USAMA
161120
SAMI AHMAD MALIK
161132

CarPool

Bachelor of Science in Computer Science

Supervisor: Mr. Shoaib Malik

Co-Supervisor: Dr. fahad

Department of Computer Science
Air University, Islamabad

June 2020

Certificate

We accept the work contained in the report titled "TITLE OF THE REPORT", written by Mr. AUTHOR1 NAME AND Mr. AUTHOR2 NAME as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Name of the Supervisor (Title)

_____

Internal Examiner: Name of the Internal Examiner (Title)

_____

External Examiner: Name of the External Examiner (Title)

_____

Project Coordinator: Name of the Project Coordinator (Title)

_____

Head of the Department: Name of the HOD (Title)

_____

September 31st, 2015

Abstract

In general, people have a hard time conciliating their schedules because of the way they move from one location to another. And students suffer from this the most especially since transportation between cities is not that great, As students, we think there should exist more suitable transportation solutions to places where transportation networks are short and cheap and helpful for students.
This report proposes a platform to help improve students mobility through carpooling, a way for vehicle owning students to share their private vehicle with non vehicle owning students in order to splitting and reducing costs. Carpooling may be one of the best solutions when there is no other mean of transportation to a specific location but naturally it is not the only one. Mobile applications take more and more part of everyone's lives, different services for carpooling with different features begin to compete with existing transportation solutions. Some people start to prefer using new carpooling services over the traditional services represented by taxi services. CarPool aims to promote carpooling by targeting students making it easier for them to adhere and use this system. By targeting students people will more likely join the service since its users are primarily other people form the same environment. To put the carpooling system in place, we have designed and developed an Android mobile application with backend servers for users to access the carpooling service through their smartphones, additionally the application involves some features that are critical to the service. By using Android Development Tools and Libraries and efficient backend solutions we have managed to make the application simple but powerful as well, which makes this application very useful for the students to use.
The combination of the smart phone and the Internet service is the trend of the future information development and software applications. Mobile phones are the most commonly used communication tools. Using mobile phones to obtain information is not only quick, but also more convenient shortcut to improve people's lives. In the paper, we propose the software development architecture based on Web services. This framework introduces the three-layer architecture of Web development into mobile phone software development. Based on the threelayer architecture, the android based CarPool system is developed.
Our app CarPool will be a unique carpooling application that would take benefits of the advantages of carpooling and try to improve and eliminate the disadvantages, all while focusing on making it a good carpooling experience for students. The realization of our project will go through the conceptual phase and then development phase. Since making a good application requires good planning first.

Acknowledgments

We wish to thank various people for their contribution to this project.we would like to express our deep gratitude to Sir Shoaib Malik, our Project supervisor, for their patient guidance, enthusiastic encouragement and useful critiques of this Project work. we would also like to thank Dr Touseef, our project co-supervisor for his advice and assistance in keeping our progress on schedule. we would also like to extend our thanks to the librarians of the Air University for their help in offering us the resources in running the program. Finally, We wish to thanks our parents for their support and encouragement throughout our study.

USAMA, SAMI AHMED MALIK
Islamabad, Pakistan

June 2020

"We think someone else, someone smarter than us,
someone more capable, someone with more resources will solve that problem.
But there isn't anyone else."

Regina Dugan

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

DSA	Data Structure and Algorithms
OOP	Object Oriented Programming
PF	Programming Fundamentals
SE	Software Engineering
SQL	Structured Query Language
UNESCO	United Nations Educational, Scientific and Cultural Organization
UNICODE	Unique, Universal, and Uniform Character enCoding
XML	Extensible Markup Language

Chapter 1

Introduction

What is CarPool?

1.1 Project Background/ Review

Carpooling (also car-sharing, ride-sharing and lift-sharing) is the sharing of car journeys so that more than one person travels in a car, and prevents the need for others to have to drive to a location themselves.
Drivers and passengers offer and search for journeys through one of the several mediums available. After finding a match they contact each other to arrange any details for the journey(s). Costs, meeting points and other details like space for luggage are agreed on. They then meet and carry out their shared car journey(s) as planned.
By having more people using one vehicle, carpooling reduces each person's travel costs such fuel costs, tolls and the stress of driving. Authorities often encourage carpooling, especially during periods of high pollution or high fuel prices. Car sharing is a good way to use up the full seating capacity of a car, which would otherwise remain unused if it were just the driver using the car.
In 2009, carpooling represented 43.5In 2011, an organization called Greenock created a campaign to encourage others to use this form of transportation in order to reduce their own carbon footprint.
Carpooling, or car sharing as it is called in British English, is promoted by a national UK charity, Carplus, whose mission is to promote responsible car use in order to alleviate financial, environmental and social costs of motoring today, and encourage new approaches to car dependency in the UK. Carplus is supported by transport for London, the British government initiative to reduce congestion and parking pressure and contribute to relieving the burden on the environment and to the reduction of traffic-related air-pollution.

Cabbing All the Way is a book written by author Jatin Kuberkar that narrates a success story of a carpool with twelve people on board. Based in the city of Hyderabad, India, the book is a real-life narration and highlights the potential benefits of having a carpool

1.2 Problem description

Many vehicle-owning Students who commute on daily basis often have unoccupied seats in their vehicles. Many non-vehicle owning students find it very difficult sometimes to find ride for travelling to and from university.

1.3 Project Objectives

Objective

- To Allow vehicle owning students to share their rides with other students for traveling to and from their institutes and cut down their fuel bills.
- To facilitate non-vehicle owning students for travelling to and from university easier and cheaper.

Goals

- Cost Effective: Much Cheaper than Cab services.
- Ease of getting ride: Riders are easy approachable, which reduces the tension of finding and catching of local transport right on time.
- Fewer Cars on the road will have reduced fuel consumption which will make environment Eco-friendly.

1.4 Project Scope

This project (CarPool) aims to develop an Android based application for carpooling for students, this application allows vehicle owning students to submit rides for specific targets and allows passengers to reserve/request rides from drivers all while being secure and having a simple interface.
This application will help students save money and also reduce the pollution of the environment and effects of vehicles, this application focuses on serving needs of students. CarPool will be intended for the students in Air University and it will support Android phones and Tablets, Users will need internet connection to use the application to offer or find a common route to travel.
The application will have a simple and easy interface, Users must register at first before

using the application, after that they must choose between a driver or a passenger, a driver can offer a drive to a specific location while a passenger can find or request a ride to a location.

1.5 The Degree of Project Report

In our FYP-1, we presented our idea that how CarPool would be beneficial. The only purpose of FYP-1 was to present and defend the idea. We have completed both tasks successfully and we also developed some mockup screens to present our idea.
However, in fyp-2, the task assigned to us was to develop a working application for two users: driver and rider along with the implementation of the core feature of our application, which was location tracking of driver and rider, fetching current location, use Firebase it's Real-time Database which is a NoSQL Fast Databaseand displaying that location on the map using Google Map API.

Chapter 2

Literature Review

Carpooling

2.1 Definition and general principle

Carpooling (also car-sharing, ride-sharing and lift-sharing) is the sharing of car journeys so that more than one person travels in a car, and prevents the need for others to have to drive to a location themselves.
By having more people using one vehicle, carpooling reduces each person's travel costs such as: fuel costs, tolls, and the stress of driving. Carpooling is also a more environmentally friendly and sustainable way to travel as sharing journeys reduces air pollution, carbon emissions, traffic congestion on the roads, and the need for parking spaces. Authorities often encourage carpooling, especially during periods of high pollution or high fuel prices. Car sharing is a good way to use up the full seating capacity of a car, which would otherwise remain unused if it were just the driver using the car.
In 2009, carpooling represented 43.5commute trips. The majority of carpool commutes (over 60members.
Carpool commuting is more popular for people who work in places with more jobs nearby, and who live in places with higher residential densities. Carpooling is significantly correlated with transport operating costs, including fuel prices and commute length, and with measures of social capital, such as time spent with others, time spent eating and drinking and being unmarried. However, carpooling is significantly less likely among people who spend more time at work, elderly people, and homeowners.
Carpooling usually means to divide the travel expenses equally between all the occupants of the vehicle (driver or passenger). The driver does not try to earn money, but to share with several people the cost of a trip he would do anyway. The expenses to be divided basically

include the fuel and possible tolls. But if we include in the calculation the depreciation of the vehicle purchase and maintenance, insurance and taxes paid by the driver, we get a cost around 100DA/km. There are platforms that facilitate carpooling by connecting people seeking respectively passengers and drivers. Usually there is a fare set up by the car driver and accepted by passengers because they get an agreement before trip start.

2.2 Carpooling Types

2.2.1 Regular

The car is often perceived as an extension of the personal space, the driver, alone in his vehicle is in a closed space; he is free to do what he likes: listen to the radio, sing, call with headsets ... Carpooling regularly is to share a dialogue, experiences, stories. In the United States an intermediate concept has developed between carpooling and the public transport line: the Vanpool. These are minibuses chartered by an employer, a public authority or a private company and made available to a group of people who regularly make the same journey.

2.2.2 Occasional

This type of carpooling is mainly used for leisure or last minute departures. The linking is often done through websites or mobile applications, which can significantly reduce travel costs, but usually requires to carpool with one or more unknown. This type of carpooling is mainly used for leisure or last minute departures. The linking is often done through websites or mobile applications, which can significantly reduce travel costs, but usually requires to carpool with one or more unknown.

2.2.3 Eventual

Participants in an event (music festival, sporting event, wedding, associative or institutional meeting ...) can organize to carpool to the venue of the event. This one-time carpool has a special feature: all participants travel to the same place on the same date. Carpooling is also used for departures on holidays or weekends, savings on a trip being even larger than the trip is long. So carpooling becomes an alternative of affordable and accessible transportation.
There are also "cultural" carpooling platforms to visit a cultural site: castles, museums, exhibitions, artists' studios, religious places, festivals, etc.

2.3 Requirement Specification

2.3.1 Existing System for carpooling

Many carpoolings applications and websites have been developed around the world. A similar carpooling system was developed in Massey University New Zealand by a group of students to allow students of Massey University, Albany campus to share their vehicle with non-vehicle owning students.
Following some examples of carpooling systems around the globe :

2.3.2 Websites

- New Zealand: https://www.asa.ac.nz/carpool
- Algeria: www.nroho.com, www.m3aya.com,www.nsogo.net
- Europe: BlaBlaCar.com, carpooling.com, GoMore.com
- France: covoiturage.fr
- USA: car.ma , www.rdvouz.com
- World: Outpost.travel , joinntravel.com , www.letsride.in

2.3.3 Mobile Applications

- New Zealand: ASA
- Algeria: YAssir,Nsogo, AMIR
- World: Uber, sRide, RideShare,
- USA: Uber, Lyft
- France: Karos, Wever, BlaBlaCar, OuiHop

2.4 Proposed System

Our purposed system is a “Carpool” application which is a ride sharing application designed just for students.students can login or signup to this application only via university email id to make sure that only enrolled students in a university used this application.
Vehicle owning students can share their rides with other students for traveling to and from their institutes and earn money.

2.5 Requirement Specifications

It involves functional and non-functional functionalities that must be performed by the system:

2.5.1 Functional Requirements

2.5.1.1 Table 1: Functional Requirement - 01

Identifier	FR-01
Title	Create Account
Requirement	Registered New User
Source	Supervisor, M. Shoaib Malik
Rationale	To registered new users
Restrictions and Risk	User can only be registered via university email
Dependencies	Android phone, Google API, Firebase server, Firebase Authentication
Priority	High

2.5.1.2 Table 2: Functional Requirement - 02

Identifier	FR-02
Title	Sign In
Requirement	Already registered
Source	Supervisor, M. Shoaib Malik
Rationale	It's essential to use this application
Restrictions and Risk	User must be registered on this application
Dependencies	Google API, Firebase server, Firebase Authentication
Priority	High

2.5.1.3 Table 3: Functional Requirement - 03

Identifier	FR-03
Title	Reset Password
Requirement	Already registered on this application.
Source	Supervisor, M. Shoaib Malik
Rationale	Reset user account password
Restrictions and Risk	Have access to university email
Dependencies	Firebase server, Firebase Authentication, Google Map Api
Priority	Low

2.5.1.4 Table 4: Functional Requirement - 04

Identifier	FR-04
Title	Switch to Driver / Rider
Requirement	Sign In
Source	Supervisor, M. Shoaib Malik
Rationale	Confirm the role of user
Restrictions and Risk	User have to choose only one role at a time.
Dependencies	Firebase server
Priority	Medium

2.5.1.5 Table 5: Functional Requirement - 05

Identifier	FR-05
Title	Vehicle Details (Driver)
Requirement	Choose vehicle type car / bike
Source	Supervisor, M. Shoaib Malik
Rationale	To make sure the vehicle type and detail
Restrictions and Risk	Nil
Dependencies	Firebase server
Priority	Medium

2.5.1.6 Table 6: Functional Requirement - 06

Identifier	FR-06
Title	Create Ride (Driver)
Requirement	Choose role of a driver
Source	Supervisor, M. Shoaib Malik
Rationale	Select riders from list
Restrictions and Risk	Driver have to choose only certain number of riders according to free seating capacity.
Dependencies	Firebase server, Google Map Api
Priority	Medium

2.5.1.7 Table 7: Functional Requirement - 07

Identifier	FR-07
Title	End Ride (Driver)
Requirement	Driver ends the ride.
Source	Supervisor, M. Shoaib Malik
Rationale	To make sure all riders dropped.
Restrictions and Risk	Nill
Dependencies	firebase server, Google Api
Priority	Medium

2.5.1.8 Table 8: Functional Requirement - 08

Identifier	FR-08
Title	Book Ride (Rider)
Requirement	Choose role of a rider.
Source	Supervisor, M. Shoaib Malik
Rationale	Select destination and pickup point
Restrictions and Risk	Rider have to choose only available pickup point.
Dependencies	Firebase server, Google Map Api
Priority	Medium

2.5.1.9 Table 9: Functional Requirement - 09

Identifier	FR-09
Title	Ride End (Rider)
Requirement	Driver pick the rider
Source	Supervisor, M. Shoaib Malik
Rationale	To make sure that driver drops a rider to destination.
Restrictions and Risk	Nil
Dependencies	Firebase server
Priority	Medium

2.5.2 Non-Functional Requirements

2.5.2.1 Table 10: Non-Functional Requirement - 01

Identifier	NFR-01
Title	User Authentication
Requirement	User must be registered via university email id

2.5.2.2 Table 11: Non-Functional Requirement - 02

Identifier	NFR-02
Title	Real time location tracking
Requirement	Mobile phone must be provide accurate GPS location of device.

2.5.2.3 Table 12: Non-Functional Requirement - 03

Identifier	NFR-03
Title	Multi-user system
Requirement	Efficient use of the system when the user increases.

2.5.2.4 Table 13: Non-Functional Requirement - 04

Identifier	NFR-04
Title	Internet connection
Requirement	User device must have a internet connection.

2.5.2.5 Table 14: Non-Functional Requirement - 05

Identifier	NFR-05
Title	Device compatibility
Requirement	Use of latest APIs and application must support most of android phone versions

2.5.2.6 Table 15: Non-Functional Requirement - 06

Identifier	NFR-06
Title	User friendly application
Requirement	Easy interface of application, not makes a user to think twice.

2.5.2.7 Table 16: Non-Functional Requirement - 07

Identifier	NFR-07
Title	Application limited to university students
Requirement	Only students with university email able to use bus system

2.6 Use Cases

2.6.1 Use Case Diagram

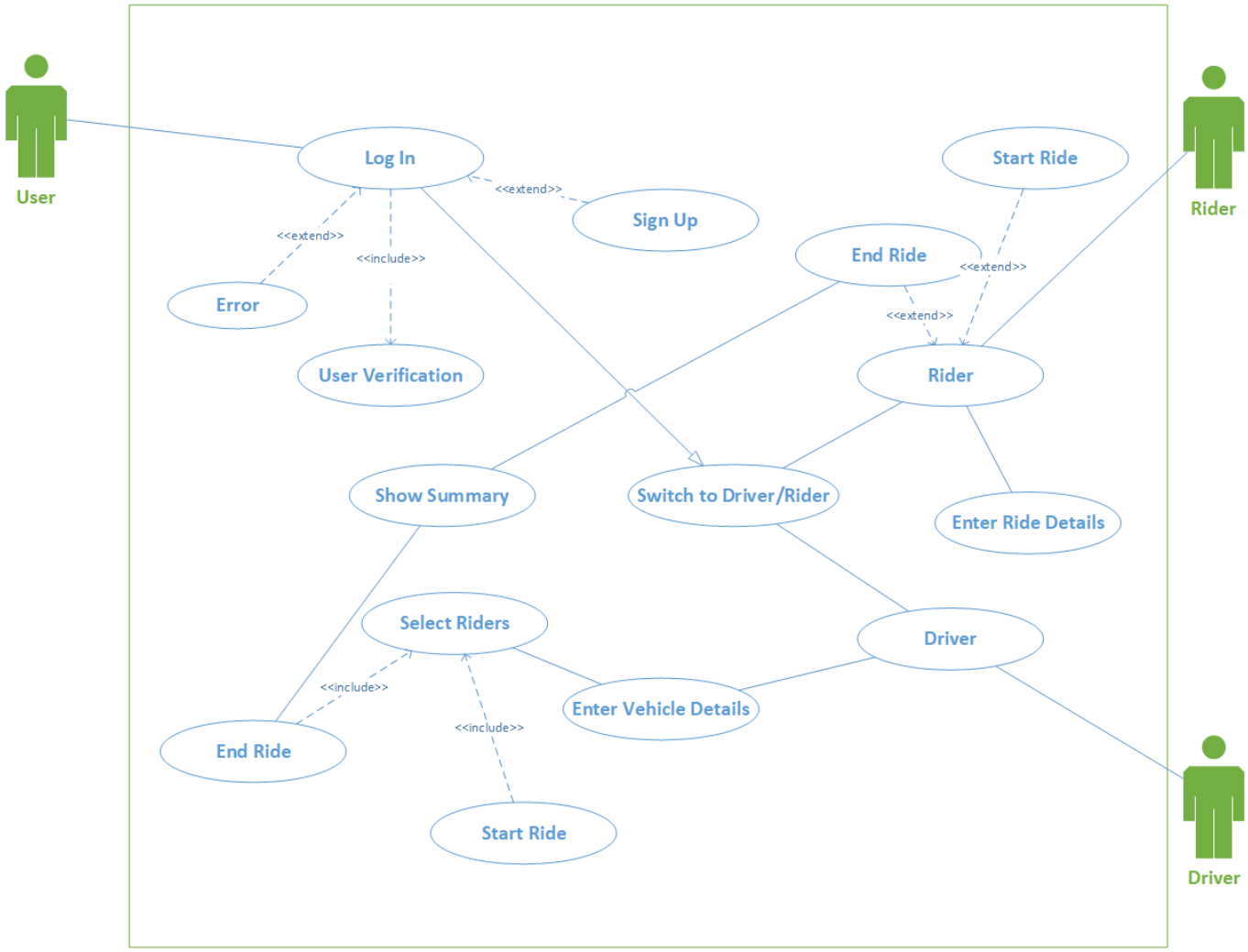

Figure 2.1: Use Case Diagram

Chapter 3

Requirement Specifications

In this chapter, first describe the existing system, its limitations or drawbacks and then explain how the new or proposed system will overcome these problems. This should then be followed by complete requirements specification for the proposed system. Describe the behavior of the system to be developed and include a set of use cases that describe interactions the users will have with the system. In addition also describe non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints). Should have the following headings:

- Existing System
- Proposed System
- Requirement Specifications
- Use Cases

Chapter 4

Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. This chapter should have the following sections:

4.1 System Architecture

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable.

4.2 Design Constraints

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made during the developing the system design.

4.3 Design Methodology

Summarize the approach that will be used to create and evolve the designs for this system. Cover any processes, conventions, policies, techniques or other issues which will guide design work. This is for deciding whether you will use structured, object-oriented or other specific methodologies. Most people will use some object-oriented technique with UML.

4.4 High Level Design

This section describes in further detail elements discussed in the Architecture. High-level designs are most effective if they attempt to model groups of system elements from a number of different views. Typical viewpoints are:

1. Conceptual or Logical: This view shows the logical functional elements of the system. Each component represents a similar grouping of functionality. For UML, this would be a component diagram or a package diagram.
2. Process: this view is the runtime view of the system. The components are threads or processes or distributed applications. In UML, this would be a process interaction diagram.
3. Physical: this view is for distributed systems. The components are physical processors that have parts of the system running on them. For UML, this would be a deployment diagram.
4. Module: this view is for project management and code organization. The components are typically files or directories. This picture shows how the directory structure of the build and development environment will be designed.
5. Security: this view typically focuses on the components that cooperate to provide security features of the system. It is often a subset of the Conceptual view.

4.5 Low Level Design

This section provides low-level design descriptions that directly support construction of modules. Normally this section would be split into separate documents for different areas of the design. For each component we now need to break it down into its fundamental units or modules. For an OO implementation in Java, our components would become packages. Then the low level design will take each package and break it down into its classes. For smaller systems, you may have a single UML class diagram that each module description refers to.

4.6 Database Design

The section should reveal the final design of all database management system (DBMS) files and the non-DBMS files associated with the system under development. Provide a comprehensive data dictionary showing data element name, type, length, source, validation rules, maintenance (create, read, update, delete capability), data stores, outputs, aliases, and description.

4.7 GUI Design

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels.

4.8 External Interfaces

External systems are any systems that are not within the scope of the system under development. In this section, describe the electronic interface(s) between this system and each of the other systems and/or subsystem(s), emphasizing the point of view of the system being developed.

Chapter 5

System Implementation

Implementation is the process of moving an idea from concept to reality. The System implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment.

5.1 System Architecture

Describe the architecture e.g. in terms of: System internal components, Functionality of the components, Communication between the components Tools and Technology Used Development Environment/Languages Used Processing Logic/Algorithms Application Access Security Describe new application access related security measures, e.g. in terms of: Security Zones/Firewalls, Encryption, Authentication, e.g. Account & Password structures and rules, Authorization, e.g. operator rights and roles, authority handling, Auditing / Access Logging, Safe Data Storage Database Security

Describe new DB related security measures, e.g. in terms of: Remote Access, Authentication (Account & Password: structure, rules), Authorization (rights/roles, handling), Anonymous and Group Users, Auditing/Logging (events, data, log handling).

Chapter 6

System Testing and Evaluation

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. Be warned that many projects fall down through poor evaluation. Simply building a system and documenting its design and functionality is not enough to gain top marks. It is extremely important that you evaluate what you have done both in absolute terms and in comparison with existing techniques, software, hardware etc. This might involve quantitative evaluation and qualitative evaluation such as expressibility, functionality, ease-of-use etc. At some point you should also evaluate the strengths and weaknesses of what you have done. Avoid statements like "The project has been a complete success and we have solved all the problems associated with ...! It is important to understand that there is no such thing as a perfect project. Even the very best pieces of work have their limitations and you are expected to provide a proper critical appraisal of what you have done. The following are different types of testing that should be considered during System testing:

- Graphical user interface testing
- Usability testing
- Software performance testing
- Compatibility testing
- Exception handling
- Load testing
- Security testing
- Installation testing

For research based projects this chapter should include complete description of evaluation metrics and analysis/discussion of evaluation results.

Chapter 7

Conclusions

The project's conclusions should list the things which have been learnt as a result of the work you have done. For example, "The use of overloading in C++ provides a very elegant mechanism for transparent parallelisation of sequential programs". Avoid tedious personal reflections like "I learned a lot about C++ programming..." It is common to finish the report by listing ways in which the project can be taken further. This might, for example, be a plan for doing the project better if you had a chance to do it again, turning the project deliverables into a more polished end product.

Appendix A

User Manual

Appendices are provided to give supplementary information, which is included in the main text may serve as a distraction and cloud the central theme.

- Appendices should be numbered using alphabets, e.g. Appendix A, Appendix B, etc.
- Tables and References appearing in appendices should be numbered and referred to at appropriate places just as in the case of chapters.
- Appendices shall carry the title of the work reported and the same title shall be written in the contents page.

References

[1] Peter J. Denning. Is computer science science? *Commun. ACM*, 48(4):27–31, April 2005. No Citations.