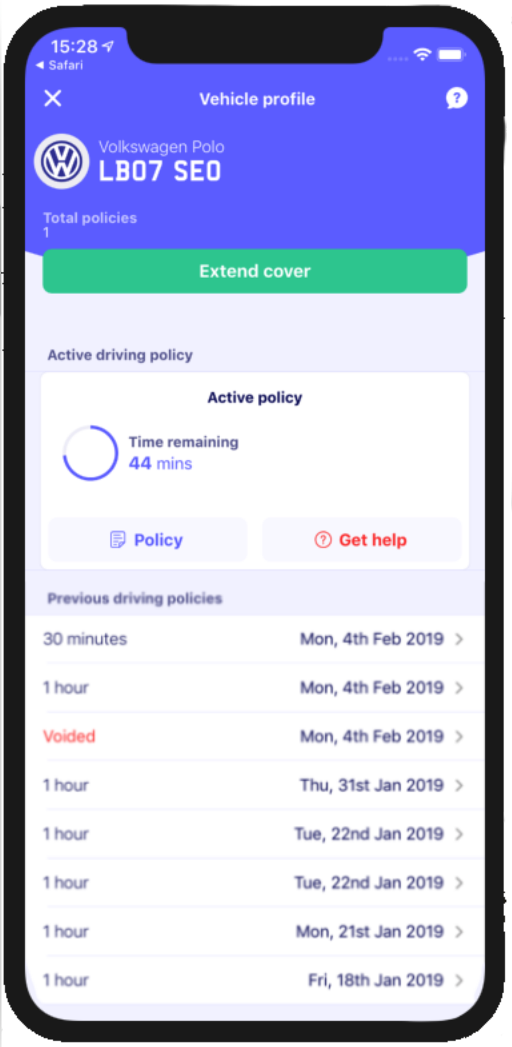
Reg plate	Total policies
KX65 VGL	1

MINI
Silver Cooper

Reg plate	Total policies
EF51 ZNM	12

Mercedes-Benz
Silver C

Reg plate	Total policies
FX 3	2



Volkswagen Polo
LB07 SEO

Total policies
1

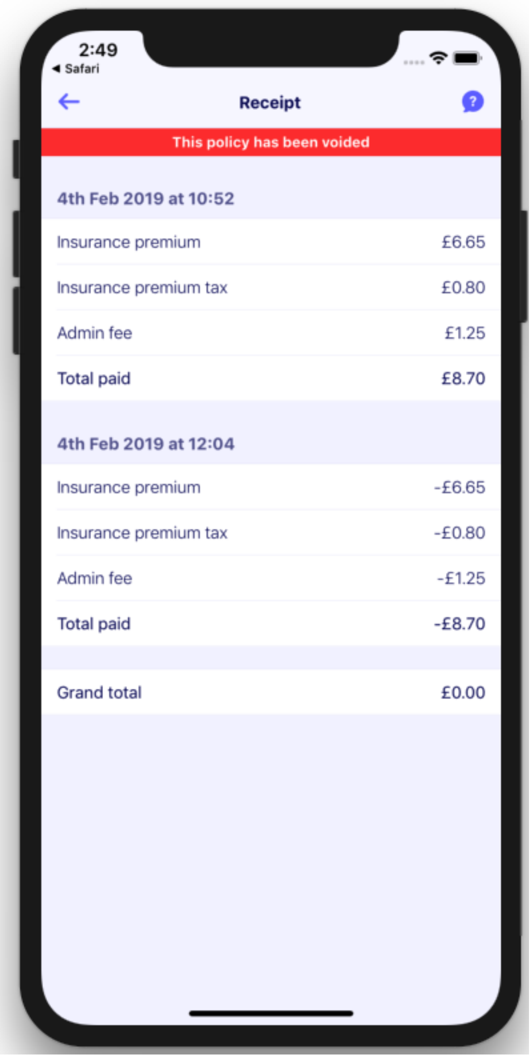
Active driving policy

Active policy

Time remaining
44 mins

Previous driving policies

30 minutes	Mon, 4th Feb 2019 ›
1 hour	Mon, 4th Feb 2019 ›
Voided	Mon, 4th Feb 2019 ›
1 hour	Thu, 31st Jan 2019 ›
1 hour	Tue, 22nd Jan 2019 ›
1 hour	Tue, 22nd Jan 2019 ›
1 hour	Mon, 21st Jan 2019 ›
1 hour	Fri, 18th Jan 2019 ›



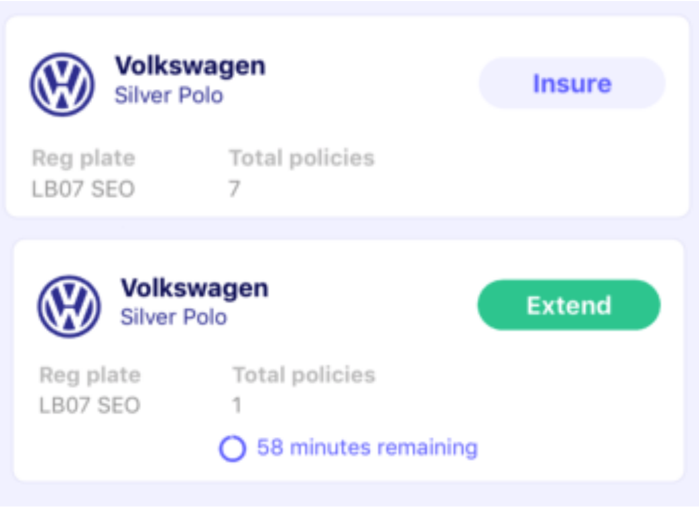
Receipt

This policy has been voided

4th Feb 2019 at 10:52	
Insurance premium	£6.65
Insurance premium tax	£0.80
Admin fee	£1.25
Total paid	**£8.70**

4th Feb 2019 at 12:04	
Insurance premium	-£6.65
Insurance premium tax	-£0.80
Admin fee	-£1.25
Total paid	**-£8.70**

Grand total	£0.00



Volkswagen
Silver Polo

Reg plate	Total policies
LB07 SEO	7

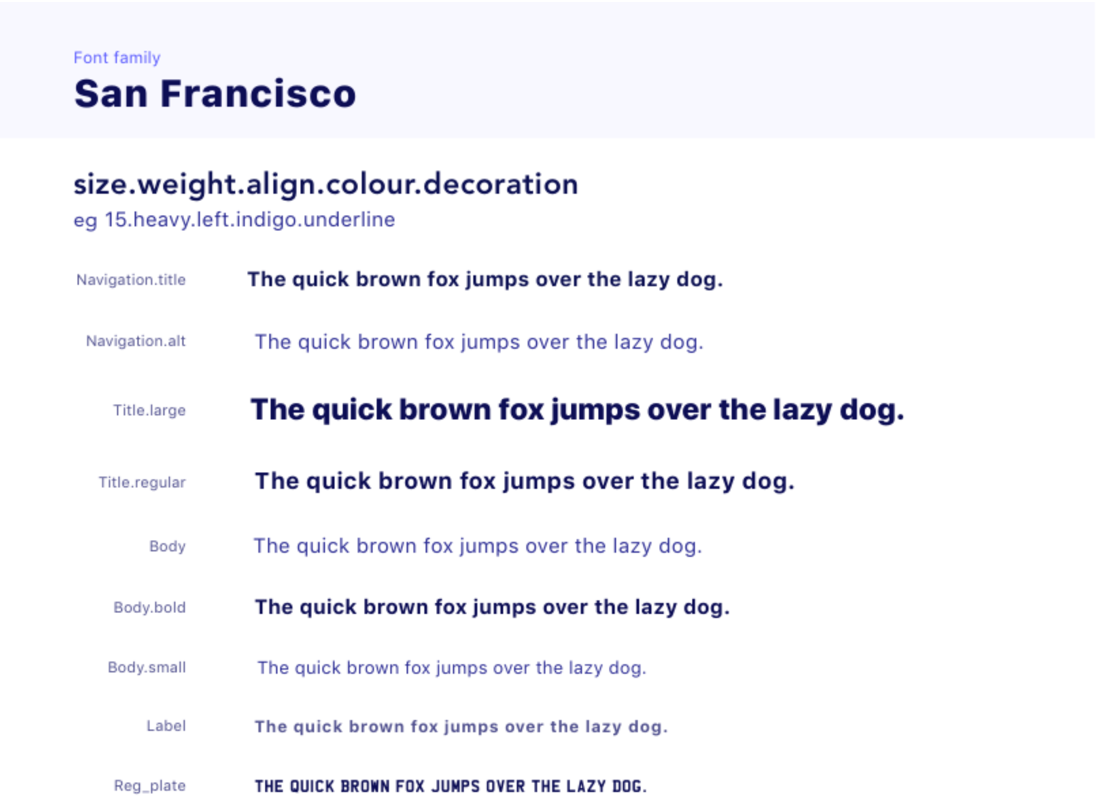
Volkswagen
Silver Polo

Reg plate	Total policies
LB07 SEO	1

○ 58 minutes remaining

Fonts

- Roboto for Android
- San Francisco for iOS



Font family

San Francisco

size.weight.align.colour.decoration

eg 15.heavy.left.indigo.underline

Navigation.title	**The quick brown fox jumps over the lazy dog.**
Navigation.alt	The quick brown fox jumps over the lazy dog.
Title.large	**The quick brown fox jumps over the lazy dog.**
Title.regular	**The quick brown fox jumps over the lazy dog.**
Body	The quick brown fox jumps over the lazy dog.
Body.bold	**The quick brown fox jumps over the lazy dog.**
Body.small	The quick brown fox jumps over the lazy dog.
Label	**The quick brown fox jumps over the lazy dog.**
Reg_plate	**THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.**

Assets

- Use xml for Android
- png for iOS

Colours

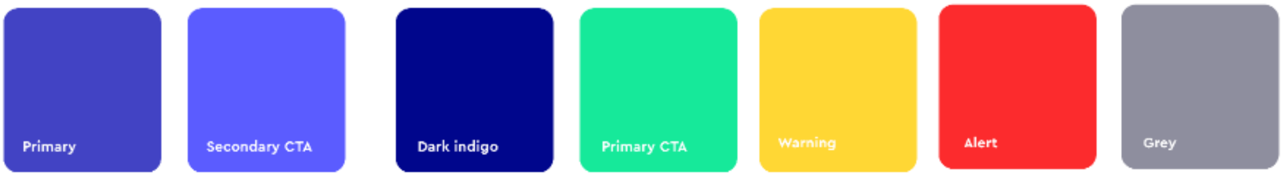
- Use ColorZilla picker Chrome extension
- http://www.colorzilla.com/chrome/

Colour

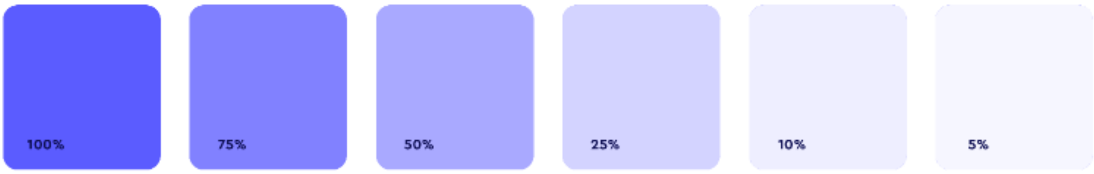
A minimal colour palette with a distinct focus on heirachy using colour to help guide a more clear understanding and progression.

Primary

Secondary



Shading



Taking action

Action green is the leading colour on any screen with the intention towards a path discovery and productivity.

R 231
G 236
B 164

Shades to create depth

Our use of depth is use sparingly in order to give a sense of depth and heirachy.