

CALIFORNIA POLYTECHNIC STATE
UNIVERSITY, SAN LUIS OBISPO

CSC SENIOR PROJECT

FORMAL REPORT

HitchRide iOS App

Abstract

HitchRide is a ride-sharing marketplace connecting regular people making similar trips so everyone benefits: the driver, the passenger, society and the environment. HitchRide allows participants to reduce their travel costs, meet new friends and conveniently reach their destination, all in a simple, satisfactory, and environmentally friendly way. We believe that we can make traveling truly enjoyable, one shared car ride at a time and help save the planet while we're at it!

Author:
Petar GEORGIEV

Supervisor:
Dr. John BELLARDO

March 22, 2017

Contents

1	Introduction	1
2	Background	2
2.1	Previous Work	2
2.2	Related Work	2
3	Key Design Decisions	3
3.1	Native vs Hybrid Apps	3
3.1.1	What is a Native App?	3
3.1.2	What is a Hybrid App?	4
3.1.3	What to Consider Before Making a Decision?	4
3.1.4	HitchRide - Native or Hybrid?	4
3.2	Parse vs Firebase vs Custom Back-end	5
3.2.1	Why Parse May Not Be the Best Option?	5
3.2.2	Use Firebase as an Alternative?	5
3.2.3	Write Custom Back-end Instead?	6
3.2.3.1	Django	6
3.2.3.2	Ruby on Rails	7
3.2.4	What Back-end is HitchRide using?	7
3.3	Basic Authentication vs Session Authentication	8
3.3.1	What is Basic Auth?	8
3.3.2	What is Session Auth?	8
3.3.3	How Does HitchRide Authenticate its Users?	9
4	Results	9
4.1	Find a Ride	10
4.2	Posting a Ride	11
4.3	User Profile	13
5	Future Work	13
6	Reflections	14
7	Conclusion	14

1 Introduction

Nowadays, traveling has become an essential part of our life. People have many needs and traveling has become one of the most important ones [8]. Traveling could be business trips, vacations, family trips or a week out planned with a loved one. There are people who only travel for business. For adventurers, those that have a natural curiosity to explore a new place, traveling is more than a hobby, it is a lifestyle. There are also people who simply enjoy traveling or want to spend their vacation in their favorite place.

Of course, there are a range of different types of transport available to us; the type we choose to use will depend on the purpose of traveling, length of the journey and our relationship to the people who may travel with us. Increasingly, cost is becoming a deciding factor for the type of transport we use, especially due to the increase in fuel costs we all wish to avoid [12].

Cars are the most popular type of transport [12]. People use cars daily for short and long journeys. This type of transport has many advantages such as their ease of use and convenience, getting you exactly where you want to be without additional buses, trains or walking needed to get to your final destination.

Rail transport is used for both goods and people and is a popular method of public transport. Many towns have a train station but there are still a lot more that don't have one, which would make traveling by train difficult to get to some destinations and will require further transportation by bus, car or taxi.

The last of the top three most popular methods of transportation are buses. They are available in all cities, towns and most villages. They are ideal for those short journeys from one part of a town to another but not as convenient for longer trips.

HitchRide aims to decrease costs of travel, while connecting you to people traveling to similar locations. It does this by providing users a ride-sharing platform. Anyone can use the platform to reduce their travel costs and find companionship. In comparison to other ride-sharing applications, HitchRide's users have to first create a profile by signing up into the system. Drivers then simply post the details of their trip and await a notification that a passenger wants to join the driver on their journey. HitchRide also allows both drivers and passengers to rate and review their experience with each other on their respective profiles. This helps prospective drivers and passen-

gers to decide whether they want to join another person’s journey based on other people’s experiences.

2 Background

2.1 Previous Work

There are a large number of ride-sharing platforms that anyone with an Internet connection can access easily. These types of applications, however, are not 100% relevant to the problem domain discussed in the previous section because they mostly focus on short distance journeys.

Many other applications provide a ride-sharing marketplace for specific users only. For example, the PolyRides application available on both Google Play and Apple App Store targets California Polytechnic State University, San Luis Obispo students as their primary users [10]. However, by targeting students from a specific university, PolyRides only solves the problems of lowering the traveling cost and expanding their users’ social network for just a small amount of people. Of course, PolyRides’ decision of targeting a specific community adds a certain level of trust and security for their users but there is more than one university in California, not to mention the whole world.

The application that comes the closest to lowering the cost of traveling and expanding a person’s social network into a larger group of people is Zimride, only available on the Apple App Store [17]. This application provides a secure ride-sharing platform for companies and universities. It doesn’t fully solve the problem of expanding people’s social network because Zimride’s users are only limited to communicate with their peers. They securely log into the private network for their corporation or university, say where they need to go, how they would like to get there and get matched with a ride or a rider in their network.

2.2 Related Work

HitchRide’s web-based application is built using Ruby on Rails. Rails is a model–view–controller (MVC) framework, providing default structures for a database, a web service and web pages, written in the Ruby language [11]. HitchRide uses MySQL, an open-source relational database management sys-

tem, Devise, a flexible authentication solution for Rails based on Warden [9] and Devise-Token-Auth, a library that provides token based authentication functionality for Rails JSON APIs. Other libraries used for HitchRide’s web application are Bootstrap, Twitter-Typeahead, Will-Paginate, Twillio, Paperclip, Geocoder and Lets-Encrypt.

HitchRide’s mobile application is built using Swift 3, a new programming language for iOS, macOS, watchOS, and tvOS apps. Swift was used to build the front-end of the mobile application while the back-end is the website’s REST-based APIs. The mobile application uses SwiftyJSON, an external library that makes it easy to deal with JSON data in Swift [13].

3 Key Design Decisions

3.1 Native vs Hybrid Apps

There are hundreds of articles detailing and debating the Native vs. Hybrid topic. Some argue that the war between the two sides is already over and that most apps are already hybrid [2]. Others take a more balanced approach, assessing the weaknesses, opportunities and threats of each strategy [4].

3.1.1 What is a Native App?

A native app is a smartphone application developed specifically for a mobile operating system [1]. Since the app is developed within a mature ecosystem following the technical and user experience guidelines of the OS (e.g. swipes, app defined gestures, left aligned header on Android, centrally aligned header on iOS, etc), it not only has the advantage of faster performance but also “feels right” [1]. What feeling right means is that the in-app interaction has a look and feel consistent with most of the other native apps on the device. The end user is thus more likely to learn how to navigate and use the app faster. Finally, native applications have the significant advantage of being able to easily access and utilize the built-in capabilities of the user’s device (e.g. GPS, address book, camera, etc). When a user sends text messages, takes pictures using the device’s default app, set reminders, or uses the device’s music app (the one that came with the phone), they’re using native apps.

3.1.2 What is a Hybrid App?

Hybrid applications are, at core, websites packaged into a native wrapper [1]. They look and feel like a native app, but ultimately outside of the basic frame of the application they are fueled by the company's website. Basically, a hybrid app is a web app built using HTML5 and JavaScript, wrapped in a native container which loads most of the information on the page as the user navigates through the application [1]. Usual suspects here are Facebook, Twitter, Instagram, mobile banking apps and many others.

3.1.3 What to Consider Before Making a Decision?

Typically, when a company decides to build a mobile app, they are either playing a catch up game with their competitors, or have identified a business opportunity previously untapped [1]. Whatever the reason, executives want the application built out and released ASAP. However, as most people know, ASAP often means that many compromises need to be made as well as that most decisions are made on the fly. Both hybrid and native approaches can get the job done but there are certain considerations that should be understood before any important decisions are made.

If a company can wait six months or more before the app is launched, a native approach makes the most sense. Native applications have the best performance, highest security, and best user experience.

However, if the desired time to market is less than six months, then hybrid could be a better alternative. The app can be built in one source code, be released across platforms, and development time and effort is considerably less compared to that of native applications.

As mobile app usage takes over traditional browsing paths (desktop/laptop), companies with a mobile presence must constantly think about the two key promoters or detractors of their application: speed and user experience [1]. Even though the performance of the app as well as the user experience vary significantly based on the development framework chosen, the native app approach is considered to be the uncontested winner in both cases.

3.1.4 HitchRide - Native or Hybrid?

HitchRide already has a fully built website, which would have made it very easy to build a hybrid mobile app. However, as mentioned earlier,

the three key factors when deciding whether to go native or hybrid are time, performance and user experience. Fortunately, in the case of HitchRide, there were more than 6 months to develop the app. Because of that, HitchRide mobile app was built natively for iOS devices. This means that as users navigate through the mobile app, the contents, structure, and visual elements are already on their phone, available for instant loading, which provides seamless user experience by enhancing performance.

3.2 Parse vs Firebase vs Custom Back-end

Modern mobile services require authentication to identify a consumer, a place to store data, push notification services, analytics, and ad management [3]. These services are collectively known as mobile back-end as a service (MBaaS). There are three clear leaders in this space: Facebook Parse, Google Firebase, and custom build back-end.

3.2.1 Why Parse May Not Be the Best Option?

Parse has one of the best free plans out there. The documentation is the easiest to understand, and the service is one of the easiest to work with, especially for beginners [7].

Parse has sample projects available to download for free, as well as custom UIViews that can be added to one's project, such as Facebook and Twitter login views. Additionally, there is a large collection of third party libraries offered for use with the service.

While the free plans are very good, the price does seem to ramp up more quickly than other services if the free allocation is exceeded [7]. Another problem with this back-end provider is that recently, Facebook announced it is shutting down, and now everyone who used Parse has to scramble to find an alternative [6].

3.2.2 Use Firebase as an Alternative?

Firebase is one of the most popular MBaaS and for good reason. Their SDK is simple to use and supports authentication with Facebook, Google, Twitter, Github, or email [6].

The real-time database is great for something like a chat app, or any other use case where data transmission is time sensitive. The JSON tree

data architecture is a bit different than traditional data architectures and definitely requires some learning time [6].

Push Notifications are simple to setup and send through the dashboard. It's possible to create custom segments, subscribe users to pre-determined topics, or even send notifications to individual devices. Firebase Analytics is also quite robust, featuring not only custom events, but also high-level data along with demographics.

3.2.3 Write Custom Back-end Instead?

Building a custom back-end can be a daunting task, but having a bit of experience with a server-side programming language and knowing a thing or two about servers, can avoid having a dependency on a company that may shut down.

In addition, since developers have full control, they can build some handy custom features. However, with great power comes great responsibility. A custom back-end takes a lot of time to build, and afterwards requires regular maintenance – and for many small apps, this cost may not be worth the benefit. Having a custom back-end also requires the developers to take the standard security precautions like getting an SSL Certificate, protecting database access, etc.

The two most popular frameworks for building custom back-ends are Django and Ruby on Rails. However, before choosing an application framework one should explore all major features and any possible issues that come with each of them.

3.2.3.1 Django

As mentioned in the previous section, Django is one of the two most popular frameworks for building custom back-ends. It is a free and open source web application framework written in Python. Django follows the model-view-controller architectural pattern. Django facilitates fast development of clean and practical web design. This framework was mainly designed to solve two problems which are: strict requirements of its veteran developers and the rigorous requirements of newsrooms. It pushes the user to come up with excellent and innovative applications [14].

Django's goal is to aid creation of complex, database-driven websites. It emphasizes reusability and plug-ability of components, fast development and

the "don't repeat yourself" principle. Python is used extensively for files, data models as well as settings. It also offers optional interfaces such as create, read, update or even delete. The interface is generated dynamically via introspection and configured through admin models [14].

3.2.3.2 Ruby on Rails

Ruby on Rails or just Rails is the other most popular back-end frameworks. It refers to an open source web application framework that operates via the Ruby programming language. The framework allows creation of web pages and applications that gather data from the web server. It also allows querying and rendering templates out of the box. Thus, the framework features a routing channel that does not depend on the web server. It gives emphasis on using already known software engineering patterns like convention over configuration (CoC), don't repeat yourself (DRY) as well as model-view-controller [14].

3.2.4 What Back-end is HitchRide using?

Even though back-ends as a service are extremely useful, HitchRide uses custom back-end. Thus, HitchRide functionality is not dependent on other products and can support features designed and developed specifically for HitchRide.

The decision of choosing Ruby on Rails as a back-end was taken after a feature-by-feature analysis of the aforementioned most popular frameworks. The research revealed that there is a slight but visible difference between the two that makes Rails a better fit for this project. Ruby on Rails mostly focuses on developing a framework to work behind the scenes for the developer with the intention of making work easier.

Initially, HitchRide was supposed to use the already built and tested website back-end. Unfortunately, this was not possible because the website's back-end was designed to only serve browsers. After thorough research on how to build a back-end that will serve both browsers and mobile applications, it became clear that a slight modification had to be made. However, during the process of implementation, the excessive number of problems encountered necessitated the back-end to be completely rewritten, which resulted in significant delays in the overall development.

3.3 Basic Authentication vs Session Authentication

Authentication is often the foundation in most mobile applications. Whether the app in question is the next Instagram or Facebook, the user needs to be able to open the door to the wonderful universe with Sign Up and Sign In functions.

In a mobile environment, this essential functionality is typically implemented by exposing the back-end to the mobile app, thus, allowing it to make changes on the server. While this sounds relatively straight forward, this is probably one of the most important tasks since it deals with user's security and privacy, and the developers need to do their best to protect it. There are two forms of authentication - Basic Auth and Session Auth - and an app can only use one of them.

3.3.1 What is Basic Auth?

The first form of authentication - Basic Auth - authenticates each individual request using a username and password pair.

The Basic Auth token is reversible, however when all communication is over HTTPS the security context is completely protected. Basic Auth is simpler to use from HTTP client libraries. Tools such as cURL provide corresponding command line options [5].

To use Basic Auth, an app must send an HTTP Authorization header containing the username and password with every request. You will find more details on Basic Auth in IETF RFC 2617. In short, a Basic Auth authorization string is composed of the word Basic followed by a base64-encoded string containing the username and password separated by a colon [5].

3.3.2 What is Session Auth?

The second form of authentication - Session Auth - authenticates each individual request using an access-token generated by the back-end.

By using Session Auth, an app eliminates exposure of passwords on every individual request. Only the initial request for setting up the session needs to be sent with the username and password. Session Auth provides an app fine grained access control whereby the back-end can grant, reject, expire, and invalidate access on a per session basis. Security can be managed without

requiring user intervention (e.g.: change of password or disabling the account) [5].

To use Session Auth, an app must first make a login request to collect an access-token from the back-end. The access-token is returned in the header of the HTTP response. This access-token can then be used for authentication in subsequent requests to the back-end [5].

Mobile Identity Connect (MIC) provides an OAuth2 authentication interface for integrating with a number of types of identity providers, including SAML-Redirect, OpenID Connect, Active Directory, and LDAP [5].

3.3.3 How Does HitchRide Authenticate its Users?

Even though Basic auth is easier to implement, when it comes to privacy and security, one should not make any compromises. Thus, HitchRide uses Session Auth via Devise and DeviseTokenAuth - two libraries that implement the second and more secure form of authentication.

Devise is a very secure and well supported library, and in conjunction with Omniauth, makes it incredibly simple to setup Facebook and Twitter login buttons in less than a dozen lines of code. At first, the engine seems too overbearing, but once understood how easy it is to override and extend Devise, using this library feels like a piece of cake [15].

DeviseTokenAuth provides a web service facade over the core Devise. It makes it much easier for clients to authenticate and maintain a resemblance of a session with the server using token authentication [15]. Similarly to Devise, DeviseTokenAuth is a well supported library and works with Omniauth.

The main reason for choosing the combination of Devise and DeviseTokenAuth is because they both support the setting up of Facebook login. Unfortunately, implementing this feature caused the development of the back-end to fall behind schedule, which itself resulted in a delay in overall development.

4 Results

The Beta version of the HitchRide mobile app provides a solution to both problems mentioned in Section 1. It provides its users with an alternative that perhaps provides a better way of traveling, which is cheaper than the regular public transportation. Travelers can also meet new people while going to their favorite destinations.

The final version of the HitchRide mobile application will have three additional features. These features are discussed in Section 5.

4.1 Find a Ride

Users can search for a ride to their desired destination by filling in text fields for "from" and "to" locations. The User can also select the date they want to travel. Additionally, the user can view all available rides by tapping on the "See all" button. See Figure 1 for an example of blank search form.

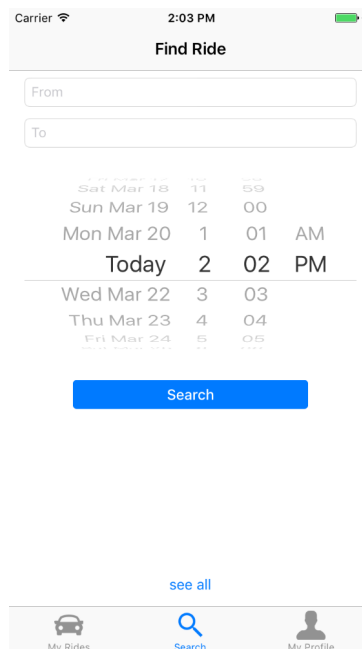


Figure 1: Find a Ride.

HitchRide shows all available rides for the next two weeks. However, if there are less than five rides found, HitchRide searches for rides to nearby locations. The default radius for nearby locations is set to 8 miles. See Figure 2 for all available rides in the database.

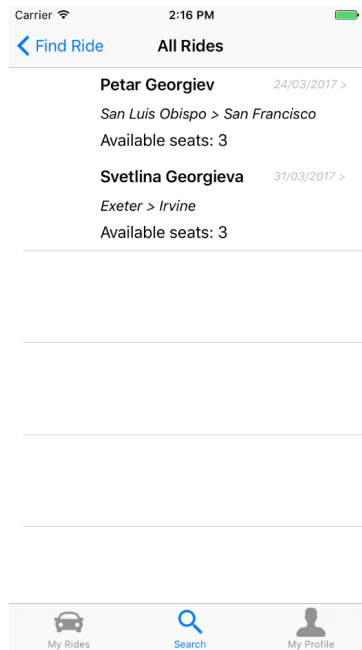


Figure 2: See All.

4.2 Posting a Ride

Users can post a ride by filling in text fields for "from" and "to" locations, gas money and short description. They can also select the date when the ride is going to occur and the seats available in the car by using the scroll menus. Touching a text field opens the default iOS keyboard and touching the category drop down menu opens the default iOS selection dialog. In order to navigate to the "Offer Ride" screen the user has to go to "My Rides" first and tap on the "+" in the top right corner. See Figures 3 and 4 for an example of a blank "Offer Ride" form.

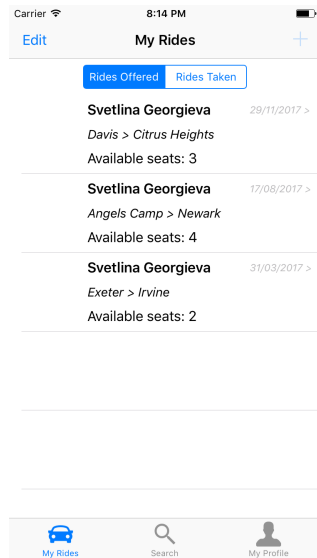


Figure 3: My Rides, Add Button Tapped.

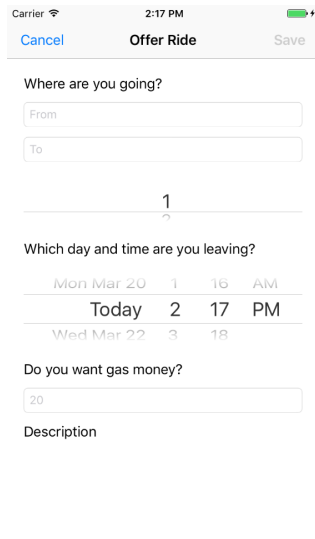


Figure 4: Offer a Ride.

The user can also edit an existing ride by simply tapping on it from the "My Rides" screen. See Figure 6 for an example of "Edit Ride" form.

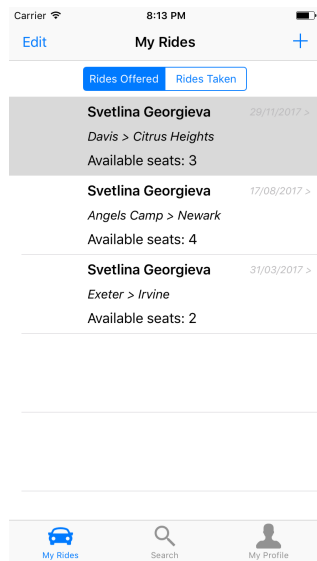


Figure 5: My Rides, Tapped on a Ride.

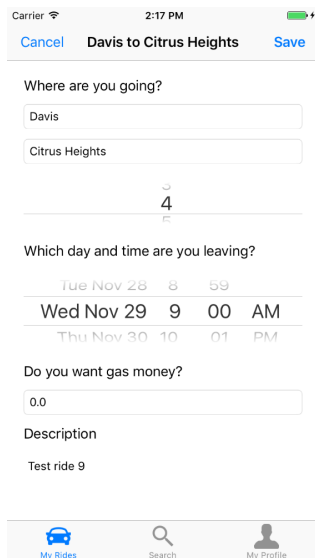


Figure 6: Edit Ride.

4.3 User Profile

Users can view and edit their profile by navigating to the "My Profile" screen. See Figure 7 for an example of a user profile screen.

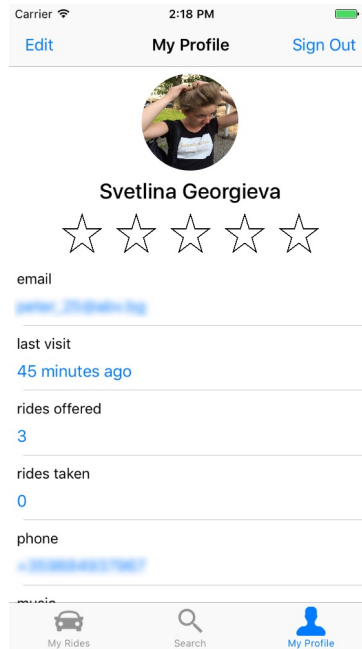


Figure 7: My Profile.

The user can sign out from their account by tapping on the "Sign Out" button at the top right corner of the screen.

5 Future Work

Three features originally intended to be included in the final version of the HitchRide mobile app were not ready.

Facebook authentication is an important feature that adds a level of extra security to HitchRide's community from a social perspective. It was not implemented because of technical issues with DeviseTokenAuth library. However, this feature has the highest priority in the list of features to be implemented before the official release of the mobile app.

”Join a Ride” is a feature that lets passengers express interest in a particular ride. The driver can review a passenger’s profile and based on their reviews accept or decline his or her ride request.

HitchRide mobile will include a five star rating system which will allow drivers and passengers to rate each other after ride has been completed. In addition to the rating, both types of users will be able to write a short review letting everyone know about their experience.

6 Reflections

Looking back on the development of the HitchRide mobile app reveals many good decisions, such as making HitchRide native app for all iOS devices, and building custom back-end.

On the other hand, an improvement could be made to the process. The choice of using DeviseTokenAuth library was made after minimal research. This caused significantly extended development time and a lack of implementation of important features.

7 Conclusion

The HitchRide mobile app successfully solves the problems it was designed to address. It is simple and minimalistic, yet completely functional. Its main attraction lies in the creative potential that it provides to any user with a smart phone. HitchRide isn’t limited to only ride-share in the traditional sense. HitchRide could also be used as a social media platform where people meet and make friends while traveling to their favorite destinations.

References

- [1] Robbie Abed. Hybrid vs native mobile apps - the answer is clear, 2016. [Accessed: 10-Mar-2017].
- [2] Matt Asay. Can we please stop fighting the native vs. web app wars? - readwrite, 2015. [Accessed: 10-Mar-2017].
- [3] Matthew David. How to choose the right mobile backend as a service (mbaaS), 2016. [Accessed: 11-Mar-2017].
- [4] Christian Karasiewicz. Swot analysis: Hybrid versus native development in ibm worklight (the mobile frontier), 2014. [Accessed: 10-Mar-2017].
- [5] Kinvey. Rest api security guide, 2017. [Accessed: 11-Mar-2017].
- [6] Kishin Manglani. Top 5 parse alternatives, 2016. [Accessed: 11-Mar-2017].
- [7] Antonio Martínez. How to choose the best backend provider for your ios app: Parse vs stackmob vs. appcelerator cloud and more!, 2012. [Accessed: 11-Mar-2017].
- [8] Hannah Peterson. 6 reasons why traveling abroad is important for young people — greenheart travel, 2017. [Accessed: 15-Mar-2017].
- [9] Plataformatec. plataformatec/devise, 2017. [Accessed: 07-Mar-2017].
- [10] PolyRides. Polyrides - polyrides makes it easy to coordinate rides with other calpoly students., 2017. [Accessed: 07-Mar-2017].
- [11] Rails. Getting started with rails - ruby on rails guides, 2017. [Accessed: 07-Mar-2017].
- [12] Within Reach. Different types of transport, 2011. [Accessed: 15-Mar-2017].
- [13] SwiftyJSON. Swiftyjson/swiftyjson, 2017. [Accessed: 07-Mar-2017].
- [14] Lindsay T. Ruby on rails vs django, 2014. [Accessed: 11-Mar-2017].
- [15] Treehouse. 10 must have ruby gems, 2011. [Accessed: 11-Mar-2017].

- [16] WikiBooks. Latex templates \LaTeX university assignment, December 2012. [Accessed: 23-Feb-2017].
- [17] Zimride. Zimride by enterprise — private network ridesharing, 2017. [Accessed: 07-Mar-2017].