

UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Freight trs: freight transportation management app (client flow)

Author:
Oleh MYKYTYN

Supervisor:
Serj MISKIV

*A thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Science*

in the

Department of Computer Sciences
Faculty of Applied Sciences



APPLIED
SCIENCES
FACULTY ●

Lviv 2021

Declaration of Authorship

I, Oleh MYKYTYN, declare that this thesis titled, "Freight trs: freight transportation management app (client flow)" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“The computer was born to solve problems that did not exist before.”

Bill Gates

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Freight trs: freight transportation management app (client flow)

by Oleh MYKYTYN

Abstract

Cargo transportation is a process as a result of which cargo is moved from one place to another by means of transport. It is an integral part of the export and import of any cargo type worldwide. Nowadays software is being developed for various areas to improve certain processes and also to simplify freight transportation.

Acknowledgements

I want to thank my parents who always support me, and also Oles Doboševych and Serj Miskiv who were always ready to give some advice.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Background information	1
1.2 Task definition	1
2 Existing solutions	2
2.1 Della	2
2.1.1 Overview	2
2.1.2 Pros and Cons	2
2.2 Uber Freight	3
2.3 Overview	3
2.3.1 Pros and Cons	3
3 Proposed approach	4
3.1 Background information	4
3.2 Architecture pattern - MVC	4
3.2.1 Overview	4
3.2.2 Pros and cons	5
3.2.3 Motivation	5
3.3 Navigation pattern - Coordinator	6
3.3.1 Overview	6
Pros and cons	6
3.3.2 Motivation	6
3.4 Application flows	7
3.4.1 Authorization flow	7
3.4.2 Main flow	7
Client side	7
Carrier side	8
4 Libraries and frameworks	9
4.1 Networking - Moya	9
4.1.1 Overview	9
4.1.2 Pros and cons	9
4.1.3 Motivation	9
4.2 Maps integration - Google Maps SDK, Google Places SDK	9
4.2.1 Overview	9
4.2.2 Pros and cons	10
4.2.3 Motivation	10

4.3	Keyboard handling - IQKeyboardManager	10
4.3.1	Overview	10
4.3.2	Pros and cons	10
4.3.3	Motivation	10
5	Backend - Django	11
5.1	Overview	11
5.1.1	Pros and cons	11
5.1.2	Motivation	11
6	Future plans	12
7	Summary	13
	Bibliography	14

List of Figures

2.1	Della website	2
2.2	Uber Freight iOS application	3
3.1	MVC pattern diagram	5
3.2	Coordinator pattern diagram	6
3.3	Authorization flow diagram	7
3.4	Client flow diagram	7
3.5	Carrier flow diagram	8

List of Tables

2.1	Pros and cons of Della website	2
2.2	Pros and cons of Uber Freight	3
3.1	Pros and cons of MVC	5
3.2	Pros and cons of Coordinator pattern	6
4.1	Pros and cons of Moya	9
4.2	Pros and cons of Google Maps SDK	10
4.3	Pros and cons of IQKeyboardManager	10
5.1	Pros and cons of Django	11

List of Abbreviations

MVC	Model View Controller
MVVM	Model View ViewModel
SDK	Software Development Kit
OS	Operating System
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
UI	User Interface
UX	User Experience

Dedicated to my beloved parents...

Chapter 1

Introduction

1.1 Background information

When I was young I was fond of cars and in particular big trucks. The interest in this area has not disappeared until now. I try to communicate with my acquaintances who work in this field, I always ask them about different aspects of one, so I realized that I want to create something what can really help in freight transportation area.

1.2 Task definition

Application for iOS platform with the following requirements:

- User-friendly
- Facilitating document management flow
- Handy communication between user types
- Order tracking functionality
- Finding the best way for driver to minimize the trip with unloaded truck trailer

Chapter 2

Existing solutions

2.1 Della

2.1.1 Overview

Della.ua is a popular website among the people working in the field of freight transportations. Della provide some base functionality like posting and accepting different transportation orders. It's just something like bulletin board.

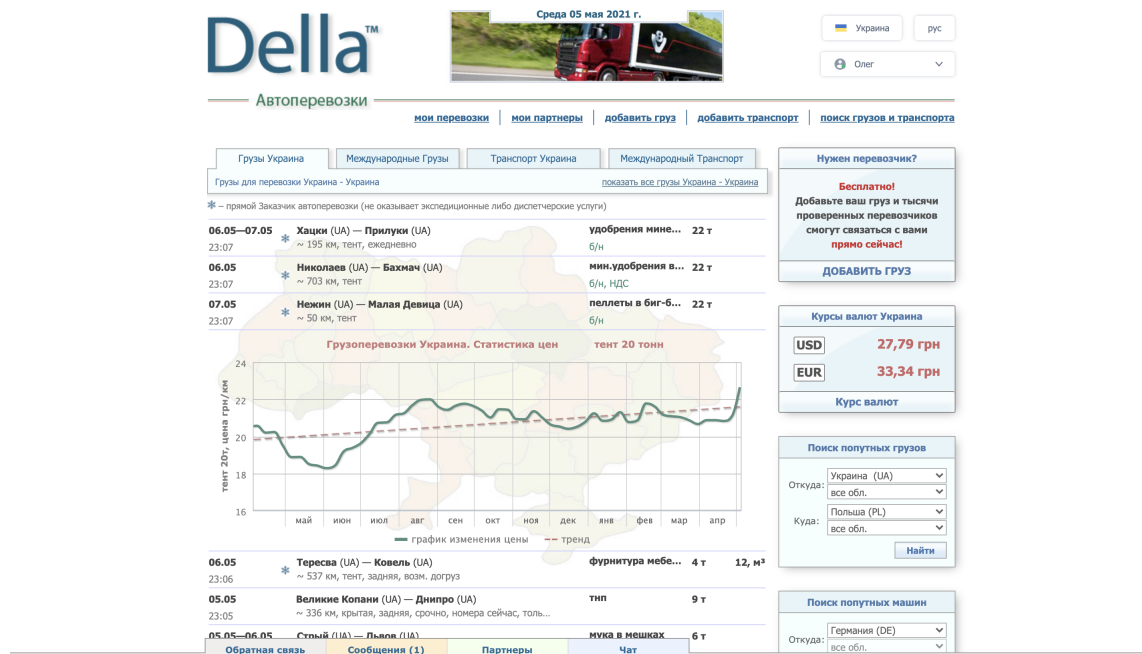


FIGURE 2.1: Della website

2.1.2 Pros and Cons

Pros	Cons
Simple in usage.	No order tracking. It's just a website(no mobile application).

TABLE 2.1: Pros and cons of Della website

2.2 Uber Freight

2.3 Overview

Uber Freight is an app that matches carriers with shippers. Actually, the rating in AppStore is not high - 3.7 out of 5. I analyzed what reviewers say and figured out that the worst parts of application are payments handling and control of relations flow between carriers and shippers. *UberFreight on AppStore, UberFreight*

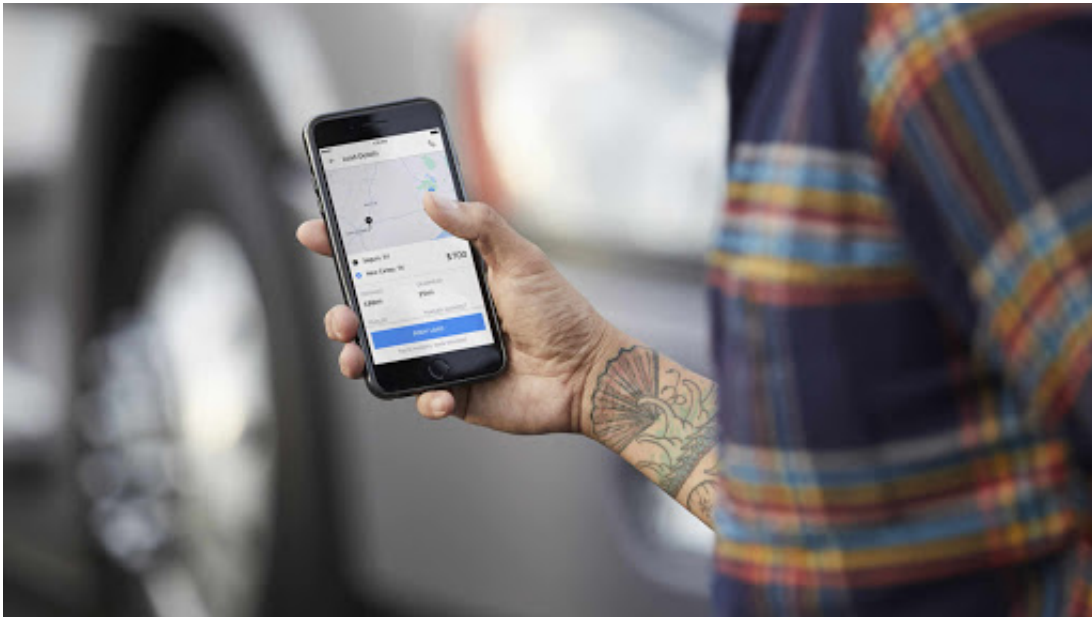


FIGURE 2.2: Uber Freight iOS application

2.3.1 Pros and Cons

Pros	Cons
Modern user interface.	Unavailable for the Ukrainian market.
Intuitive navigation.	Inconvenient process of interaction between carriers and shippers.
	Inconvenient payment process.

TABLE 2.2: Pros and cons of Uber Freight

Chapter 3

Proposed approach

3.1 Background information

Usually, it is hard for transportation companies to find an optimal order in terms of the ratio of the distance traveled by the truck with the cargo and without it. According to statistics, trucks travel 40% of the total journey at idle. For example, in western Ukraine there are many orders for the export of grain products to the ports of Odessa, so the truck goes to the destination place with the cargo, but in the opposite direction it usually goes without it, because it is difficult to find a suitable order on the way back. Because of such a situation the company's profit is lower than it could be and also this produces useless emissions of exhaust gases into the atmosphere. I decided to create a freight transportation management application with a friend of mine, who is also fond of trucks. So, we planned our tasks and I have to concentrate my work on creating a part of application which is connected to the Client side of application(one of user type).

3.2 Architecture pattern - MVC

3.2.1 Overview

MVC is Apple's preferred way for developing applications for its platforms. MVC is a software development pattern based on three components:

1. **Model** is where your data is stored. Things like persistence, object models, managers etc.
2. **View** is a layer which displays the data from Model for users.
3. **Controller** is something like mediator between the view and the model. It converts the data to such format that View can display.

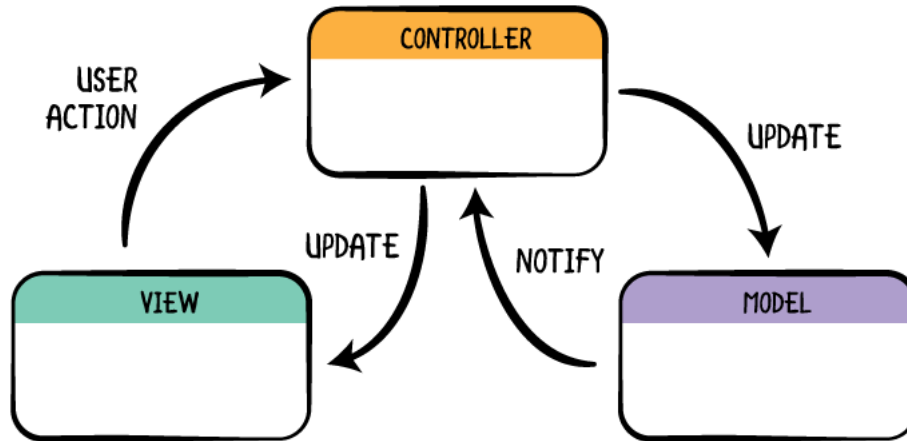


FIGURE 3.1: MVC pattern diagram

Actually, on iOS View Controllers uncomfortably combine the View and Controller, because they are usually used as Controllers.

3.2.2 Pros and cons

Pros	Cons
Very simple design.	"Massive View Controller" - because of combination of the View and the Controller into the ViewController the MVC pattern is sometimes called "Massive View Controller". View Controllers classes become to big and unreadable. So it is a bad choice for projects with complex logic.
Everyone knows this pattern, if not then it can be explained very quickly.	
It's straightforward, if you want to code something - you are just doing it without any additional complexities like wrappers, abstractions, mediation services etc.	
Can be easily rewritten to MVVM.	

TABLE 3.1: Pros and cons of MVC

3.2.3 Motivation

As I have already mentioned above, this approach is straightforward, simple and most importantly it's quick implementable one. These points were the most important for me during working on my graduation work.

3.3 Navigation pattern - Coordinator

3.3.1 Overview

The main point is that Coordinator pattern allows us to adjust the application flow like you wish. This pattern helps us to remove the navigation logic from the view controllers and, in turn, makes controllers more reusable and easy manageable.

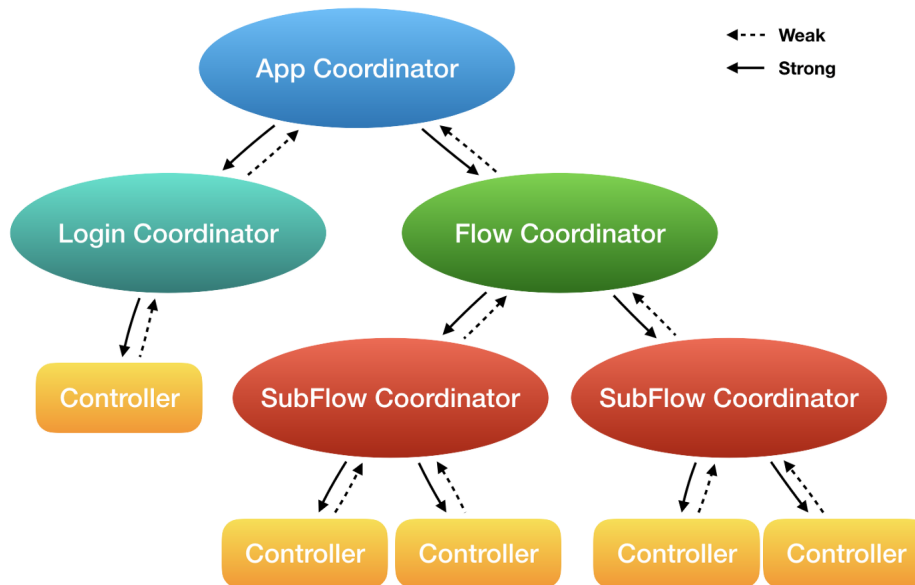


FIGURE 3.2: Coordinator pattern diagram

Pros and cons

Pros	Cons
Removes navigation logic from view controllers.	Easy to occur memory leaks.
Makes view controllers more reusable.	
Allows to perform different transition animations and keeps it as separate logic.	

TABLE 3.2: Pros and cons of Coordinator pattern

3.3.2 Motivation

I have some experience working on projects with different navigation setups and I understood that the best approach for me is exactly Coordinator pattern. It provides a great project scalability, besides this point it's very easy to understand and implement.

3.4 Application flows

3.4.1 Authorization flow

The application provides authorization via phone number. It's pretty simple and straightforward:

1. User enters a phone number.
2. If a specific number is already registered in the database, then user will be redirected to the main flow, but if one is not registered yet, then user will be redirected to the registration flow.

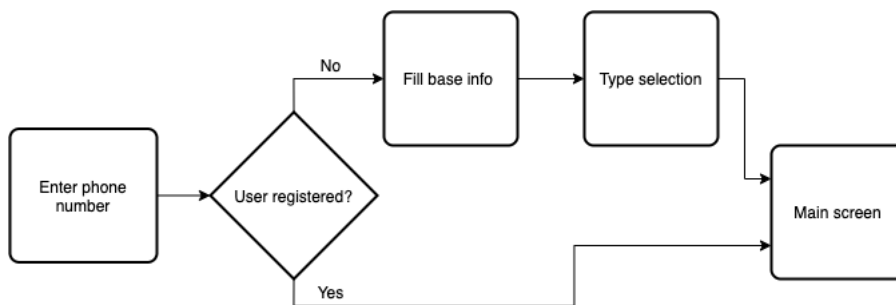


FIGURE 3.3: Authorization flow diagram

3.4.2 Main flow

There are 2 sides:

1. Client side - responsible for creating orders.
2. Carrier side - responsible for transportation.

Client side

If the client wants the order to be transported from point A to point B he should create an order in the application and pay it. The funds will be frozen until the order is delivered to its destination place, then the funds will be transferred to the Carrier. The application provides the following basic features for the Client: order creating, secure order payment and real time order tracking.

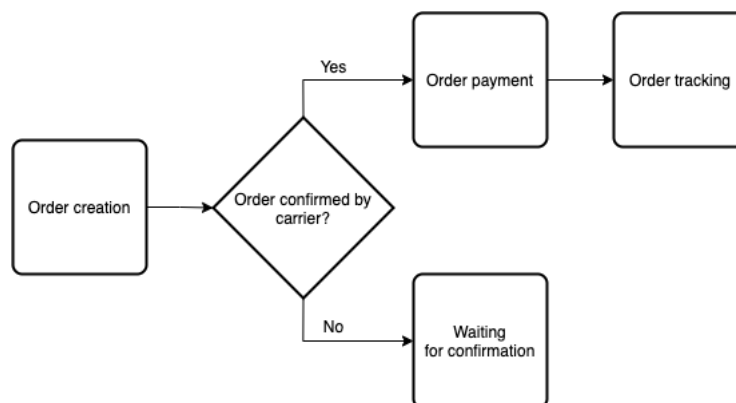


FIGURE 3.4: Client flow diagram

Carrier side

Carrier is divided into 2 user types: Manager and Driver. Manager is a type of user who can monitor all available orders, confirm an order and assign to the available driver. Driver is a type of user who receives an order from the Manager, then he receives a road to the loading and unloading point, and fulfills the order. The application provides the following basic features for the Carrier: Managers can search through the list of orders, sort them by different filters(cargo weight, loading place, unloading place etc), assign order to driver, track driver's progress in real time; Drivers can accept/reject order, use navigation to loading and unloading place provided by application, read comments to the order.

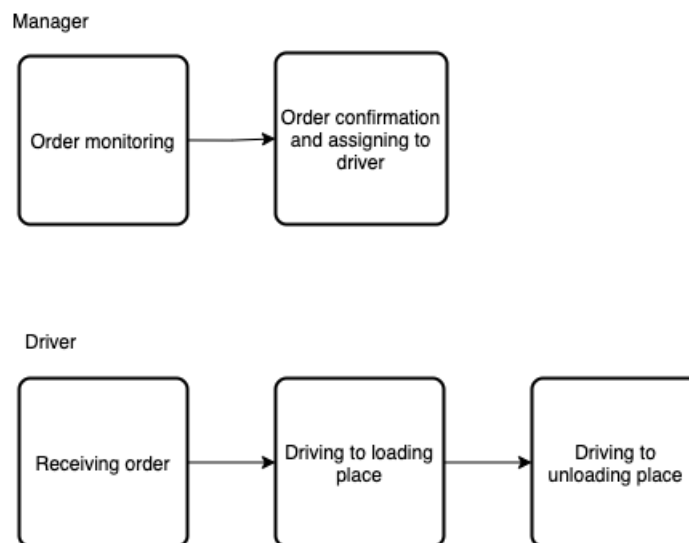


FIGURE 3.5: Carrier flow diagram

Chapter 4

Libraries and frameworks

4.1 Networking - Moya

4.1.1 Overview

Moya is a network abstraction layer. It uses Alamofire(very popular Swift-based, HTTP networking library) and provides us to set up a network manager quickly.

4.1.2 Pros and cons

Pros	Cons
Simple in usage.	Requires installation.
Helps to structurize and divide network requests.	
Covers many of the difficulties of making networking calls.	

TABLE 4.1: Pros and cons of Moya

4.1.3 Motivation

Moya is a wrapper over the Alamofire library, Alamofire is a wrapper over the URLSession(native iOS class for network data-transfer tasks). Since Moya is a wrapper over two services, it hides many of tricky moments, making you free to concentrate on app's business logic, what leads to clean and readable code. As the project is big enough and uses many API calls, Moya will be a great helper to integrate with.

Moya, Moya Setup

4.2 Maps integration - Google Maps SDK, Google Places SDK

4.2.1 Overview

Maps SDK for iOS helps to include map based logic into your application. This SDK automatically handles access to the Google Maps servers, map display, and response to user gestures such as clicks and swipes. There is also an ability to add different pins, polylines, overlays etc to your map.

4.2.2 Pros and cons

Pros	Cons
Simple in usage.	Requires installation and registration.
Provides a lot of data(place number, images, website, info, working hours...).	SDK is Paid.

TABLE 4.2: Pros and cons of Google Maps SDK

4.2.3 Motivation

Since it was planned to create also an Android version of such application, Google Maps SDK is a must for this OS. Combining this fact and the fact that I have a great experience in working with Google Maps SDK in the past, it was decided to use this particular SDK. As I said before, Google's SDK provides different place data(number, images, website...) what will be most likely helpful in the future. The native iOS MapKit framework is very similar to the Google's one, but it doesn't provide such extended information. *Mapkit vs Google Maps, Google Maps SDK*

4.3 Keyboard handling - IQKeyboardManager

4.3.1 Overview

IQKeyboardManager allows you to prevent the issue of keyboard covering text views and text fields without any code and additional setup.

4.3.2 Pros and cons

Pros	Cons
Works Automatically.	Requires installation.
No manual work and setup.	
Works on all orientations.	
Supports toolabar.	

TABLE 4.3: Pros and cons of IQKeyboardManager

4.3.3 Motivation

The main thing is that this manager saves a lot of time by saving the developer from writing additional code for keyboard handling. If you want to handle something like keyboard on chat screen then you can just disable IQKeyboardManager on that screen and handle keyboard manually. I have some practice with this manager in many projects, so since this one has a lot of input fields, it's a great one to integrate with IQKeyboardManager. *IQKeyboardManager*

Chapter 5

Backend - Django

5.1 Overview

Django is free open source high-level Python Web framework that provides opportunity to create well-structured and clean-code applications.

5.1.1 Pros and cons

Pros	Cons
Fast.	Not suitable for small projects.
Secure.	Monolithic.
Great scalability.	
Easy syntax because of Python.	

TABLE 5.1: Pros and cons of Django

For the database PostgreSQL is used. PostgreSQL is the world's most advanced open source relational database with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. [Postgresql](#)

For the server side Heroku is used. Heroku is a cloud platform as a service (PaaS) supporting several programming languages including Python. [Heroku Wiki](#), [Heroku](#)

5.1.2 Motivation

Since the application is planned to grow and expand quickly, Django can be a great candidate for the backend side. Thanks to Django's monolithic architecture and lot's of important functionality initially bundled with a framework the application can have a really quick start.

Chapter 6

Future plans

1. Finish remaining features.
2. Testing.
3. Release to AppStore.
4. Fixing bugs according to users feedback.
5. Implementing chat between Users.(sockets)
6. UI and UX redesign.
7. Android application development.
8. Migration from regular UIKit to SwiftUI(new declarative Swift syntax framework), migration from MVC pattern to MVVM.

Chapter 7

Summary

The idea to create such iOS application was born about a year ago, but due to studying and working I didn't have enough time to start implementing the project.

Now I'm adapted to work and study simultaneously, and the thought of graduate work gave me enough motivation to start working on "Freight trs".

In this work was described iOS application for freight transportation management "Freight trs", whose aim is to make lives of people working in the field of freight easier. In general, all my expectations were justified, but a few moments still remain not implemented, but I'm highly motivated to continue working hard, so it would be finished soon.

Bibliography

Google Maps SDK. URL: <https://developers.google.com/maps/documentation/ios-sdk/overview>.

Heroku. URL: <https://www.heroku.com/>.

Heroku Wiki. URL: <https://en.wikipedia.org/wiki/Heroku>.

IQKeyboardManager. URL: <https://github.com/hackiftekhar/IQKeyboardManager>.

Mapkit vs Google Maps. URL: <https://medium.com/swlh/mapkit-vs-googleplace-c32f9413d1fb#:~:text=MapKit%20is%20the%20integrated%20Apple,the%20map%20displays%20or%20interactions..>

Moya. URL: <https://github.com/Moya/Moya>.

Moya Setup. URL: <https://medium.com/trendyol-tech/get-instantly-a-network-module-using-moya-in-swift-5f918ecdfb58>.

Postgresql. URL: <https://www.postgresql.org/>.

UberFreight. URL: <https://www.uber.com/us/en/freight/>.

UberFreight on AppStore. URL: <https://apps.apple.com/us/app/uber-freight/id1183931851>.