



SOFE 3490U - Software Project Management

Lab 2 Project proposal

Name	ID
Aakash Patel	100616630
Karan Patel	100621178

Introduction

We will be working on the UberRUSH and we will be providing insight on the delivery service model that Uber is planning to provide to small businesses who wish to deliver their products to their clients seamlessly as possible. We will be designing two mobile application versions for both the business end users and the RUSH drivers. The reason we decided to tackle this problem is because small businesses for various reasons need quick delivery of their products. The only way to deliver products is by one of the employees driving to the delivery destination or to have the product shipped by a postal company. This can take a lot of time and might cost more to the company. With UberRUSH we can give companies the opportunity to get whatever they need delivered immediately. This fast delivery system could even help decrease the timeline of a project. This project will help us understand how the Uber application has been developed and additionally, we will learn more about the business model that helped Uber succeed.

In this report we will outline the following:

- Our objective on the UberRUSH project
- UberRUSH measures of success
- Brief description on the infrastructure that will be required for this application

Objective

The objective of this application will allow us to expand the ride sharing business by implementing a delivery service to an already successful ride sharing model. This service will help thriving small businesses to outsource their deliveries and our product will provide assurance to the businesses that their items are being delivered properly and on time with a tracking system that shows progress of the delivery. As well as the confirmation of the received package. Businesses using this service will have an interface to provide delivery instructions. This will all be done by using the UberRUSH application and a separate application will be provided for the RUSH drivers delivering the package. This application will provide an in depth interface which will display delivery instructions provided by the shipper and the drivers will

receive any notifications along with mapped routes that the driver can take to reach his/her destination.

Measure of Success

To measure the success of this application, we'll implement a ranking system which is in the normal Uber application. The sender and the receiver will have the opportunity to rate the driver and/or give feedback towards the driver and also the application itself. Lower ratings and more complaints will mean the driver is not satisfying the clients, and complaints against the application means the client is not satisfied with the application. Additionally, the number of the times the application is visited can also be a measure of success. If clients are repeatedly using UberRUSH's services then that means the service is in high demand. Simultaneously, if a lot of driver are enrolling in the program and visiting their respective app to look for clients then that even means that the business is profitable and successful. The applications dependency could also be measured by checking if the right driver is going to the right address. This means the right address from where he/she must pick up the package and the right address to which the driver must drop the package. This will measure the dependency of the application itself to give the driver the right information. An application is also successful if it is reliable, this could be tested by seeing how many times the app has crashed or how many times a client had a hard time putting in information.

Requirements

Client application	<ul style="list-style-type: none">• Request a delivery to be made• Rate the driver• Give feedback towards the application• Sender and receiver can track their package• Sender must get confirmation when package has been delivered• Sender must be able to save the address of a address they frequently would like their packages to get delivered to• Sender must be able to add a deadline by which the driver must deliver a package• See driver's maximum handling capacity• Sender can see ratings of a driver that is being
---------------------------	--

	suggested by the application
Driver application	<ul style="list-style-type: none"> • View delivery request nearby of driver's location • Provide current location • Provides both the sender and receiver locations • Maps most efficient route for the driver • Provides delivery instruction • Provides mapped routes from sender or receiver • Submit maximum package handling capacity information • Tracks total hours worked and distance travelled per shift • Submits driver's working time to database • Can take delivery conformation signatures for sender • Rate both the clients

The table above lists the requirements of the drivers application and the clients application.

Architecture:

The pattern we would like to implement for these applications is a 'Layered architecture pattern'. This pattern consists of a presentation layer which can be said to be the front-end of the application, the business layer which is the back-end of the application and the database layer.

Front-end portion:

A hybrid framework would be convenient to program this application as it would be more convenient to not have to program for both iOS and Android individually. So this application can be created with HTML, CSS and JS using the Framework7. This framework is good for prototyping as it has the capability to build indispensable apps so if a demo or progress of the application needed to be shown, this framework would come in handy.

Back-end portion:

Node.JS will be used for the backend and to connect to frameworks and other API's. Interacting with the database will be easily done in Node. Express.js will be used to set up the middleware and allow us to create many routes for the interaction with the server. We will also use libraries such as Socket.io for realtime web applications and this enables realtime, bi-directional communication between the clients and the server.

Database portion of this application will utilize technologies such as Schemaless which is built by Uber on top of MySQL, Riak, and Cassandra. Schemaless is for long-term data storage and is highly scalable compared to normal SQL solutions.

Gurafu and μ ETA are web services built on the DropWizard framework and is used by Uber for mapping and efficiency for routing and this will help us tremendously on mapping directions for the driver and showing them the most efficient directions.

Uber uses Twilio for text messaging and to implement push notifications in the iOS app, Uber uses Apple Push Notifications Service, and for the Android app they use the standard Google Cloud Messaging service. We will utilize this method for sending the user notification for convenience and for user satisfaction that Uber already provides. This way of implementing push notification will allow either the driver or the sender or shipper to receive notification.

The payment method will be implemented in a cashless manner. We will utilize PayPal's credit card scanning service such as Braintree for safe transactions.

References

- Brovkin, D. (2017, December 31). How Uber was made. medium.FreeCodeCamp.org. Retrieved February 25, 2019, from <https://web.archive.org/web/20190225150901/https://medium.freecodecamp.org/how-uber-was-made-da3c631066d0>
- Lozinski, L. (2016, July 19). Uber Engineering's Tech Stack: The Foundation. Uber Engineering Blog. Retrieved February 25, 2019, from <https://web.archive.org/web/20190225150744/https://eng.uber.com/tech-stack-part-one/>
- McKillen, R. (2012, March 6). Ryan McKillen's Answer to "What is the technology stack behind Uber?". Quora. Retrieved February 25, 2019, from <https://www.quora.com/What-is-the-technology-stack-behind-Uber/answer/Ryan-McKillen>
- Trivedi, R. (2018, April 11). Top 5 hybrid Mobile App Frameworks in 2019. Official Blog - WebsOptimization. Retrieved February 25, 2019, from <https://web.archive.org/web/20190225150615/https://www.websoptimization.com/blog/hybrid-mobile-app-frameworks/>