Niceway.to
Crowd Sourced Scenic Route Sharing

Submitted March 2016 in partial fulfilment of the conditions of the award of the degree MSci (Hons) Computer Science

Craig Knott
cxk01u

With Supervision from Max L. Wilson

School of Computer Science and Information Technology
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature _____

Date _____/_____/_____

Abstract

Project abstract

Words in text | 19,552

Calculated with the TeXCount web service
`http://app.uio.no/ifi/texcount/online.php`

Contents

1 Introduction

- Powerful opener, similar to proposal opener
- Mention other software briefly and why it's bad
- Main objective of this project, and how it will be achieved
- Where this project will not focus, if any

2 Motivation

- Over arching motivation - **what is being solved?**
- Mention previous work for client
- Personal ties to the project?
- What is good about this project existing?
- Mention good HCI (see yr 3 diss)

3 Background Information & Research

3.1 What is Scenic Route Mapping

- Not sure about this section, just kept it because I had 'What is fuzzy logic' in my diss last year

3.2 Existing Systems

- Since routing/mapping began.. there has been many pieces of software.. here are a few

SW X

- What is it
- What is does
- Why it's good
- Why it's not

SW Y

- What is it
- What is does
- Why it's good
- Why it's not

SW Z

- What is it
- What is does
- Why it's good
- Why it's not

3.3 Platforms and Tools

- This section introduces some potential platforms and tools that could have been used in the project, along with justifications for and against...

3.3.1 System Back End

Backend X

- What is it
- What is does
- Why it's good
- Why it's not

Backend Y

- What is it
- What is does
- Why it's good
- Why it's not

Backend Z

- What is it
- What is does
- Why it's good
- Why it's not

3.3.2 Front End Programming Language

- not sure if this is overly necessary. Could condense this language + framework into one?

Front End X

- What is it
- What is does
- Why it's good
- Why it's not

Front End Y

- What is it
- What is does
- Why it's good
- Why it's not

3.3.3 Front End Design Framework

- not sure if this is overly necessary. Could condense this language + framework into one?

Front End X

- What is it
- What is does
- Why it's good
- Why it's not

Front End Y

- What is it
- What is does
- Why it's good
- Why it's not

4 System Specification

- little bit of intro speil

4.1 Functional Requirements

- C+V from design spec

4.2 Non-Functional Requirements

- C+V from design spec

5 System Designs

In this section, all of the design aspects of this system have been detailed and justified.

- containing a comprehensive description of the design chosen, how it addresses the problem, and why it is designed the way it is.

5.1 UI Design

- Screenshots of initial designs + justifications
- Screenshots of final designs + justifications + reasons for changes
- Screenshots of actual final system + reasons for changes

5.2 Navigation/Control Flow Design

- The generally expected path for a user to take through the system + picture
- Explain how design facilitates this
- Explain navigation allowing random access

5.3 Internal Design

- Models/Controller/Views
- Languages
- Flow of data from database -> view (don't forget AJAX calls)
- See image from last year diss

6 Software Implementation

In this section, the actual implementation of the software has been detailed, including: what tools were used in the implementation, how the software was implemented, and any issues that were encountered during the implementation process.

- Screenshots of initial designs + justifications
- Screenshots of final designs + justifications + reasons for changes
- Screenshots of actual final system + reasons for changes

6.1 Key Implementation Decisions

- From the background research section, list all the technology I chose to use and why

6.2 Implementation Methodology

- To help manage the implementation of such a large piece of software, the adoption of some methodology was necessary. It was decided that the best methodology would be an agile one, with heavy use of Kanban, using the tips laid out by Henrik Kniberg [2]. In order to accomplish this, at the beginning of the implementation stage, after the requirements specification had been detailed, the entire project was split into user stories. Each of these stories detailed a specific action that a user of the system would be able to accomplish, along with how long it should take to implement, how important it was, and a way of testing its completion. These stories were then organised onto a digital Kanban Board, using a service called Trello[1].

  Each week, a set of tasks would be selected to be worked on for that week. The amount of tasks selected would be dependent on how much was completed, on average, in the weeks before, so that reasonable estimates could be made (obviously excluding the first few weeks). This ensured a decent portion of work was being completed per week, and that progress was constant. During the week, tasks would be selected from the available pool, prioritising those that were prerequisites of others, or had a high importance, and would then be worked on until completion. After the completion of a task, a new task would be selected, and work would begin on this. This was an extremely effective method of managing the implementation, as any small tasks that were necessary could be added to the board, and there was an assurance they would eventually be completed, and nothing would be overlooked. It has also been shown that it is much easier to reach goals if they have been written down [3], which a Kanban Board was the perfect tool for.

  Also mention weekly meetings with max and use of Gantt chart. Also mention this is what I did at work and in my last diss and found it the best way to work for me?

6.3 Detailed Description of the User Interface

- In this section, each individual screen of the system has been displayed, along with a detailed explanation of why it is effective, and why it has been implemented as it has.

[1] `http://www.trello.com`

- Potentially don't need this? Or a slimmed down version

6.4 Implementation of System Components

- Potentially don't need this
- Do last, look at last year's diss

6.5 Problems Encountered

- Look through problem log document and pick out key things, especially those with lessons
- What happened / what this affected / how the project was affected / what I would do differently / why it happened

7 External Aspect

- Very similar to proposal
- as well as explicitly addresses how your project fulfilled (or not) its original intentions with regard to its 'external aspect'.
- maybe put this section AFTER evaluation? (or just combine the two...?)

8 Evaluation of the Project

8.1 Functional Testing

- In this section, each of the functional requirements laid out in section 4.1 have been evaluated in turn, to ensure the system meets them. Knowledge of the inner workings of the system is not actually necessary to understand these tests, as they simply check whether functionality is present, and are not concerned as to how the system actually implements it (this is known as black box testing [1]). A complete listing of all the tests conducted, and their results, can be found in appendix **??**.

8.2 Non-Functional Testing

- Look at non-functional requirements and talk about if they were met

8.3 User Feedback Testing

- user feedback / questionnaire / focus group / test users? + their feedback

8.4 Successes and Limitations of the Project

- As a result of the test...
- x was good
- y was bad

9 Further Work

- 2-4 bigs things that I would do next time (either changing something I did, or adding/removing something)

10 Summary & Personal Evaluation

- Personally, I feel as thought the project was a /success|failure/
- I felt as though I /failed to rise|rose/ to the challenge
- One of the areas I feel as though was weaker within the project
- If I could work on this project again

References

[1] Boris Beizer. *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc., 1995.

[2] Henrik Kniberg. Scrum and xp from the trenches. *Lulu. com*, 2007.

[3] Susan B Wilson and Michael S Dobson. *Goal Setting: How to Create an Action Plan and Achieve Your Goals*. Publisher - Amacom, 2008.

A Appendix