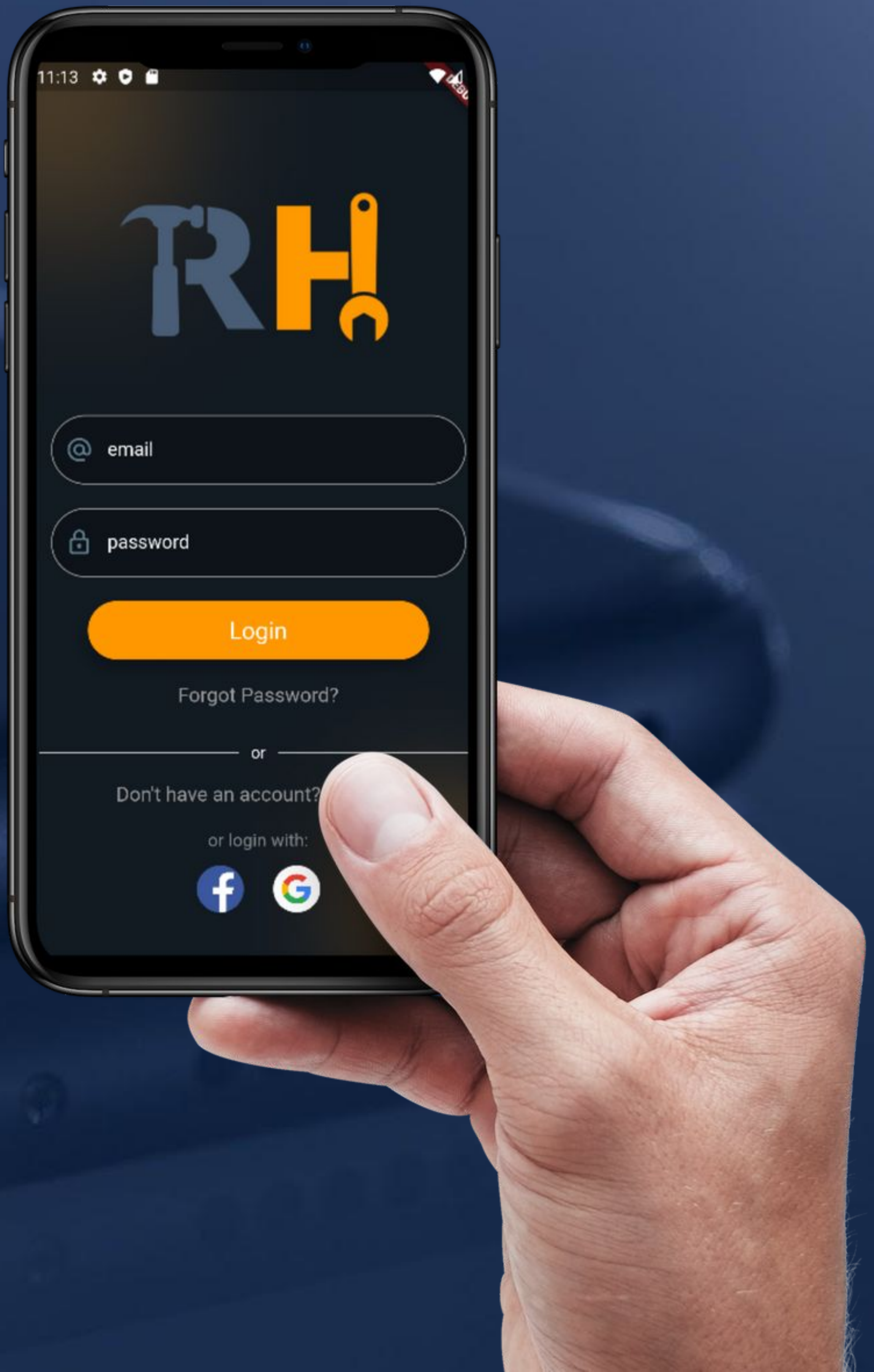
Cache Money

REVERSE HAND



Strive to make life easier by integrating **entire** process

ReverseHand

ReverseHand is a **mobile application** built with the **vision** of connecting local contractors and customers with a focus on **reducing the power imbalances** customers may face when seeking trade services. To achieve this, the mobile application enables customers to advertise their need for services through job postings where contractors can submit bids for selection and employment.

Contractor / Tradesman

Browse Jobs

Place Bid

Report client / job

Client / Customer / Consumer

Create Job

View Bids

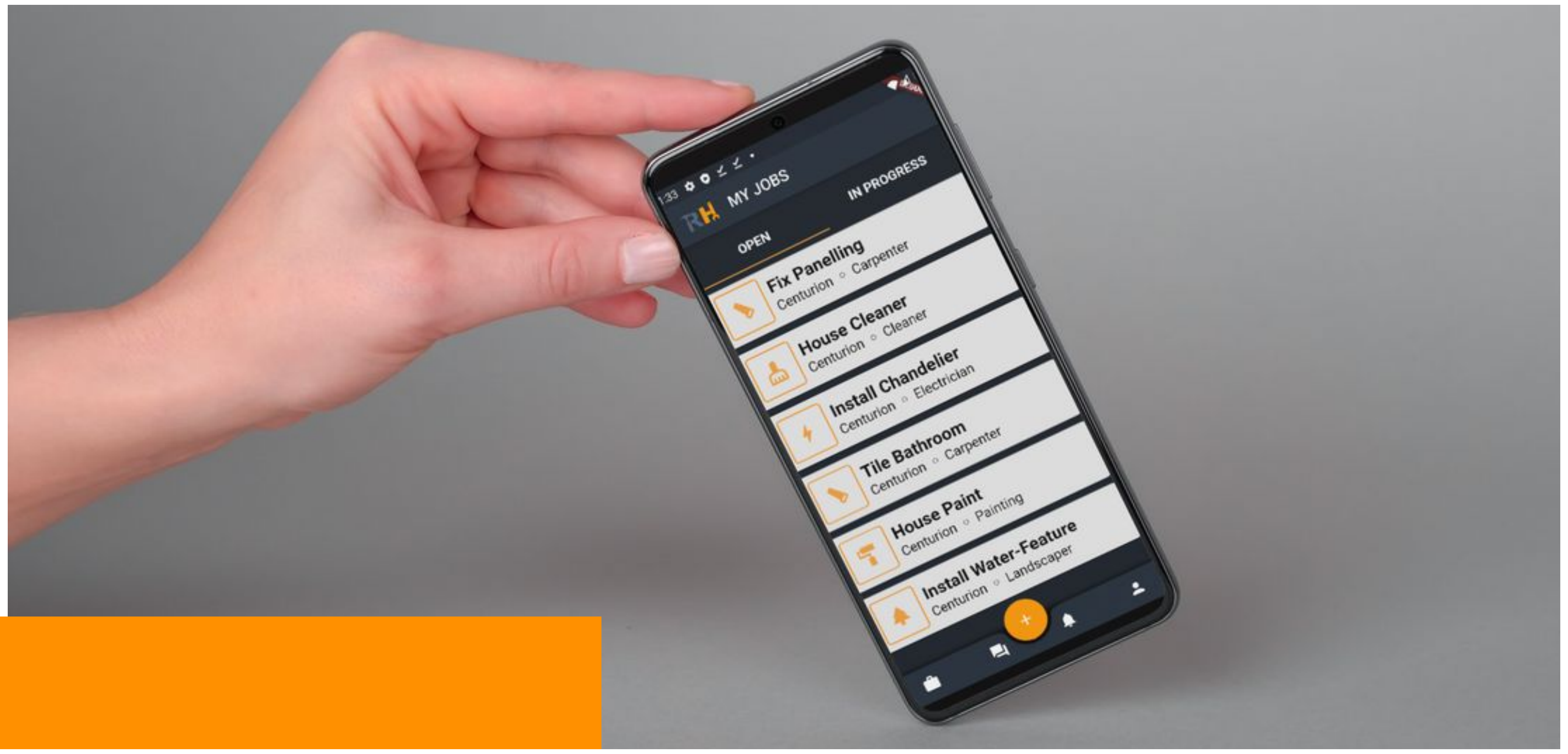
Favourite & Accept Bid/s

Pay Contractor

Admin

View User Metrics

View Reports from Users

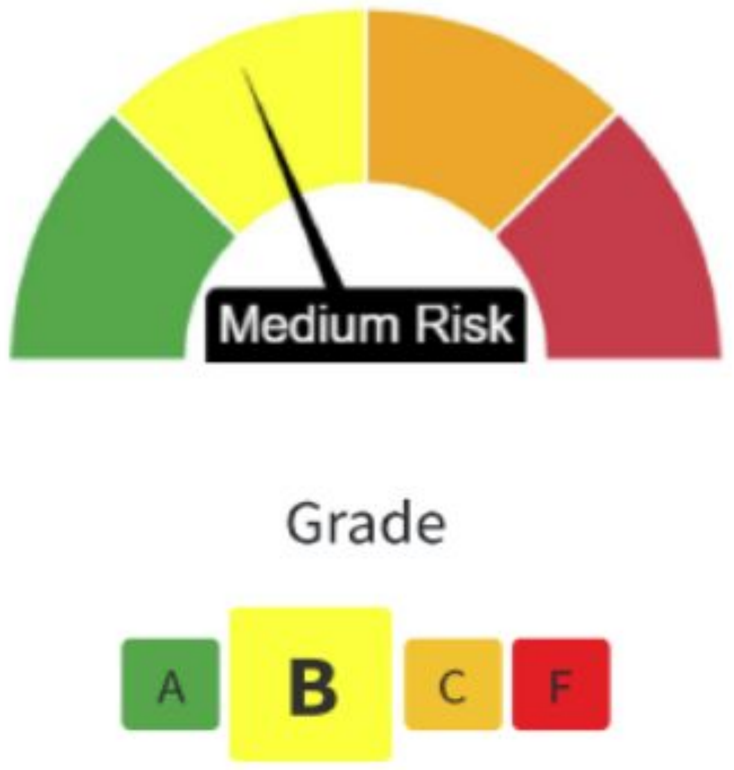


System Overview

Non-Functional Testing

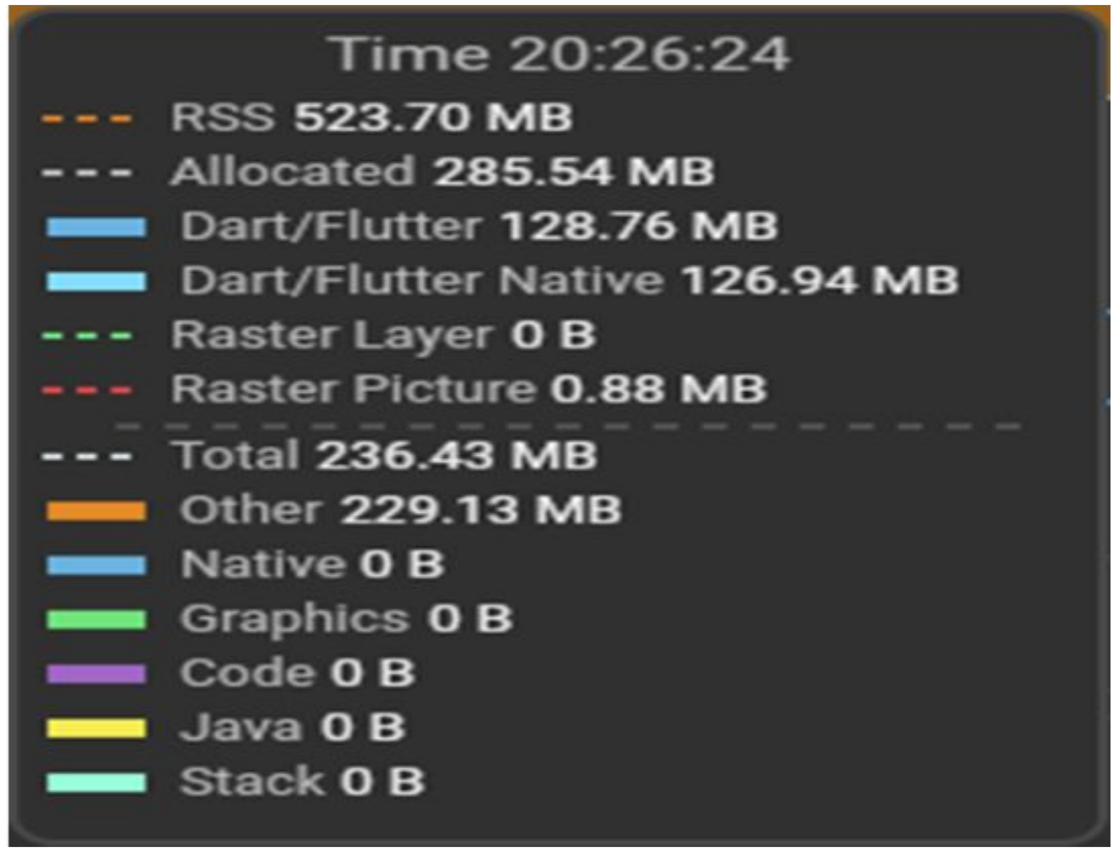
1. Security Testing

Static analysis performed using the Mobile Security Framework(MobSF). Application received a grade of B for security.



2. Performance Testing

Performance profiling and a form of load testing was performed on the mobile application. On average application used 230 MB of RAM when running.

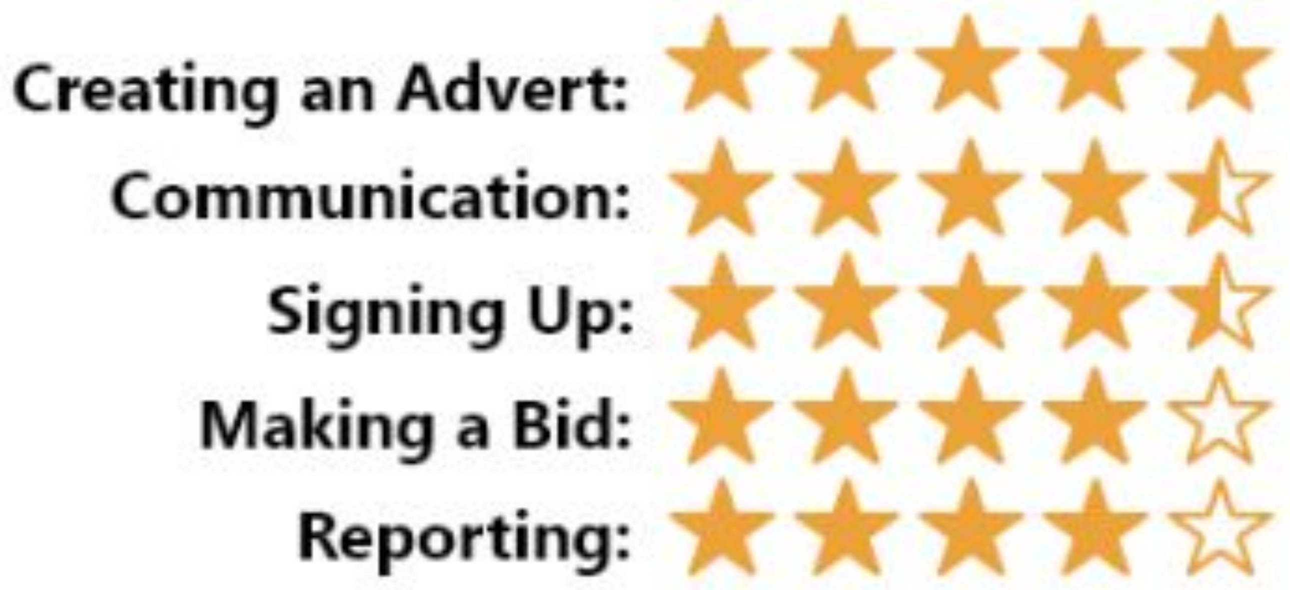


```
Time 20:26:24
RSS 523.70 MB
Allocated 285.54 MB
Dart/Flutter 128.76 MB
Dart/Flutter Native 126.94 MB
Raster Layer 0 B
Raster Picture 0.88 MB
Total 236.43 MB
Other 229.13 MB
Native 0 B
Graphics 0 B
Code 0 B
Java 0 B
Stack 0 B
```

Non-Functional Testing

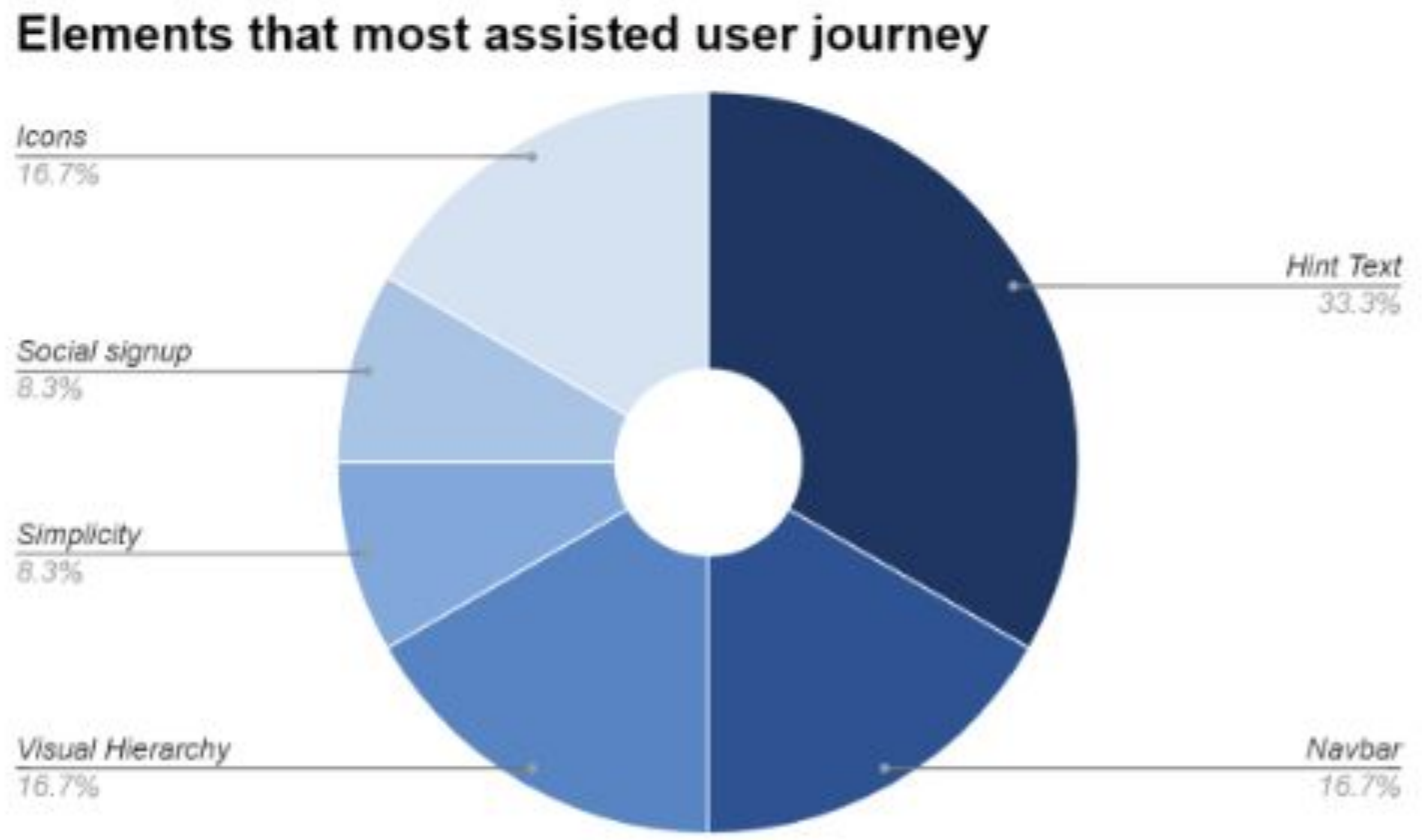
3. Usability Testing

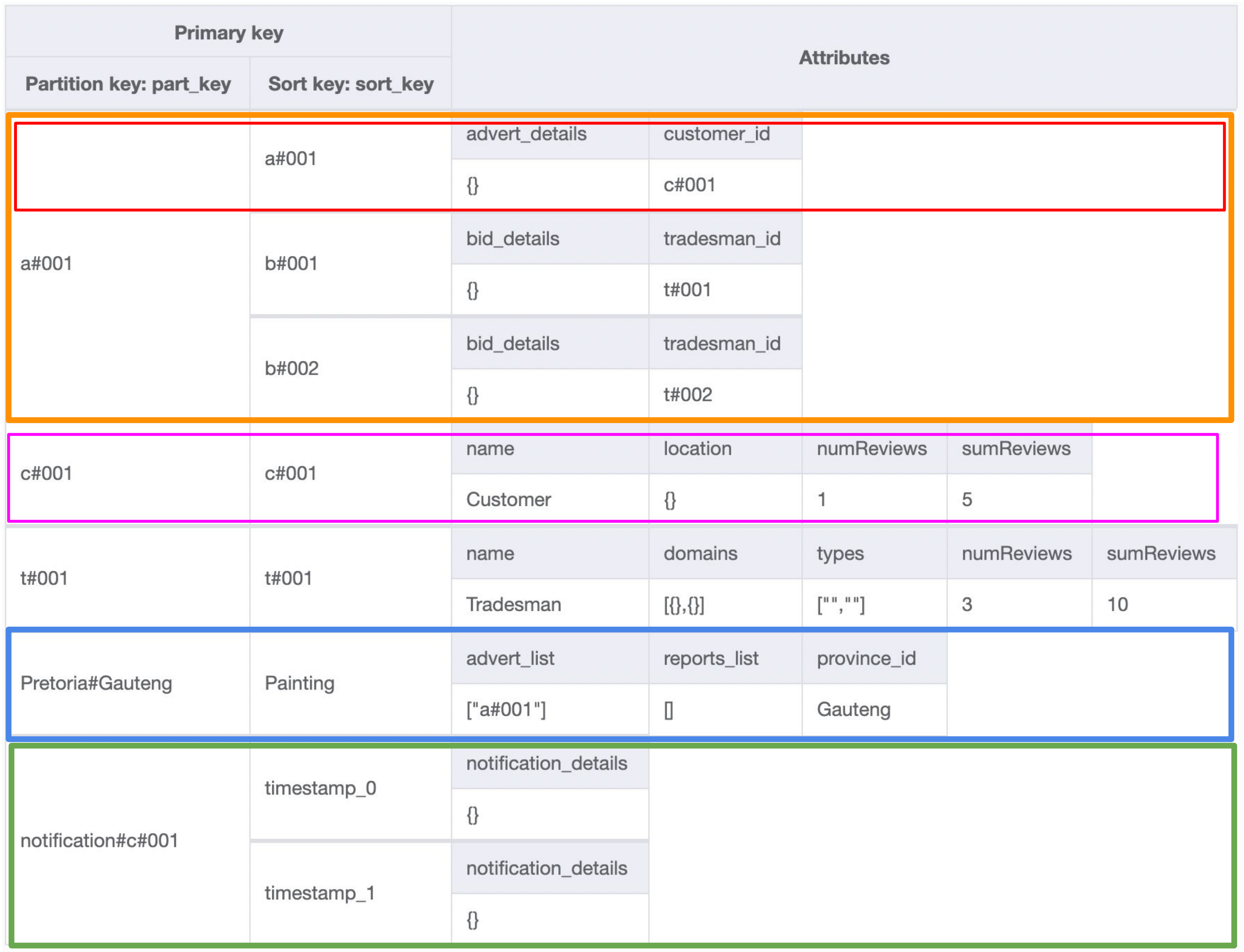
Ease of use:



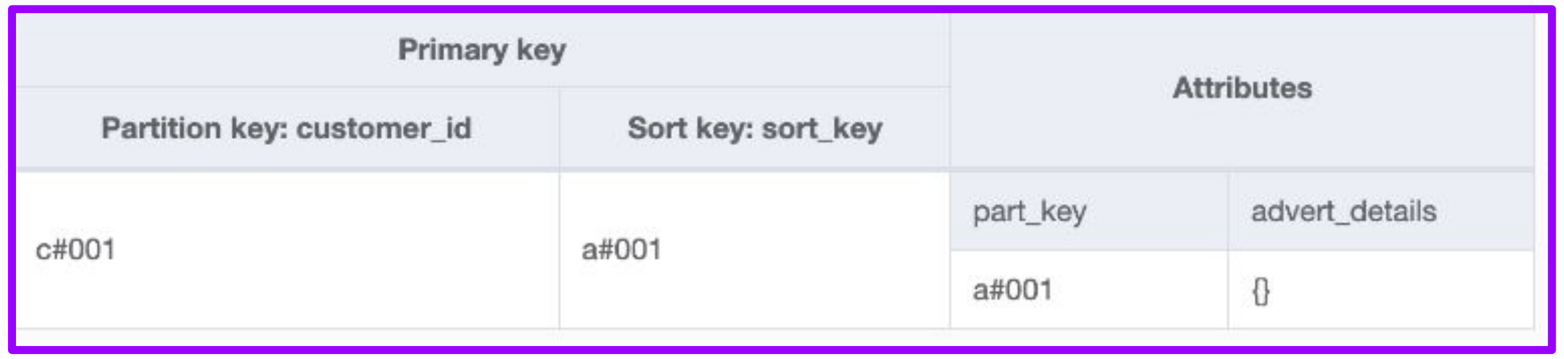
Creating an Advert:	★★★★★
Communication:	★★★★½
Signing Up:	★★★★½
Making a Bid:	★★★★☆
Reporting:	★★★★☆

Components that contribute to usability:





Primary key		Attributes				
Partition key: part_key	Sort key: sort_key					
	a#001	advert_details	customer_id			
		{}	c#001			
a#001	b#001	bid_details	tradesman_id			
		{}	t#001			
	b#002	bid_details	tradesman_id			
		{}	t#002			
c#001	c#001	name	location	numReviews	sumReviews	
		Customer	{}	1	5	
t#001	t#001	name	domains	types	numReviews	sumReviews
		Tradesman	[{},{}]	["",""]	3	10
Pretoria#Gauteng	Painting	advert_list	reports_list	province_id		
		["a#001"]	[]	Gauteng		
notification#c#001	timestamp_0	notification_details				
		{}				
	timestamp_1	notification_details				
		{}				



Primary key		Attributes	
Partition key: customer_id	Sort key: sort_key		
c#001	a#001	part_key	advert_details
		a#001	{}

Database

Type of Database

NoSQL Key-Value

Single Table Design

Benefits

Cost

Performance

Scalability

Maintainability

Terminology & Strategies

Composite PrimaryKey

Items

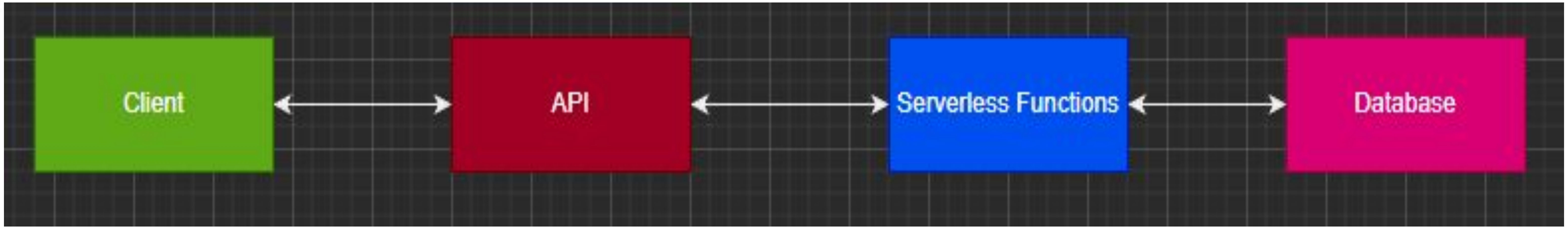
Secondary & Sparse Indexes

Architectural Styles

Microservices architectural style

Independently deployable services: allows resource intensive services to be scaled independently of the rest of system.
Highly Scalable
Can update an existing service without rebuilding and redeploying the entire system

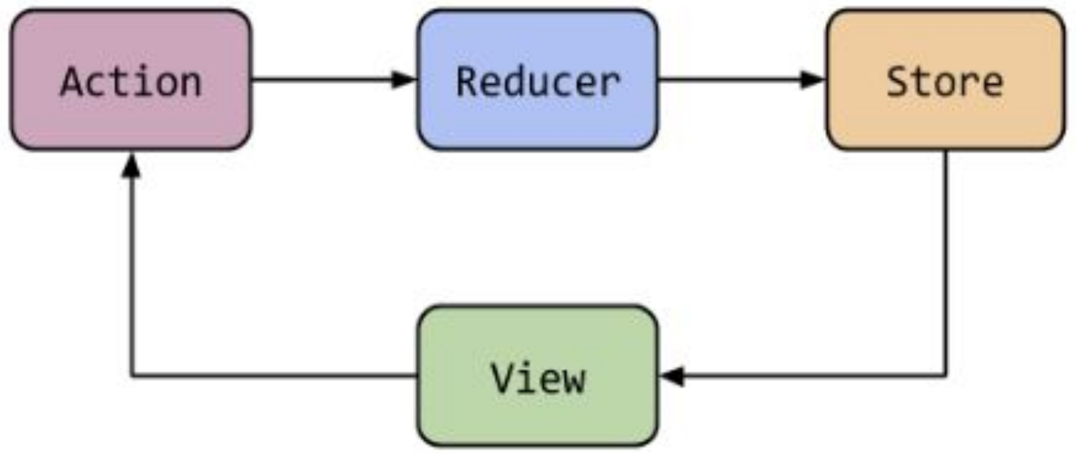
System events flow

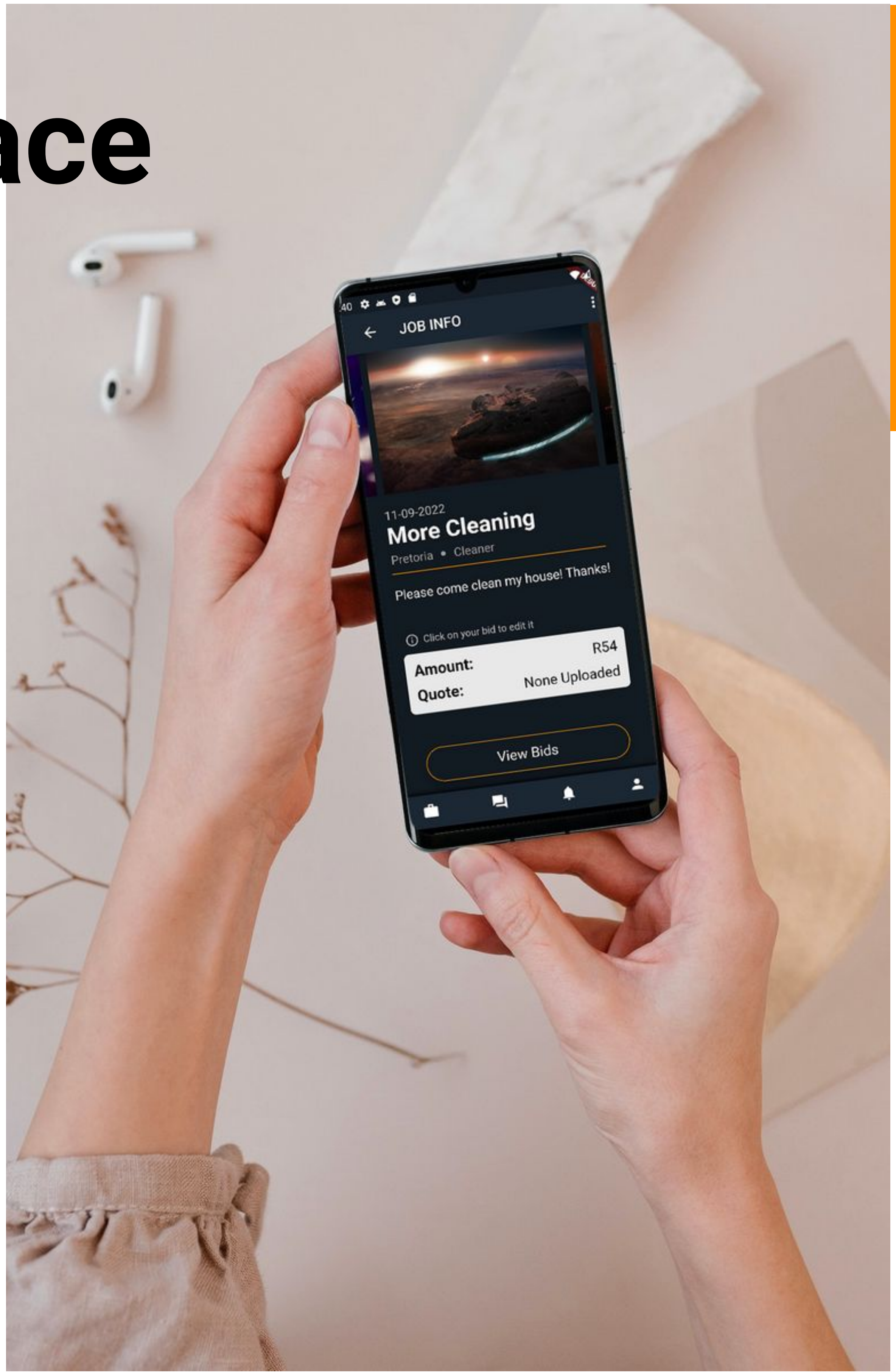


User Interface

Redux Design Pattern

State Management



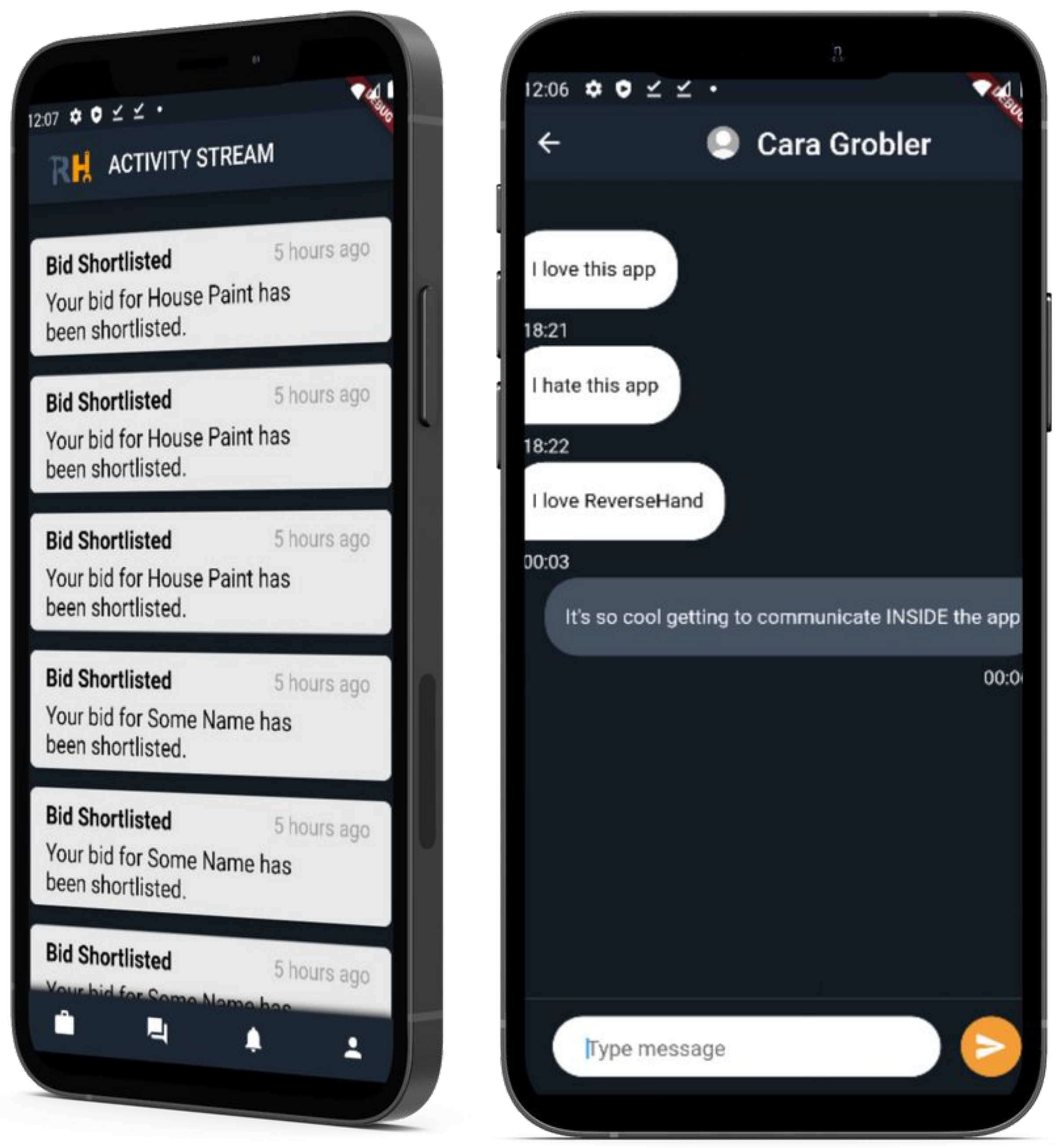


"What makes good UI?"

- Maintainable
- Adaptable
- Visibility of System Status

Notifications

We accomplish real time notifications and chat messages by making use of **websockets**


12:07

ACTIVITY STREAM

Bid Shortlisted 5 hours ago
Your bid for House Paint has been shortlisted.

Bid Shortlisted 5 hours ago
Your bid for House Paint has been shortlisted.

Bid Shortlisted 5 hours ago
Your bid for House Paint has been shortlisted.

Bid Shortlisted 5 hours ago
Your bid for Some Name has been shortlisted.

Bid Shortlisted 5 hours ago
Your bid for Some Name has been shortlisted.

Bid Shortlisted 5 hours ago
Your bid for Some Name has

12:06

Cara Grobler

I love this app
18:21
I hate this app
18:22
I love ReverseHand
00:03
It's so cool getting to communicate INSIDE the app
00:0

Type message

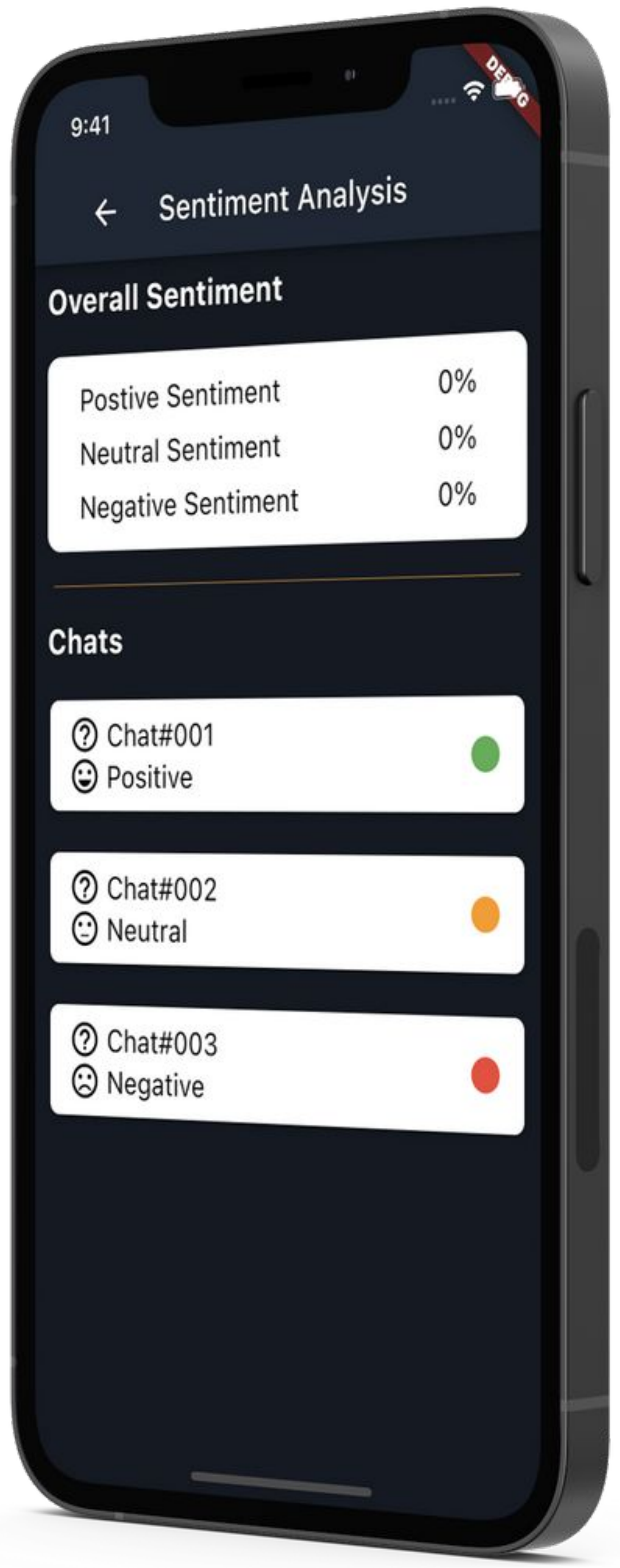
Chat App

Sentiment Analysis

Gather Feeling

AFINN-165 wordlist

Works with emojis


9:41

Sentiment Analysis

Overall Sentiment

Postive Sentiment	0%
Neutral Sentiment	0%
Negative Sentiment	0%

Chats

Chat#001
Positive ●

Chat#002
Neutral ●

Chat#003
Negative ●

Serverless Infrastructure

Our entire backend system is serverless. We interact with our database by making use of serverless functions.

The reason a serverless infrastructure was chosen is due to the ease at which it can be scaled since the application will need to be able to support many users concurrently.

Another important factor was cost and using a serverless infrastructure as well as implementing our own cost optimization techniques.

Cost Optimizations

When using a serverless infrastructure the two mains costs are data storage and execution time. Do to these facts the entire app lazy loads its data as to not pay for execution time that my result in no gains.

If there was a decision between execution time and storage space then storing a larger item was always chosen as execution time is more expensive than storage space.

When it comes to retrieving the user's inputed location we cost optimize the service by bundling requests with a session token as well as staggering request on every third character.

Payment System

Security

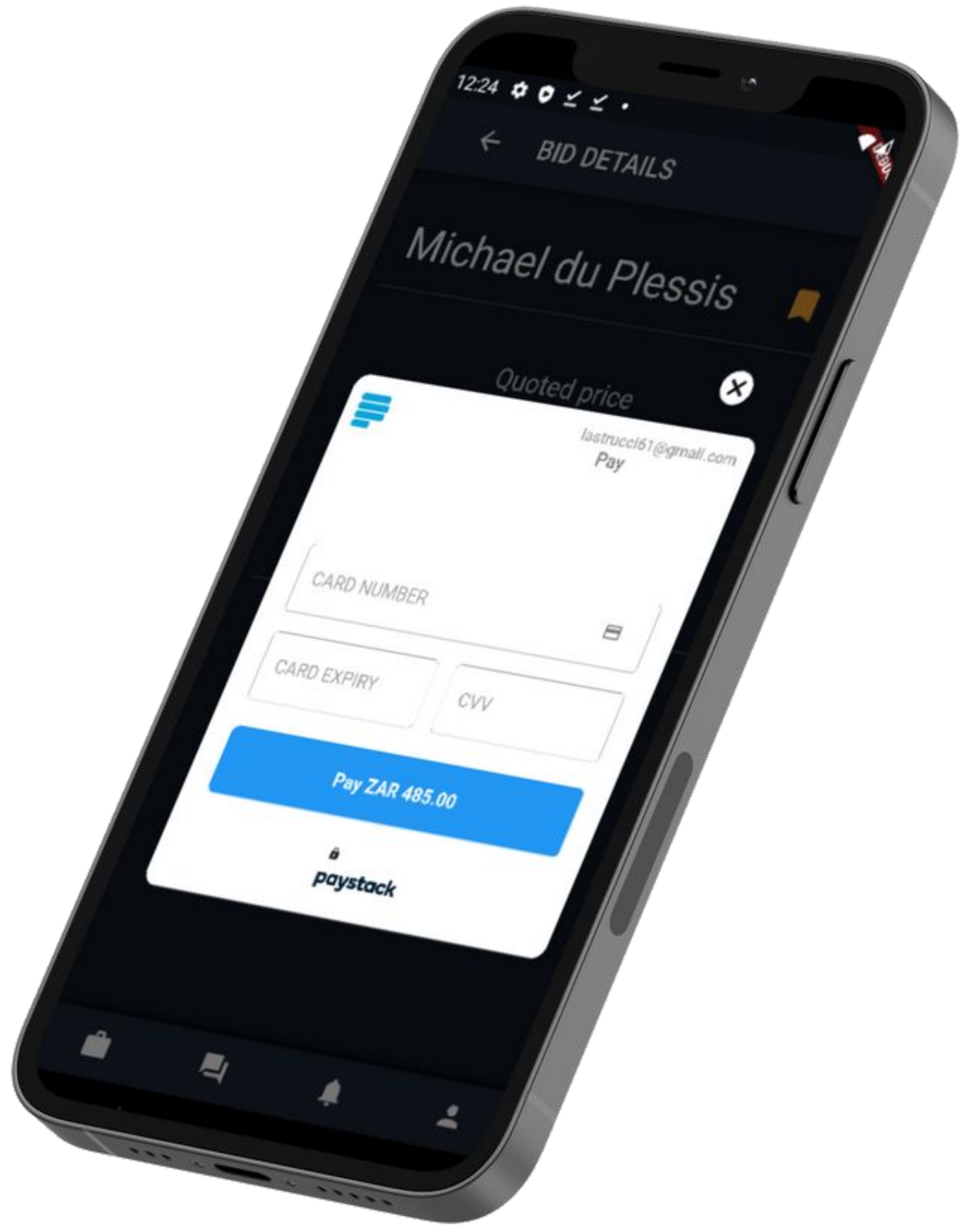
Advanced fraud detection
PCI DSS Level 1 certified

Convinience

Make payments at any time

Hassle Free

Pay instantly,
We hold the money, until the job gets closed



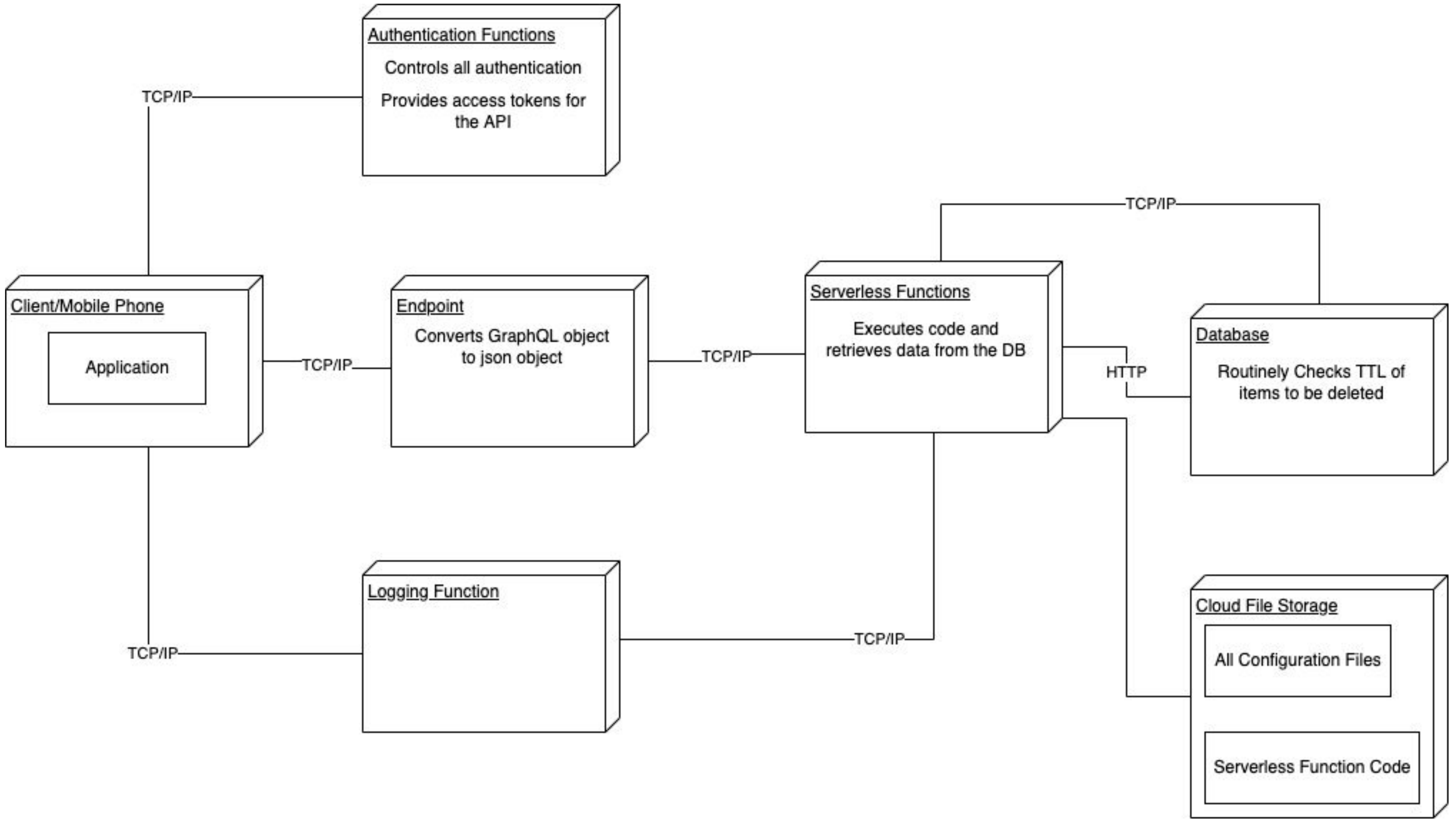
OAuth2

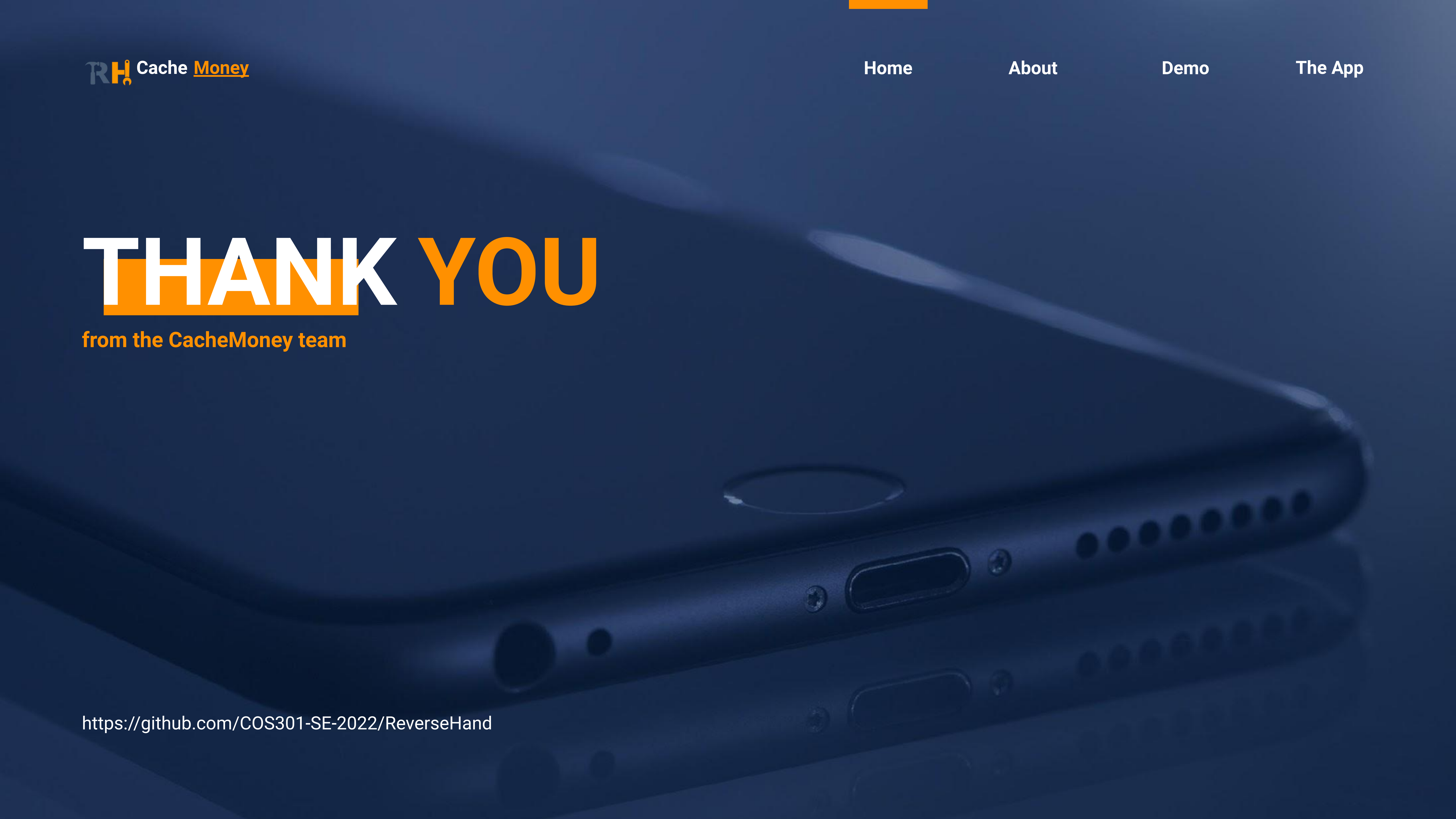
- Industry standard for Online Authorisation
- Delegated access
- Access & Identity

Logging

- In-app Custom events
- Logs are streamed and processed

Deployment Model





Cache Money

Home About Demo The App

THANK YOU

from the CacheMoney team

https://github.com/COS301-SE-2022/ReverseHand