

A global trip price calculation system

Stefan Schenk

dept. name of organization (of Aff.)

name of organization (of Aff.)

City, Country

email address



- titel; - (ondertitel); - naam organisatie of bedrijf; - auteursnaam; - naam of namen van begeleider(s) van de HvA en vanuit het bedrijf.

CONTENTS

Preface	2
1 Abstract	2
2 Introduction	2
3 Context	2
4 Assignment	3
4.1 Origin of the assignment	3
4.2 Description	3
4.3 Research question	3
4.4 Sub questions	3
5 Literature Review	3
6 Chapter 1 - proposed approach	3
7 Realization	3
8 Conclusion	3
9 Recommendations	3
10 References	3
11 Glossary	3
12 Appendices	3
13 Reflection	3
References	3

PREFACE

Before you lies the graduation report that displays all the accomplishments and research conducted during the final phase of my Bachelor Software Engineering study at the Amsterdam University of Applied Sciences, written to fulfill the graduation requirements. Allow me to briefly elaborate on the events that motivated me to reach this point in my career.

Before I began my study, I was a marine in the Royal Dutch Marine Corps. As their latin motto *qua patet orbis* ("As Far as the World Extends") suggests, I was sent to exotic places around the world. During these trips there was either lots of waiting, or lots of hard and dangerous work. A common activity during the waiting hours of the average marine was watching series, but I wasn't really into that. Before I was sent to Afghanistan, I bought a new laptop on which I installed visual studio and downloaded C++ tutorials, convinced that I would be learning how to program during my off-duty hours. My efforts, however sincere, didn't convince me that I was progressing my understanding of the matter very much. During one of my boat trips along the coast of West-Africa I tried once more, but this time I started experimenting with Javascript instead of C++, yielding more tangible results.

After five years I decided to quit the marines because I felt that I could accomplish greater things with a proper education, and went to an open house of the HvA, HBO-ICT. There I was confronted with my interest for logic and programming once more, and decided that I was going to become a Software Engineer. During my time at the HvA, I enjoyed the logic and elegance of algorithms and patterns in code very much, next to solving complex problems, and learning new things every day in general. I've learned some shallow concepts about machine learning in the Big Data course which interested me the most. Systems that could learn complex tasks for themselves.

Right before I started writing this thesis, I followed a minor Artificial Intelligence at the University of Amsterdam, and right before that I worked at taxiID as a vacation job. That's where I was offered an internship where I would develop a virtual assistant. This aligned very well with my minor and my interest in artificial intelligence in general. However, because of priorities within the company, the assignment changed to rebuilding a price calculation system from scratch that had to be used around the globe, *qua patet orbis*.

To many this may sound as a boring challenge. If

you are not one of those people, you may think about the many ways in which this problem may be solved. And this thesis will provide useful solutions for encoding locations and handling geospatial data in an intelligent and performant way. I was also challenged by the fact that my project had to supercede the existing system in effectiveness and efficiency. The moment I was introduced to my assignment reminded me of a chapter in a book called Clean Code, in which this exact pursuit is used as a common example for which the book would provide solutions; "Now the two teams are in a race. The tiger team must build a new system that does everything that the old system does. Not only that, they have to keep up with the changes that are continuously being made to the old system. Management will not replace the old system until the new system can do everything that the old system does." [1].

I would like to thank Dan Stefancu, Marco Strijker and Martin Zwaneveld for their insightful criticism that has led to the most useful lessons during my internship.

Stefan Schenk

Andijk, 01-03-2018

1. ABSTRACT

Abstract—

Index Terms—encoding, locations, geospatial,

2. INTRODUCTION

This document is a model and instructions for \LaTeX . Please observe the conference page limits.

3. CONTEXT

taxiID is an Amsterdam based company providing end-to-end cloud solutions for taxi companies. Founded as a startup that successfully introduced smartphone taxi booking in The Netherlands, taxiID offers a wide range of IT solutions to serve the taxi market, including a passenger app, a driver app, administrative panels, and track and trace hardware.

taxiID solutions have proven to be a reliable set of tools for all size businesses. For independent taxi companies with 2 cars or a companies with large fleets, affordable solutions are available. taxiID's goal is to deliver affordable, time-saving solutions for taxi companies to allow for convenient planning and dispatching without requiring local installation.

Tough based in Amsterdam, the development team is located in Medemblik, consisting of two mobile developers, two backend developers, a designer and two project managers.

taxiID's customers are located across the globe. This introduces challenges when developing applications that rely on clearly defined locations and infrastructure of all kinds.

4. ASSIGNMENT

A. *Origin of the assignment*

YourDriverApp (YDA) requires a pricing calculation functionality that is similar to the existing taxiID implementation. All functionalities within the current system align with the clients demands, but some features bring certain difficulties along, for example: region names are too vague for specific database queries. Some features could be abstracted so more possibilities can be implemented, some features are still unimplemented, and some features could be improved along the way.

B. *Description*

A system must be implemented in which group admins can define pricing rules based on user defined locations and time schedules, that can be used for calculating a passengers trip price, or show prices of different products based on the trip the passenger is about to make. For example: a passenger may book a taxi ride from Utrecht to Schiphol using the passenger app. Available products are presented with their respective prices based on the distance and duration of the trip using the pricing rules that were created by the group admin of a taxi company.

The system must be usable in countries with a poor postal code system. There should be a way for a group admin to describe locations in a way that are precise and consistent with reality, meaning that a defined location should be usable from outside of the system, or at least be interpretable. An example of this requirement would be: a taxi company that operates in Afghanistan. A passenger wants to be picked up on some road near the mountains. How would a group admin describe that location in order to define a price beforehand?

The system should be accessible to other systems, meaning that applications that currently rely on the old system should be able to migrate to the new system. As the old system shouldn't be used for new applications, as it was not designed for this use case, the new system should. It should have a single responsibility, and should be autonomous in that regard.

C. *Research question*

From the description of the problem, one main important question could be derived. How can a generic price calculation system be implemented that is usable across the globe?

D. *Sub questions*

Which varieties of address formats exist around the world?

Is it possible to redefine different address formats as a generic address? - Gps - Zip code

To what extent do address formats have an impact on performance of a price calculation?

5. LITERATURE REVIEW

6. CHAPTER 1 - PROPOSED APPROACH

7. REALIZATION

8. CONCLUSION

9. RECOMMENDATIONS

10. REFERENCES

11. GLOSSARY

12. APPENDICES

13. REFLECTION

REFERENCES

- [1] R. C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship. Pearson Education, August 2008, pp. 4, 5.