**TravelPal**

Planner for travel & everyday events

**Benedict Chukwubuikem Benson-Obilom**

**20447376**

Final Year Project – **2023**

B.Sc. Single/Double Honours in

Computer Science/Computer Science and Software Engineering



Department of Computer Science

Maynooth University

Maynooth, Co. Kildare

Ireland

A thesis submitted in partial fulfilment of the requirements for the B.Sc.

Single/Double Honours in Computer Science/Computer Science and Software

Engineering.

Supervisor: **Joseph Duffin***.*

Contents

[Declaration i](#_Toc476122285)

[Acknowledgements ii](#_Toc476122286)

[Abstract iii](#_Toc476122287)

[List of Figures iv](#_Toc476122288)

[List of Tables iv](#_Toc476122289)

[**Chapter one: Introduction** 1](#_Toc476122290)

[Summary 1](#_Toc476122291)

[1.1 Topic addressed in this project 1](#_Toc476122292)

[1.2 Motivation 1](#_Toc476122293)

[1.3 Problem statement 1](#_Toc476122294)

[1.4 Approach 1](#_Toc476122295)

[1.5 Metrics 2](#_Toc476122296)

[1.6 Project 2](#_Toc476122297)

[**Chapter two: Technical Background** 3](#_Toc476122298)

[Summary 3](#_Toc476122299)

[2.1 Topic material 3](#_Toc476122300)

[2.2 Technical material 3](#_Toc476122301)

[**Chapter three: The Problem** 4](#_Toc476122302)

[Summary 4](#_Toc476122303)

[3.1 Project UML documentation 4](#_Toc476122304)

[3.2 Problem analysis 4](#_Toc476122305)

[**Chapter four: The Solution** 5](#_Toc476122306)

[Summary 5](#_Toc476122307)

[Depending on your type of project, you may not need to include all of these: 5](#_Toc476122308)

[4.1 Analytical Work 5](#_Toc476122309)

[4.2 Architectural Level 5](#_Toc476122310)

[4.2 High Level 5](#_Toc476122311)

[E.g. Packages, Class Diagrams, etc. 5](#_Toc476122312)

[4.2 Low Level 5](#_Toc476122313)

[E.g. Method specifications, Algorithms, etc. 5](#_Toc476122314)

[4.2 Implementation 5](#_Toc476122315)

[**Chapter five: Evaluation** 6](#_Toc476122316)

[Summary 6](#_Toc476122317)

[5.1 Solution Verification 6](#_Toc476122318)

[E.g. use your equations to verify the correctness of your solution 6](#_Toc476122319)

[5.2 Software Design Verification 6](#_Toc476122320)

[5.3 Software Verification 6](#_Toc476122321)

[5.3.1 Your test approach (i.e. unit testing, sub-system testing, system testing) 6](#_Toc476122322)

[5.3.2 Your tests (e.g. scenarios, test cases, test data, etc.) 6](#_Toc476122323)

[5.3.3 Your test results 6](#_Toc476122324)

[5.3.4 An interpretation of the results 6](#_Toc476122325)

[5.4 Validation/Measurements 6](#_Toc476122326)

[5.4.1 Results 6](#_Toc476122327)

[5.4.2 Explanation of Results 7](#_Toc476122328)

[5.4.3 Analysis of Results 7](#_Toc476122329)

[5.4.4 Comparison with previous solutions (if relevant) 7](#_Toc476122330)

[**Chapter six: Conclusion** 8](#_Toc476122331)

[**Summary** 8](#_Toc476122332)

[**5.1** **Contribution to the state-of-the-art** 8](#_Toc476122333)

[**5.2** **Results discussion** 8](#_Toc476122334)

[**5.3** **Project Approach** 8](#_Toc476122335)

[**5.3** **Future Work** 8](#_Toc476122336)

[**References** 9](#_Toc476122337)

[**Appendices** 10](#_Toc476122338)

[Appendix 1 Schematic of the hardware associated with this project. 11](#_Toc476122339)

[Appendix 2 Code developed for this project. 12](#_Toc476122340)

[Appendix 3 UML Class, Use Case and sequence diagrams for this project. 13](#_Toc476122341)

[Appendix 4 Screen shots of the project implementation 14](#_Toc476122342)

## Declaration

I hereby certify that this material, which I now submit for assessment on the program of study as part of **Computer Science & Software Engineering** qualification, is *entirely* my own work and has not been taken from the work of others - save and to the extent that such work has been cited and acknowledged within the text of my work.

I hereby acknowledge and accept that this thesis may be distributed to future final year students, as an example of the standard expected of final year projects.

Signed: Date: 09/11/2023



## Acknowledgements

## Abstract

Style

The abstract should be a microcosm of the full report.

The abstract must be self-contained, **without** abbreviations, footnotes, or references.

The abstract must be between 150-250 words.

The abstract must be written as one paragraph, and **should not** contain displayed mathematical equations or tabular material.

The abstract should include three or four different keywords or phrases, as this will help readers to find it.

Ensure that your abstract reads well and is grammatically correct.

## List of Figures

[Figure 3‑1 UML class diagram overview for this project. 4](#_Toc445718606)

## List of Tables

[Table 2‑1 Table of interest: Aspect of your implementation 2](#_Toc445714278)

[Table 2‑2 Data sources used in your implementation 2](#_Toc445714279)

# **Chapter one: Introduction**

## Summary

Chapter 1 describes my "TravelPal" project, a Spring Boot application designed to facilitate travel planning. It outlines the initial inspiration and motivation behind the project, the problem statement it aims to address, the approach taken to develop the application and the requirements for success.

## 1.1 Topic addressed in this project

TravelPal is a simple user-friendly travel itinerary planner.

The project focuses on developing an application that assists users in creating and organizing their itineraries.

Inspired by a previously proposed Weather Forecast application, TravelPal aims to provide a platform for users to plan their travels, integrating functionalities to create, manage, and review itineraries and destinations.

The project is visualized through class diagrams, emphasizing a structured approach to database design.

## 1.2 Motivation

At the core of the TravelPal is a simple idea, make travel planning easier. When you look at most travel apps out there, they try to do too much. They're packed with features, buttons, and options that can become overwhelming, and even take the fun out of planning a trip.

Initially, TravelPal was ambitious and wanted to go down the same road to create an all-in-one app that could handle everything. It aimed to provide users with the tools to plan trips, create dynamic itineraries, customize routes with real-time information, and access destination-specific weather forecasts. But I quickly realized that this "jack of all trades" approach might make the app less efficient, a lot more complicated to develop on my own (as my first react project), and overwhelming for users.

This realization sparked a change in direction. My diary entries and the images of my work show a shift from trying to include everything to focusing on what matters most: creating a user-friendly app that does the hard work of travel planning without the fuss. The class diagrams became less cluttered, the features more focused, and the vision for my user interface became clearer.

In short, the motivation behind TravelPal is about giving users a straightforward, easy-to-use app that takes the stress out of travel planning. It's about doing less, but better.

## 1.3 Problem statement

Describe the technical problem needed to be solved in your project. Note that most projects solve both a more abstract, high-level problem and a specific, technical problem: your problem statement is the detailed technical problem (your motivation should cover the more abstract high-level problem).

## 1.4 Approach

Summarise how you addressed solving the problem.

Provide an overview of how you analysed the problem, how you designed a solution, and how you evaluated your solution. (e.g. use of models, simulation, prototypes, real-world experiments, cases studies, etc.). What important variables did you control, ignore, or measure in your evaluation.

## 1.5 Metrics

Describe how you are going to evaluate your work.

## 1.6 Project

List, and briefly describe your significant achievements in the project (probably 3-5 of these in a typical project). If you have come up with any contributions

# **Chapter two: Technical Background**

## Summary

The purpose of this chapter is to show your depth and breadth of reading and understanding of the problem domain

## 2.1 Topic material

(Research material, if used, from published journals and conference proceedings; less academic publications, if required by the project, from other sources) – for example, what other work researchers have done already in this area, what results they have produced, what work has been done in related areas, what software already exists to solve this or similar problems, etc.

## 2.2 Technical material

(From any source: including books, websites) – for example, how to write a web server, how to use specific Java features, how to use Ajax, how to use UML to validate your design, etc.

NB: Note that material relating to the motivation or non-technical background should **NOT** go here, but rather in the introduction

Table 2‑1 Table of interest: Aspect of your implementation

|  |  |
| --- | --- |
| **Column description 1** | **Column description 2** |
| A | Text 1 |
| B | Text 2 |
| C | Text 3 |

Table 2‑2 Data sources used in your implementation

|  |  |  |
| --- | --- | --- |
| **Column description 1** | **Column description 2** | **Column description 3** |
| X | 22 | 33 |
| Y | 33 | 456 |
| Z | 17 | 22 |

# **Chapter three: The Problem**

## Summary

The purpose of this chapter is to clearly explain the technical problem and/or identify the user requirements.

## 3.1 Project UML documentation

Provide any model(s) of the problem (e.g. equations, ERD’s, UML Use Cases & Scenarios, Activity Diagrams, etc.)



Figure 3‑1 UML class diagram overview for this project.

## 3.2 Problem analysis

Provide any analysis of the problem, leading to a greater understanding

There should be no decisions made in this chapter

# **Chapter four: The Solution**

## Summary

The purpose of this chapter is to clearly identify, discuss, and justify the decisions you make

## Depending on your type of project, you may not need to include all of these:

## 4.1 Analytical Work

E.g. Equations, etc. that describe your solution

## 4.2 Architectural Level

E.g. Implementation Diagrams

## 4.2 High Level

## E.g. Packages, Class Diagrams, etc.

## 4.2 Low Level

## E.g. Method specifications, Algorithms, etc.

## 4.2 Implementation

Discuss anything interesting here; put full source code in an appendix or attachment

# **Chapter five: Evaluation**

## Summary

Chapter 5 describes……..

## 5.1 Solution Verification

## E.g. use your equations to verify the correctness of your solution

## 5.2 Software Design Verification

How did you show that your design worked properly?

Using a model of your solution. E.g. use UML interaction diagrams to verify each scenario.

## 5.3 Software Verification

How did you demonstrate your software worked properly?

If you have not tested your software, then you cannot rely on your results. Clearly describe:

### 5.3.1 Your test approach (i.e. unit testing, sub-system testing, system testing)

### 5.3.2 Your tests (e.g. scenarios, test cases, test data, etc.)

### 5.3.3 Your test results

### 5.3.4 An interpretation of the results

## 5.4 Validation/Measurements

How did you measure how well your solution solved the problem.

### 5.4.1 Results

### 5.4.2 Explanation of Results

### 5.4.3 Analysis of Results

### 5.4.4 Comparison with previous solutions (if relevant)

**Chapter six: Conclusion**

**Summary**

Chapter 5 identifies and discuss the implications of your work.

**5.1 Contribution to the state-of-the-art**

If you made a contribution to the state-of-the-art, clearly identify it here.

**5.2 Results discussion**

Discuss whether your results are general, potentially generalizable, or specific to a particular case. Identify threats to the validity of your results (e.g. limitations, risks introduced by your approach, etc.)

**5.3 Project Approach**

Discuss your project approach

**5.3 Future Work**

Discuss future work, based on what you have done (and not done)

# **References**

Action Research : A definition . (2015). Retrieved February 25, 2016, from http://valenciacollege.edu/faculty/development/tla/actionResearch/ARP\_softchalk/ARP\_softchalk\_print.html

Hammersley, M. (1993). On the teacher as researcher. In M. Hammersley (Ed.), *Educational Research: Volume One: Current Issues* (pp. 211–231). The Open University.

Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Source: Administrative Science Quarterly Qualitative Methodology*, *24*(4), 602–611. Retrieved from http://www.jstor.org/stable/2392366

Kemmis, S. (1993). Action Research. In M. Hammersley (Ed.), *Educational Research: Volume One: Current Issues* (pp. 175–190). The Open University.

Kolb, D. (1984). *Experiential learning*. New Jersey: Prentice Hall.

McNiff, J., Lomax, P., & Whitehead, J. (2003). *You and Your Action Research Project* (2nd ed.). London & New York: London & New York.

**Appendices**

Include here all extra material, e.g. your source code, project management (optional) including: the task list, Gantt Chart diagrams (or equivalent), discussion of any significant deviations from plan, and how you managed them, discussion of what you would do differently if you repeated the project.

## Appendix 1 Schematic of the hardware associated with this project.

## Appendix 2 Code developed for this project.

## Appendix 3 UML Class, Use Case and sequence diagrams for this project.

|  |
| --- |
|  |
| Appendix 4 Screen shots of the project implementation |
|  |