Airwhare – Wireless Tour Guide

BEng Group Project

Dirk Holshausen dth02
Andy Li al202
Satchitanand Patel sp2502
James Turner jet02

Abstract

Location-enhanced applications are emerging as an important feature of mobile computing. There is wide spread use for this type of application allowing for massive growth in this area in the near future. However for this technology to be widely adopted it needs to be cheaper than existing technologies such as GPS, and readily available. In answer to this we have developed AirWhare, a tour guide of the Imperial College campus that is built using Intel's Place Lab software. We have used Place Lab to listen for the unique MAC addresses of 802.11 wireless access points, referencing their position in a database. This allows users to estimate their current position on a laptop or PDA while also providing them with relevant information pertaining to their area. Through testing we show that we can achieve an average accuracy of 20-50 meters with almost complete coverage of the campus.

Acknowledgements

Professor Morris Sloman m.sloman@imperial.ac.uk - our supervising professor.

James Howard jamesh@cs.washington.edu - for help on questions posted on www.sourceforge.net

Yatin Chawathe yatin.chawathe@intel.com - for help on questions posted on www.sourceforge.net and help through direct email correspondence.

Brendan Maguire b.maguire@imperial.ac.uk - for help connecting our PDA's to the college network.

Peter Gillings p.gillings@imperial.ac.uk - for providing information and images relating to Imperial College, and for explaining the Tanaka touch screen information systems.

Matthew Balyuzi m.balyuzi@imperial.ac.uk - for providing us with the geographical location of the imperial network's access points.

James R Moody jrm198@doc.ic.ac.uk - for providing us with the geographical location of the DoC access points.

Contributors

Some of the material contained in this document has been taken from documentation on the Placelab web site, some of the main contributors of this material being:

Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello and Bill Schilit.

Table of Contents

1 Introduction

The following is the third and final report for Airwhare. This is a system that will allow people to navigate around the Imperial College campus using a mobile PC that acts as a tour guide.

This is the culmination of three months work in which our four team members created a novel product from scratch using untested technologies in new ways. The report will explain in detail the total development of the system from concept to completion and testing.

1.1 The Brief

The exact remit can be found at:

www.doc.ic.ac.uk/~ih/gp/proposals/mss2.html

The finished product should be a system capable of supplying the user with relevant information pertaining to an area of the Imperial College South Kensington Campus. This information will be accessed via a mobile PC and should be include information for the whole campus. The system should be able to act as a fully capable tour-guide allowing those unfamiliar with the campus layout to fully navigate it.

1.2 About the Group

Our group consists of four members:

Sachit Patel	sp2502@doc.ic.ac.uk	
Andy Li	al202@doc.ic.ac.uk	Secretary
Dirk Holshausen	dth02@doc.ic.ac.uk	
James Turner	jet02@doc.ic.ac.uk	

Our project supervisor is Professor Sloman mss@doc.ic.ac.uk

1.3 A Word on the Previous Reports

Two previous reports have been produced detailing the development of Airwhare. These are the *Project Specification* and the *Interim Report* which can be viewed at: https://www.doc.ic.ac.uk/project/2004/362/g0436240B/

Although this final statement has been written independently and must be treated as so, further information particularly concerning the earlier stages of development may be sought in these two reports.

The Project Specification - submitted 22nd October 2003 details our initial plans for the system before any development had occurred.

The Interim Report – submitted 12 November 2004 is an account of our progress half way though the development of Airwhare. It is therefore more detailed and relevant than its predecessor and provides a natural prologue to this final report.

Much of the relevant detail in the previous reports have been included here for your convenience.

2 Background information

We began the project by compiling a dossier of information relevant to our planned system. This would allow us to approach the development stages with a sound understanding of the salient concepts involved whilst considering pre-emptive solutions to problems we may encounter.

Principally our researched focused on similar systems which had already been, or were in the process of being developed and the technologies which we were considering implementing.

2.1 Existing Solutions

Whilst creating our own solution to this specification we thought it wise to make an analysis of other products that perform similar tasks. We were aware that there would be no system which would meet out specifications completely; however we were keen to build up as detailed a practical overview of the project as possible. To this end we researched several products that shared particular faculties, required in our system:

1. Temples of Taipei: Handheld Tour Guide for Mobile Devices

This is a mobile tour guide for the Buddhist temples in Taipei, the capital city of Taiwan.

Figure 2.1.1 – the temples of Taipei PDA based tour guide
This is one of the best looking products we have researched, using Macromedia Flash to produce the user interface. It also includes several interesting and useful features, such as video clips and scrolling panoramic scenes to allow tourist familiarize themselves with an area.

This product is exceptional in its ease of us with its large and uncluttered interface and scrollable map. These are two facilities we would do well to incorporate in our system. However, this product does not include a positioning service, which would enable tourist to navigate these sites with greater ease.

2. Pocket Tour Guide 2.2004: A mobile guide to Israel

This is a mobile tour guide that covers thousands of tourist destinations in Israel.

Figure 2.1.2 – the PDA based tour guide to Israel

This is not a very attractive system – created in HTML it is reminiscent of an antiquated webpage. It does however implement a Global Positioning System (GPS) to allow its users to locate where they are in the country with some accuracy. It is also loaded with information from regional maps to restaurant menus and bus timetable. The versatility of this product would make it available to a wide group of potential users and increase the interest and effectiveness of the product.

2.2 Other Technologies available to us

The idea of using a PDA as a wireless tour guide is a relatively new one, but even so, there are a few examples on the Internet where people have started investigating the possibilities. As with any new technology or idea, there are many different ways to approach the problem. Some groups have used tablet PCs instead of PDA's, and they have used "infra-red" or "Bluetooth" as opposed to "Wi-Fi".

One of the main uses for "infra-red" is in a gallery, where the user can point the device at an exhibit and get more information about the item, the artist, etc. (http://www.hci.cornell.edu/projects.php?id=3 *accessed 20/12/04*). Infra red is a good idea for this sort of problem as it enables the user to be very specific about what he/she is interested in. This enables the user to be selective with the data available.

Another of the uses is in major towns and cities, where GPS doesn't work due to the high buildings. (http://www.newscientist.com/article.ns?id=dn6058 *accessed 19/12/04*). New Scientist suggests

"…emergency services in particular could find the system an essential back-up"

If used in conjunction with GPS it can take over in the exact situations that GPS fails: in cities with many tall buildings. This is because it is likely that these buildings, which are shielding the satellite signals, are also full of wireless access ports. These can be used to show the user's location in the city by triangulating the signals from nearby points, and uses the relative signal strengths to determine the position. It is currently not as accurate as GPS:

"[Wi-Fi] …can provide accuracy to within 20-30 metres, whereas the GPS average is 8 to 10 metres.

But with improved algorithms that take into account, say, the height of the base station above the ground, or the building materials in the vicinity, LaMarca says 'We could get on par with GPS' in an area as densely populated as downtown Seattle"

It is in the above situation where Wi-Fi is showing that it can be a viable option as a tourguide, especially as it is inexpensive, the infrastructure is already in place (or is growing rapidly) and it is very portable.

Right is a picture of the handheld PC that is offered by the Tate modern as a trial (http://news.bbc.co.uk/1/hi/technology/2225255.stm *accessed 21/12/04*)

"When you enter a gallery room, the hi-tech guide flashes to indicate that more information is available. You can then click for more information".

That could be a section of video describing the work or talking about aspects of it, or it could be a page of text, giving further information about the artist."

(a) Exhibit picture (d) Video playback

Figure 2. Exhibit illustrations

(http://csdl.computer.org/comp/proceedings/wmte/2004/1989/00/19890195.pdf *accessed 2/1/05*). This museum in Taiwan is using PDAs to show videos of dinosaurs to children to promote learning.

3 The System Features

The following lists a summary agenda for the features we wished to include in our project. These were compiled, based on our initial specification and details acquired whilst researching similar systems.

The compiled list was then broken down into two priority groups - the first consisting of the attributes that we considered essential for the system to operate competently. Second importance was given to all those features that we desired to incorporate as additional features to improve the usability of the system or its technical excellence.

3.1 Minimum Requirements

- The system must act as a guide to the university campus, thereby holding relevant information to key areas of interest on the campus
- The system must hold a graphical representation of the campus thereby allowing a user to easily select information relating to their position
- The system must include locating functionality in order to track the user's position over map. This must operate over a minimum of ten access points for a basic accuracy.
- The system must operate on a mobile PC of the maximum weight of a laptop

3.2 Extensions

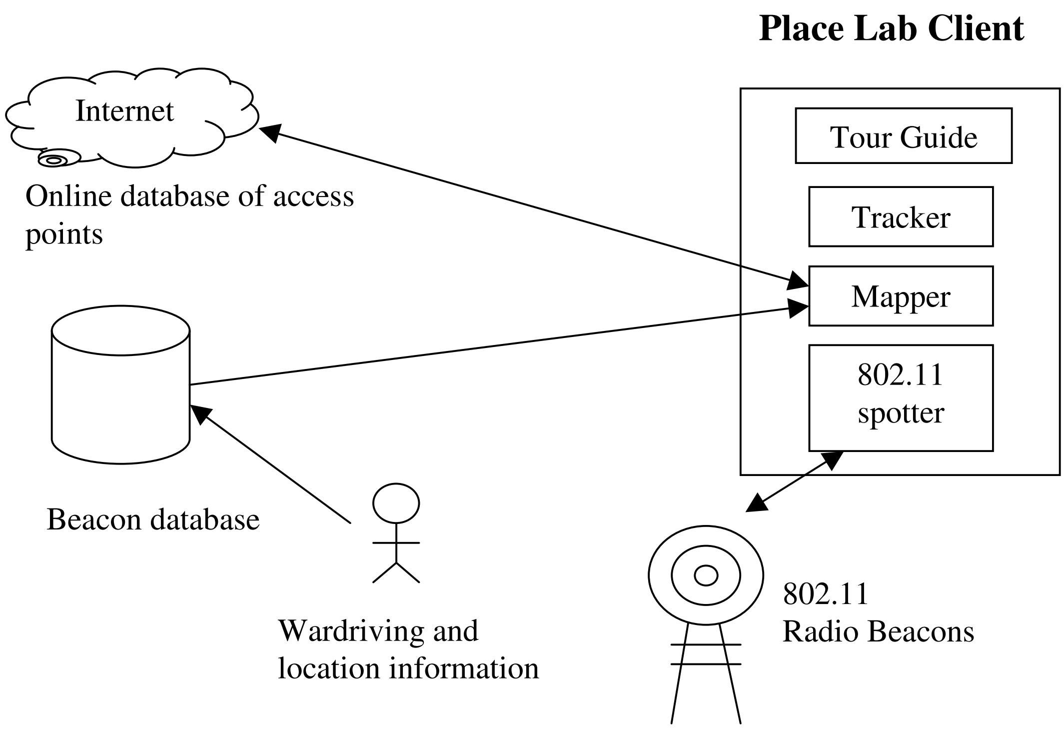
- The system must operate on a mobile PC of the maximum weight of a palmtop
- The tracking map must have high enough resolution to accurately show the user's position.

4 A Word on Organisation

A failing of many teams when approaching such a project is a poor sense of organisation and as a result a poor coordination of the work. Symptoms of such erratic activity would be an imbalanced workload amongst team members, vital work being left undone and resentment running throughout the group. We were keen to avoid all such problems and therefore made a point of regularly meeting to discuss what jobs were being done and what tasks were left to do. Our routine of two formal meetings a week (with informal discussions on most days) allowed us to organise ourselves efficiently, gauging the workload on each member and distributing further work equitably. This constant communication also allowed us to appreciate who was most skilled in each area, and which tasks suited them best. A log of our group meetings may be viewed in Appendix 6.

However it would also important for us to maintain flexibility in the team. A task was often allocated to a team member before the total time required of him could be appreciated. Often jobs would be more complex than anticipated, or require expertise unavailable to the team. Our ability to communicate quickly and effectively has allowed us to spontaneously change direction, consider other technological options available and complete the project in the most efficient way – wasting as little time and resources as possible.

5. Architecture Overview



5.1. File Structure

The following shows the file structure of \placelab-win32\run\

FILE	BRIEF EXPLANATION
MapLoaderDT.bat	Calls the PHP script online which loads the database mapper.db
MapDemoDT.bat	Loads map.wad, which opens the standard map.
MapMain.bat	Combines the above two files into one batch file.
MapLoaderAerialDT.bat	Takes imperialapsaerial.txt as an argument loading the database mapper.db
MapDemoDTAerial.bat	Loads mapaerial.wad, which opens the aerial map.
MapAerial.bat	Combines the above two files into one batch file.
imperialapsmapped.txt	Flat text files holding access point information.
imperialapsaerial.txt	

The following shows the file structure of \placelab-win32\placelabdata\data.

FILE	SUB-FILES	BRIEF EXPLANATION
Mapper.db	NA	Database of the known access points
map.wad	apcache.txt defaults.txt applaceset.meta map_queenstower.gif maps.index map_queenstower.meta	
mapaerial.wad	apcache.txt defaults.txt applaceset.meta map_aerial.gif maps.index map_aerial.meta places.index place1.txt to place10.txt num1.gif to num27.gif	
imperial.log	NA	Used for testing to simulate moving around the campus.

Further explanation of file structure of \placelab-win32\placelabdata\data

Hashtable Wads are zip files with the following structure:

```
+ map.wad
        - apcache.txt (a text file of the APs in the form: latitude longitude SSID BSSID)
        - defaults.txt (a Sectioned File described below)
        + maps
            - maps.index (a Sectioned File described below)
            - mapname.jpg/gif/png
            - mapname.meta (a Sectioned File described below)
        + places
            - places.index (a Sectioned File described below)
            - place-set-name.txt (a Sectioned File described below)
            - icons
                - num1.jpg/gif/png
```

maps.index has the following structure:

```
[Maps]
mapname=mapname.meta
```

mapname.meta has the following structure:

```
[Map]
# origin_lat and origin_lon are lower left corner values
```

origin_lat=47.349829
origin_lon=-122.98327
pixels_per_lat=100.01
pixels_per_lon=9994.39828
# OR you may specify the upper right corner's lat and lon
upper_right_lat=47.449829
upper_right_lon=-122.99327

image=images/mapimage1.jpg/gif/png

Without the use of GPS devices to plot the actual latitude and longitudinal values of our maps we have had to set up our maps on a basic coordinate system with (0,0) being the bottom left corner. This makes it more time consuming to plot all the access points on the map as each access point has to be entered into the database with its expected relative latitude and longitude value and then the map has to be loaded to see if the access point has been plotted in the correct position.

places.index has the following structure:

```
[Places]
place-set-name=place-set-name.txt
```

place-set-name.txt has the following structure:

```
[place name]
# image references are relative to the root of the wad
icon_file=places/icons/num1.jpg
lat=46.19829
lon=-122.8132
text=Some descriptive text
url=http://www.doc.ic.ac.uk
type=radical/AP/etc

[other place name]
etc ...
```

defaults.txt has the following structure:

```
[Map]
mapname=1

[Places]
placeset1=1
placeset2=1
placeset3=1
# ... and as many place sets as you would like to specify
```

The defaults file specifies what the wad author thinks should be the first set of things that people might want to see. There is no guarantee that they will be the first things displayed, it is only a suggestion. Also note that while you must specify a map if you have a defaults file, you needn't specify any place sets. Defaults files are not required. Note that while place set definition file references in *places.index* are relative references, the image file references actually in the place set files are relative to the root of the wad. This is done so that people can put their images wherever they want in the wad, since ../ type references are not allowed.
(*With reference from http://placelab.org/toolkit/doc/javadoc/org/placelab/demo/mapview/WadData.html*)

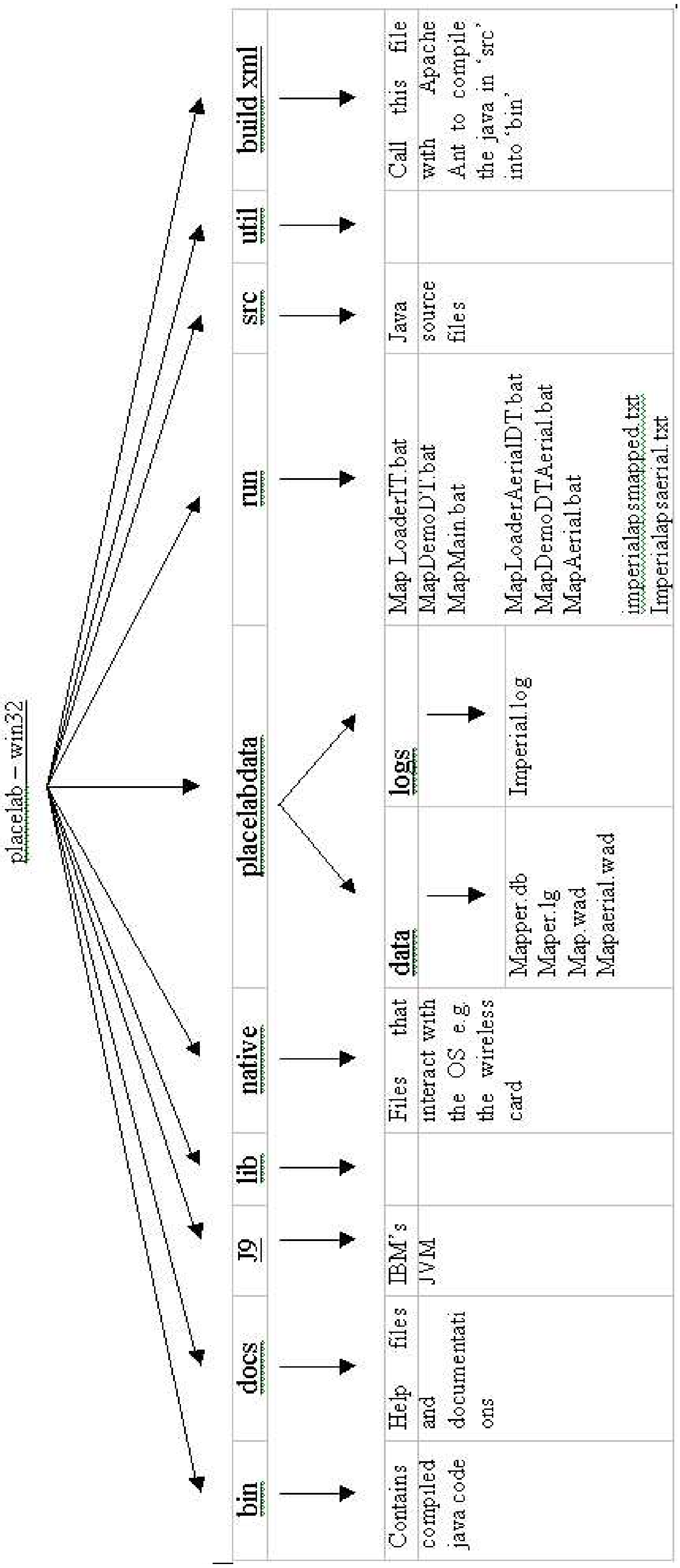
The software is invoked in the following manner:

The file *MapLoaderDT.bat* calls *org.placelab.mapper.MapLoader* which reads the file *imperialapsmapped.txt* and creates the file *mapper.db*. The department responsible for the maintenance of the College network supplied the details of the file *imperialapsmapped.txt.* To load the database with the access points for the aerial map the file *MapLoaderDTAerial.bat* is run which calls the same file but reads *imperialapsaerial.txt.*

The file *MapDemoDT.bat* reads in the two files *map.wad* (or *mapaerial.wad* if the aerial map is loaded) and *mapper.db*. All access points held in *mapper.db* are plotted on the relevant map. Place Lab will then compare these access points to those it can currently detect in the area. The access points detected are then used to provide an estimation of the user's current position which is shown as a red circle on the map. *MapDemoDTAerial.bat* reads *mapaerial.wad* to load the aerial map.

Using two maps for our project provides us with a small problem as the maps do not exactly 'overlay' one another. For example on our coordinate system (5,9) may be the library on the main map but could be the Computing department on the aerial map. This means that we need two databases of access points depending on which map is loaded. So the user is not aware of this problem we have combined *MapLoaderDT.bat* and *MapDemoDT.bat* into a single batch file *MainMap.bat*. Similarly *MapLoaderAerialDT.bat* and *MapDemoDTAerial.bat* have been combined into *MapAerial.bat.*

5.2 Place Lab File Structure



6. Developing the System

6.1 Implementing PlaceLab

With the exception of the small amount of native code that is written for each spotter, Place Lab is written entirely in Java 2 Micro Edition (J2ME). Place Lab currently runs on a number of platforms including Windows XP, Linux, OS X, Pocket PC 2003 and Symbian cell phones providing support for spotting 802.11(a,b&g), GSM and Bluetooth beacons. The Place Lab APIs for spotters, mappers, and trackers are consistent across all platforms assisting developers in porting their applications to different platforms, *e.g.* as we have done from a full-featured Windows laptop to an iPAQ h5500 Pocket PC. [1]

6.1.1 Wardriving

Wardriving is the gathering of statistics about wireless networks in a given area by listening for their publicly available broadcast beacons. Wireless access points (APs) announce their presence at set intervals (usually 100 milliseconds) by broadcasting a packet containing their service set identifier (SSID) and several other data items. A *stumbling utility* running on a portable computer of some sort (a laptop or PDA) listens for these broadcasts and records the data that the AP makes publicly available.

Wardriving usually entails driving around in your car while running a stumbling utility, and recording beacons from nearby APs although you can just as effectively simply walk around with a Wi-Fi equipped PDA in your pocket. Most stumbling utilities have the ability to add GPS location information to their log files, so that the geographical positions of stumbled APs may be retained and plotted on electronic maps.

Place Lab can use mapping data from the large databases produced by the *war-driving* community. By pooling their wardrives together and applying some simple averaging, these groups have produced estimated locations for millions of beacons. Public domain wardriving software has been developed for most computing platforms, and there are many aggregation websites to which wardrives can be submitted. While war-driving has traditionally been performed in order to provide information about where nearby network access can be obtained, Place Lab uses these maps in reverse to infer where we are given a particular network is nearby. Since the positions of beacons are being inferred from observations tied to GPS estimates, war-driving databases only contain estimates of beacon positions. The error in these estimates translates into a decrease in the accuracy of location estimates made by Place Lab. However, what these databases lack in accuracy they make up for in coverage making them highly useful for Place Lab. As an example, wigle.net is the largest of the 802.11 war-driving repositories, and contains over 2 million known AP positions, and the recent "World Wide War-drive" added 275,000 new access points over an 8 day period (worldwidewardrive.org). [1] + [3]

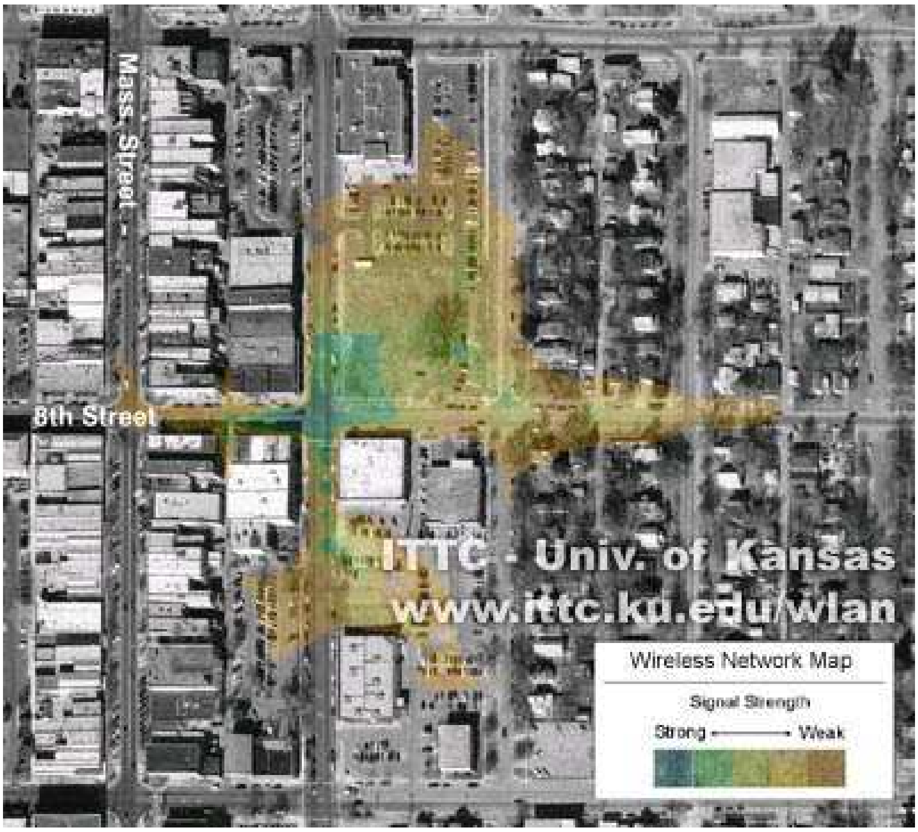
Different stumbling utilities were tested to see which would be most suitable for our project. These included:

- Access Point Hunter - can find and automatically connect to whatever wireless network is within range. It can be used for site surveys, writing the results in a file.
- Ethereal - free network protocol analyzer for Unix and Windows. It allows you to examine data from a live network
- NetStumbler - Windows Utility for 802.11b based Wireless Network Auditing.
- MiniStumbler - Network Stumbler for Pocket PC 3.0 and 2002.
- Wififofum - an 802.11 scanner designed for PDAs running PocketPC 2003
- PocketWarrior - Wi-Fi Surveying Tool for the Pocket PC. Supports GPS.

Other network stumbling utilities were briefly looked at but not considered as suitable for our project. The most practical way of 'wardriving' around the college campus was to walk around using a Wi-Fi equipped PDA. We used Wififofum (www.wififofum.org) on the PDA to scan and then save logs of access points that we had detected. NetStumbler was used on a Windows laptop as a backup and to check we were getting the correct readings from our PDA. Without a GPS we needed some way of adding location information to each access point so it could be plotted on our maps of the college campus. Matthew Balyuzi, a Network Technology Analyst at Imperial's Information and Communication Technology (ICT) department kindly provided us with the geographical location of each access point. This information could now all be combined in our database and plotted on our maps.

6.1.2 The Network

We will be using Imperial College's Wireless network to estimate our position using the Place Lab software. We have a list of over 80 access points maintained by Imperial around the South Kensington campus. Due to the need to demonstrate our tour guide we have concentrated on obtaining access points around the Queens Tower in the centre of the campus and along Prince Consort Road on the north of the campus. Concentrating on these areas allows us to provide a more accurate representation of where the user is. 802.11b wireless access points should produce a circular footprint with a radius of about 100m. However due to many factors including the positioning of the access point, nearby buildings and structures, adjacent open areas, and even the weather, the footprint from an access point is often far from circular. This means that you may not detect an access point in a neighbouring building but may detect another from all the way down a street. This is illustrated in the figure below.



The above picture is from http://www.ittc.ku.edu/wlan/index.shtml (accessed 28/12/04) and shows how the wireless signal is greatly affected by the nearby buildings.

The tracking algorithm we are using averages where the user is, taking into account all access points currently detected that have location information stored in the database. The more access points we hold in our database, the more accurate the user's estimated position will be. This is especially useful when picking up 'one off' signals from access points that may be relatively far from the users current location. e.g. if the user picks up 10 access points in close proximity and then another signal from further away, the user's current location remains around the 10 access points.

To use our tour guide system on either a laptop or PDA the user must connect to the Imperial network. At present this means connecting through a Virtual Private Network (VPN) and providing a user name and password. To connect to the VPN users have to register and then wait 24 hours for their account to be activated. A guest user name and password could be issued for users visiting the college. This account could expire after 24 hours or have other security features. To connect using a Pocket PC users need to install the Imperial College Certificate Authority certificates onto the device. These are used to ensure that you are connecting to the Imperial Wireless Network and not another network purporting to be the Imperial Network.

6.1.3 The Map

We will be using two maps for our tour guide system. The standard map gives a graphical representation of the campus showing numbered buildings, wheelchair access, the main entrance and surrounding road names. The other map is an aerial photograph of the campus. Allowing users the option of having both maps will hopefully provide them with a better idea of where they currently are.
Identifying numbers have been overlaid on the aerial map which users can click on to see more information about that building. These numbers correspond to the same building numbers on the main map.

6.1.4 Implementation Language Choices

We have four languages in which we can implement our user interface:

- HTML
- Java, specifically Java 2 Micro Edition (J2ME)
- C# (Visual Studio .NET)
- VB.NET

The best language to use on the laptop may not necessarily be the most suitable language for the PDA. These are discussed in detail below.

HTML (Hypertext Mark-up Language)

This is the coded format language used for creating hypertext documents on the World Wide Web and controlling how Web pages appear.

Laptop:

This solution would be the simplest to implement. We are all familiar with the language; there is lots of documentation associated with HTML and there are plentiful examples of good and bad website design. Using a browser such as Internet Explorer we could run the HTML pages in kiosk mode which removes all toolbars expanding the page to fill the screen. This is the approach they have used in the Tanaka Business School entrance for their touch screen information systems.

Pocket PC:

It can be difficult to design a ‘clean’ interface that is uncluttered and that users can actually read using HTML. Kiosk mode cannot be used on a Pocket PC. However users unfamiliar with PDAs may feel more at home using Internet Explorer, a browser they have used on their desktops.

Java

An object-oriented programming language developed by Sun Microsystems. It was specifically designed for the distributed environment of the Web and can be used to create applications that can run on a single computer or distributed among several computers in a network. Java offers two entirely different user interface libraries, the Abstract Windowing Toolkit (AWT) and Swing, for building client-side interfaces. Swing is a sophisticated user interface library that can mimic virtually every feature in traditional native code programs. Again, we are all experienced Java developers (but not very experienced with the AWT and Swing libraries),

Laptop:

The language allows us enough flexibility to implement features that HTML would restrict us to. As the Place Lab software is written predominantly in Java it makes sense to continue using Java for the user interface.

Pocket PC:

Java is not so straightforward to use on the Pocket PC as the Pocket PC does not come with any Java Virtual Machine (JVM) although it is possible to install a JVM. Jeode from http://esmertec.com and NSICOM CRÈME from http://www.nsicom.com were both tested and seem to be solid VM implementation. However running Java using these JVMs proved to be difficult and small problems kept on holding us up. Without native support for Java this is not a simple option. Place Lab comes with its own VM. IBM's j9 is used so that Place Lab's Java code will run on a Pocket PC. However this is a compact version of j9 and does not support the graphical libraries needed to design an interface in Java on the Pocket PC.

C#

The C# language is an object-oriented language that is aimed at enabling programmers to quickly build a wide range of applications for the Microsoft .NET platform. The goal of C# and the .NET platform is to shorten development time by freeing the developer from worrying about several low level plumbing issues such as memory management, type safety issues, building low level libraries, array bounds checking , etc. thus allowing developers to actually spend their time and energy working on their application and business logic instead. In C#, the new Windows Forms is the library of choice for building sophisticated client applications. However, some may argue that C# is “"a short description of the Java language and platform" if the words C# and the .NET platform were replaced with words Java and the Java platform.

Laptop:

C# is similar enough to Java for us to feel competent enough to program in it. Microsoft Visual Studio provides us with a development environment that is relatively easy to use, with a large amount of documentation available both on and off-line providing help if required.

Pocket PC:

.NET is not just for desktop programming. It also allows us to build Pocket PC applications using C# or VB.NET. Emulators are provided so that there is no need to upload the code onto the Pocket PC after every small alteration just to test it.

VB.NET

Laptop:

Visual Basic is another .NET language that we could use. Although Visual Basic does not appear to be a difficult language, no one in the group has any direct experience in it.

Pocket PC:

VB.NET becomes a viable option on the Pocket PC. Along with C# it has been designed to be used to develop applications on the Pocket PC.

Obviously, HTML would not provide anywhere near the levels of features we could implement using Java or C#. We decided to choose C# Visual Studio .NET over Java and VB.NET. Java and C# are similar not only in syntax, but in architectural language design decisions, both languages were object orientated, and both gave great flexibility. The area where C# was strongest in (in our opinion), was that in using Visual .NET, it allowed us to produce graphical user interfaces more quickly and easily. VB.NET was a contender here, especially due to its suitability for Pocket PC development, but C# provided similar suitability and we were more familiar with the syntax. We found that Visual Studio provided a much cleaner and better visual environment to develop GUIs, and removed the burden of learning Java's Swing and AWT class libraries.

6.2 The Place Lab Architecture [1]

The Place Lab architecture consists of three key elements:

1. Radio beacons in the environment,
2. databases that hold information about beacons' locations,
3. Place Lab clients that use this data to estimate their current location.

These subsections are described in more detail below.

6.2.1 Radio Beacons

Place Lab works by listening for the transmissions of wireless networking sources like 802.11 access points, fixed Bluetooth devices, and GSM cell towers. We collectively call these radio sources *beacons.* They all employ protocols which assign beacons a unique or semi-unique ID. Hearing this ID greatly simplifies the client's task of calculating their position. For our project we will be using 802.11 access points to calculate our position. These access points have been plotted on the map of Imperial College by using their unique MAC addresses.

With over 100 Imperial College owned access points in the relatively small area of the South Kensington campus, our Place Lab client will be able to estimate its position fairly accurately. On 'stumbling' around the college many more private access points were seen. If required these access points could be added to our database to further increase the accuracy of our Place Lab client.

Place Lab devices need only interact with radio beacons to the extent required to learn their IDs. Place Lab clients do not need to transmit data to determine location, nor do they listen to other user's data transmissions. In the case of 802.11, receiving beacons can be done entirely passive by listening for the beacon frames periodically sent by access points. These beacon frames are sent in the clear, and are not affected by either WEP or MAC address authentication.

6.2.2 Beacon Databases

Imperial College and the Department of Computing know the locations of their 802.11 access points since this information is commonly recorded as part of a deployment and maintenance strategy. These data sets tend to be large, and the maps are typically quite accurate. While these data sets were not originally built for doing beacon-based location estimation, it only requires a format-translation step to add this data to Place Lab and location-enable the institution's building or campus.

The database is a flat text file with four tab-separated columns:

latitude longitude SSID BSSID

Running org.placelab.mapper.MapLoaderDT with the filename as an argument loads the text file into the mapper.db database.

Our Place Lab client relies on the availability of Imperial's 802.11 access point locations. Spotting a beacon does not improve the location estimates of our client. This access point needs to have its relevant location information stored in our database. In our architecture, the *beacon database* plays the important role of serving this beacon location information to client devices. We hold databases in two locations, one held on a central server and the other cached locally on the device. Both forms have their individual advantages and disadvantages. The centrally stored database is accessed by our client, which runs a PHP script to download all the access points.

This allows us to update the database only in one place rather than on each separate device. This is used as the default option for our Place Lab client on the laptop.

We use the other option on the PDA, caching a local database. This saves the user the added step of having to connect to the Internet to allow the software to download the access point information. While connecting to the internet is often simple on a laptop, connecting to the internet on a PDA and maintaining that connection while roaming the campus can be difficult or confusing for users unfamiliar with PDAs.

6.2.3 Place Lab clients

The Place Lab clients use live radio observations and cached beacon locations to form an estimate of their location. To make client both extensible and portable, client functionality is broken into three logical pieces: spotters, mappers and trackers.

Spotters are the eyes and ears of the client and are responsible for the observing phenomenon in the physical world. Place Lab clients typically instantiate one spotter per radio protocol supported by the device. The spotter's task is to monitor the radio interface and share the IDs of the observed radio beacons with other system components. An observation returned by a spotter is of little use if nothing is known about the radio beacons.

The job of the *mapper* is to provide the location of known beacons. This information always includes a latitude and longitude, but may also contain other useful information like altitude, age of the data or power of the transmitter. Mappers may obtain this data directly from a mapping database, or from a previously cached portion of a database. This cache could contain beacons for a large area, say the entire United States and Europe, or may as in our case, due to capacity concerns, just contain information relating to Imperial College's South Kensington campus.

The *tracker* is the Place Lab client component that uses the streams of spotter observations and associated mapper data to produce estimates of the user's position. The trackers encapsulate the system's understanding of how various types of radio signal propagate and how that relates to distance, the physical environment and location. Trackers may use only the data provided to them by the spotter and mapper, or may use extra data like road paths and building locations to produce more accurate estimates. As an example, Place Lab includes a simple tracker that computes a Venn diagram-like intersection of the observed beacons. Place Lab also includes a Bayesian particle filter tracker that can utilize beacon-specific range information. While computationally more expensive, the Bayesian tracker provides about a 25% improvement in accuracy and allows Place Lab to infer richer information like direction, velocity and even higher level concepts like mode of transportation (walking, driving, etc.). [1]

6.3 Graphical User Interface

"At its root, every design decision, from a single pixel to the information architecture, should be judiciously made through a careful consideration of what each design decision affords, and conversely costs, the users." (IBM accessed 12 November 2004)

The user interface (UI) design and development is a key component of our project. Without careful consideration of the UI, software features maybe lost, the end-user will get frustrated, all leading to a view of a poor piece of software development. In this section, we aim to provide some background on standard UI design and why it is so important. This will also comprise of UI design specifically for a PDA. Following this will be a brief section on our choice of implementation language. Lastly, we will focus on the key component, which is the actual designing of our interface.

6.3.1 User Interface Motivation

Developers should see the functionality of a system in combination with the UI. Too many developers see the UI as an add-on (Dray accessed 13 November 2004)[A.L2]. Most users typically do not make distinctions between the underlying functionality and the way it is presented in the UI. To users, "the UI **is** the system". Recent data suggest that:

- the user interface is 47% to 60% of the lines of code
- a UI is minimally 29% of the software development project budget
- a UI may take as much as 40% of the development effort.

Dray (correctly) argues that the user interface is more than how the screen looks; it includes all aspects of the system design that influences the interaction between the user and the system. What it comes down to is that well designed systems are "usable". It is about the whole user experience. A key principle that is prevalent in our design of our UI is that we try and let the user focus on the task i.e. finding information about the tour-guide, without having to pay attention to the technology tool itself. In some sense, the user interface should appear "ubiquitous" to the key component of our project, the information about the college and the precise location of the user.

Usable systems are easy to learn, remember and use, efficient, and designed to minimise errors and to promote user satisfaction. Here are some benefits summarised by Dray of a well designed UI:

- reduced errors
- more focus on tasks to be done, rather than on the technology tool
- reduced rework to meet user requirements
- fuller utilization of system functionality
- higher end-user satisfaction.

6.3.2 Some principles of UI design

In Talin's paper "A Summary of Principles for User-Interface Design" (accessed 15 November 2004), he represents a compilation of fundamental principles for designing user interfaces, which have been drawn from various books on interface design and from his own experiences. In designing our UI, these following principles are the most relevant and will be adopted:

1. The principle of feature exposure

This is to let the user see clearly what functions are available. According to the Myers Briggs personality classification, there are two types of personality associated with analysing sensory details of the environment, intuitive and sensible. Some psychological studies say that "sensibles" outnumber "intuitives" in the general population by about three to one.
According to Talin "Intuitives prefer user interfaces that utilize the power of abstract models -- command lines, scripts, plug-ins, macros, etc. Sensibles prefer user interfaces that utilize their perceptual abilities -- in other words, they like interfaces where the features are "up front" and "in their face". Toolbars and dialog boxes are an example of interfaces that are pleasing to this personality type."
This means that our user interface is probably going to be a graphical user interface, instead of a command line program. Features of the program need to be easily exposed so that a quick visual scan can determine what the program actually does. Of course, this also means that some of our features will not be exposed right away. "In this case, it is best to structure the application like the layers of an onion, where peeling away each layer of skin reveals a layer beneath. There are various levels of hiding.

2. The principle of help

This is to understand the different methods a help a user needs. This is vital in getting the user to learn the system quickly and to use it efficiently. The "perfect" UI can be argued to have no need for any help at all, as the functionality should be so simple that it is self-explanatory. However, others may argue that the "perfect" UI must consist of some sort of help. In our project, we have decided to adopt the safer latter approach.
There are five basic types of help, corresponding to the five basic questions that users ask (Laurel, 1991)

1. **Goal-oriented:** "What kinds of things can I do with this program?"
2. **Descriptive:** "What is this? What does this do?"
3. **Procedural:** "How do I do this?"
4. **Interpretive:** "Why did this happen?"
5. **Navigational:** "Where am I?"

Different strategies maybe employed to addressing the needs of each question. For example "about boxes" is one way of tackling the needs of question 1. Question 2 can be answered with a standard "help browser" or "tool tips. A help browser can also be useful in responding to question 3, but these can sometimes be more efficiently addressed using "cue cards", interactive "guides", or "wizards" which guide the user through the process step-by-step. The fourth type has not been well addressed in current applications, although well-written error messages can help. The fifth type can be answered by proper overall interface design, or by creating an application "roadmap". Tree structures have also been adopted, especially in the design of web pages.

3. The principle of user testing

This is to recruit help in spotting the inevitable defects in your design. There are many bugs in UIs which are not easy to sport and may be only detected while watching someone else use the program. Seeing actual end-users test the user interfaces has been shown to be an extraordinarily effective technique for discovering design defects. However, there are specific techniques that can be used to maximize the effectiveness of end-user testing. These are outlined by both Tognazzini and Laurel and can be summarized in the following steps:
Set up the observation. Design realistic tasks for the users, and then recruit end-users that have the same experience level as users of your product.
Talk about and demonstrate the equipment in the room.
Explain how to "think aloud". Ask them to verbalize what they are thinking about as they use the product, and let them know you'll remind them to do so if they forget.
Explain that you will not provide help.
Describe the tasks and introduce the product.
Ask if there are any questions before you start; then begin the observation.
Conclude the observation. Tell them what you found out and answer any of their questions.
Use the results.
It is much easier to deal with design defects before it’s implemented than after. Therefore, it’s more efficient to build prototypes and test that before building the real program. Tognazzini suggests that you need no more than three people per design iteration. Any more than that and you are just confirming problems already found.

4. The principle of aesthetics

This is about creating a program that is visually appealing. It does not have to be a visual work of art, but it is important that it is not ugly. A cleaner looking interface will look more professional and there will be a greater tendency to “appreciate” all the features built into the system. Users don't like using programs that feel sluggish or slow. There are many tricks that can be used to make a program "feel" snappy, and usually a “crisp” clean interface is one of them.
In Bruce Tognazzini’s “Tog on interfaces”, he poses three questions which leverage the user's strengths and create an interface that helps them achieve their goals.

1. What are the user's goals?
2. What are the user's skills and experience?
3. What are the user's needs?

We will aim to answer these questions in the design of the UI. To see more about general principles of user interface design and contrasting design decisions, please see the Appendix 3.

6.3.3 PDA User Interface Design

"If you're trying to make a simple design on a small screen, you're not providing value to the user when you give them a whole slew of buttons. You're being lazy or indecisive, and the result of your inability to make tradeoffs is a poor user experience." (Rob Haitani, original designer of the Palm OS)

The most unique challenge in designing user interfaces for a mobile device is the significantly reduced screen size compared to a traditional desktop application. A PDA UI is not about densely packing buttons or data into a small display or spreading the functionality across endless sub-screens.
Here are some key recommendations in designing an effective PDA user interface (PDA Aesthetics and Interface Design accessed 12 November 2004)
Keep **page length to fit the screen width** in order for the user to avoid scrolling.
Try and keep flat hierarchy for navigation. Necessary information should **never be more than three levels deep**; otherwise the user may be lost.
The **inclusion of a back button** on every page that has the same function as it has in a regular browser can be very important as it is one of the most used features on a mobile device.
Provision of appropriate navigation to bring users back to a **home page or entry** page after visiting a specific page on an item can be very valuable in not losing the user.

Figure 1. Example of a usable design for a PDA Web Page

6.4 User Interface Design and Implementation

The graphical user interface needs to present the end-user with two main features.

The user needs to access the location tracking map (to see where they are)
The user needs to find out information about a specific building

Below, is a visualisation of our proposed design.

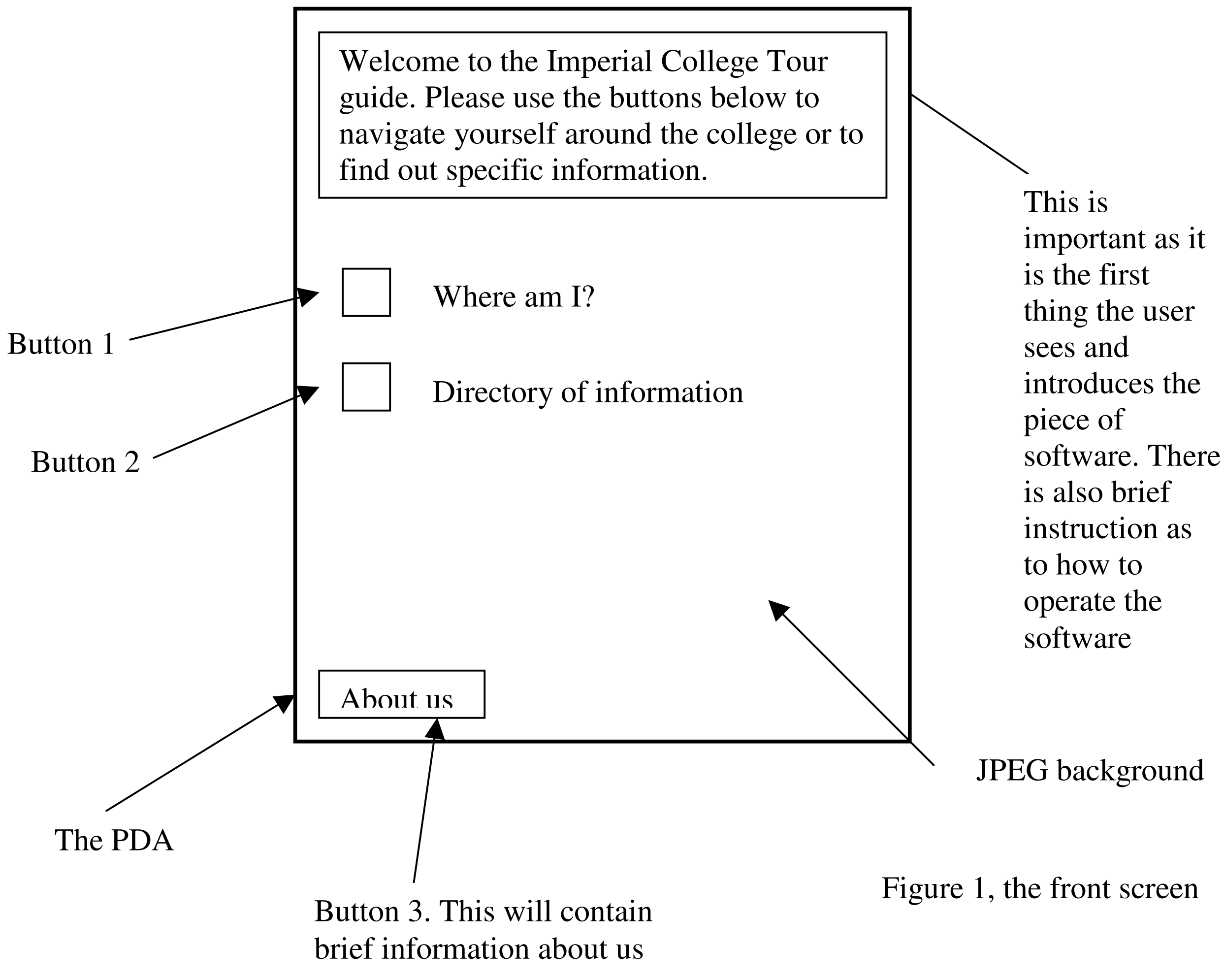
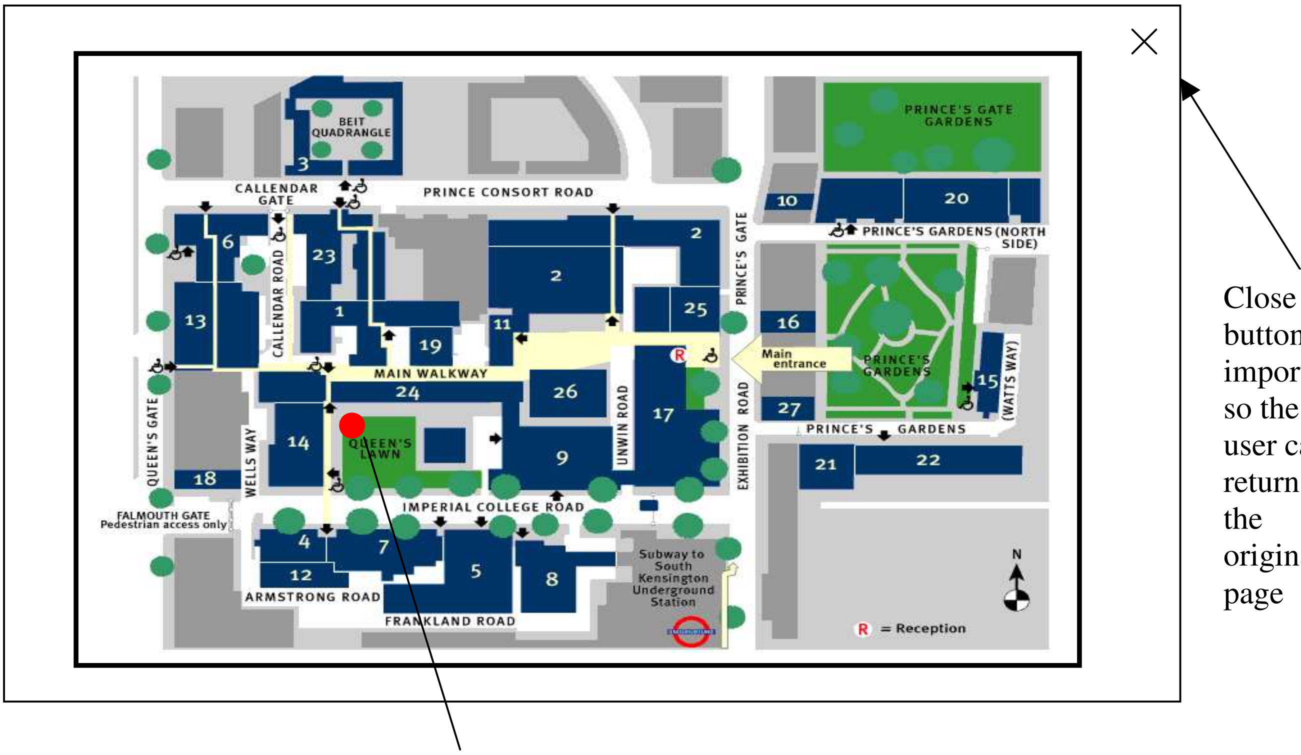


Figure 1, the front screen

The three buttons will each bring up a pop-up box regarding the relevant information. The motivation behind a pop-box is that the user will still be able to see the front-screen, and hence will be able to easily get back to their original location. Also, a background picture shall be added to enhance the look of the interface

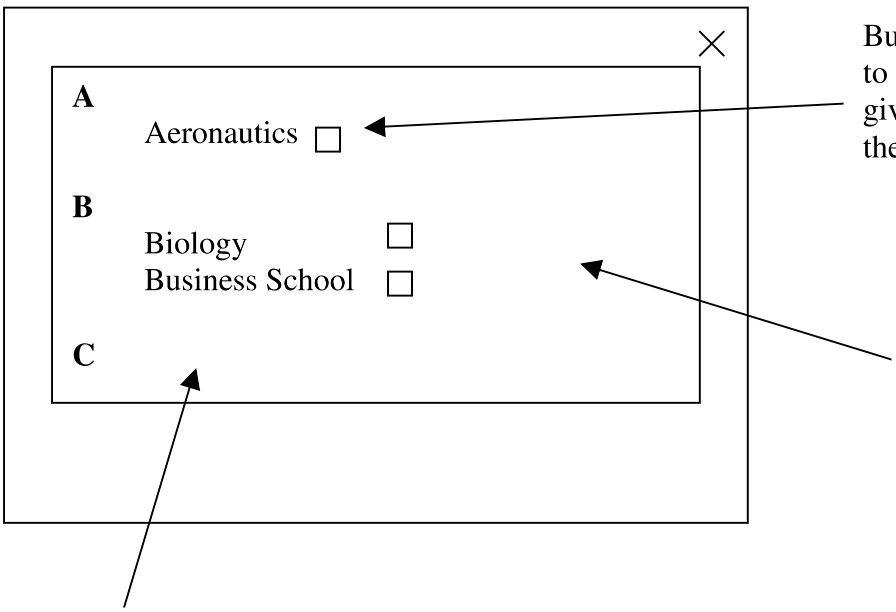
One thing to note is that on the top left hand corner of this screen, there will be menus giving the user a variety of options. This is in-built into PlaceLab and not part of the user interface design.



Close button, important so the user can return to the original page

Red target indicating user's location

Figure 2, "Where am I?" button



Button 4. These buttons next to each building name will give another pop up giving the user information

This list needs to be scrollable. This is vital as screen space is very limited

A-Z List of building

Figure 3, "Directory" button

The practice of object inheritance is well documented in Java. C# visual studio provides visual inheritance. This is Microsoft's way to describe implementation inheritance of a visual object such as Windows Forms. Inheriting forms is a powerful feature that allows cross-project standardization of common elements on a form that may be changed in one place rather than in countless individual forms. This is particular helpful in designing the forms which brings up information about each individual building, as the structure of each form will be the same. As our tour guide will contain information on several buildings, we have just shown three in the UML diagram below. Of course, this UML model can be easily extended to cope with many buildings with ease.

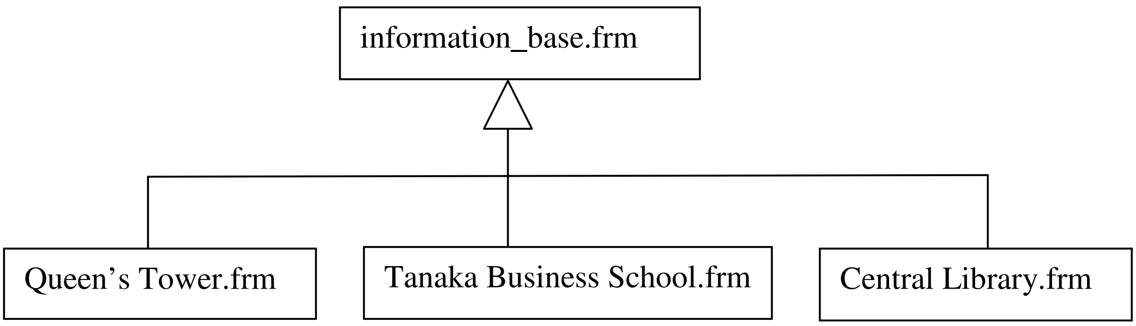


Figure 4, Information content UML diagram

Figure 5, shows the base framework of the interface regarding building information content.

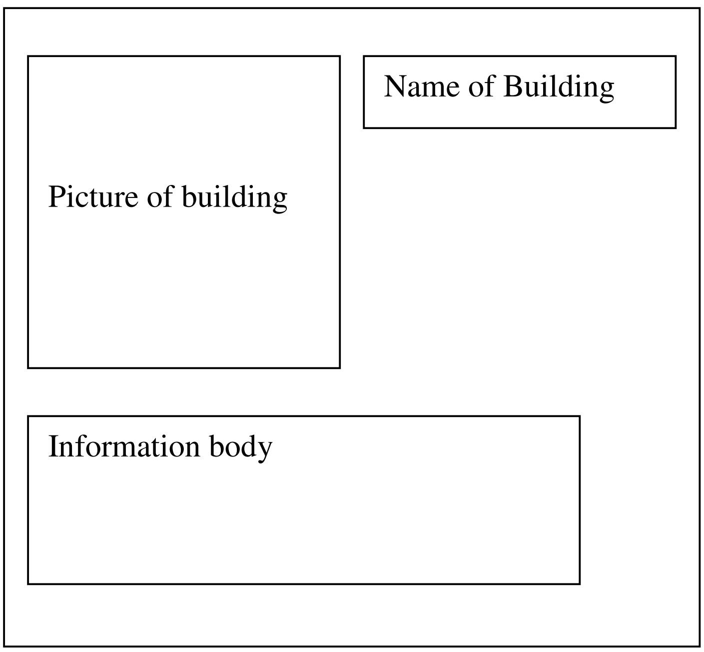


Figure 5, information_base.frm

6.5 Prototypes

In this section there are the prototypes we produced from the initial design. These prototypes were tested in terms of usability, where suggested enhancements were given.

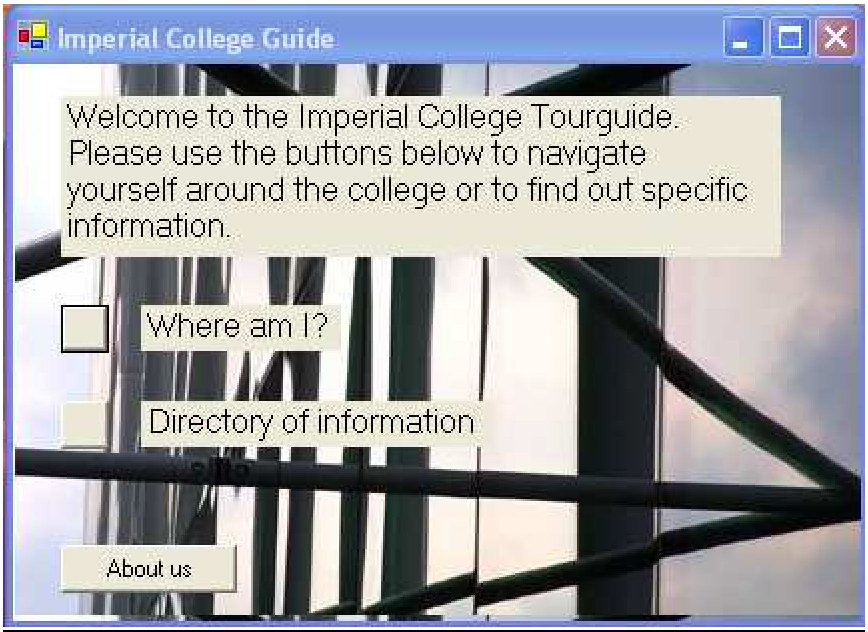


Figure 6, main screen

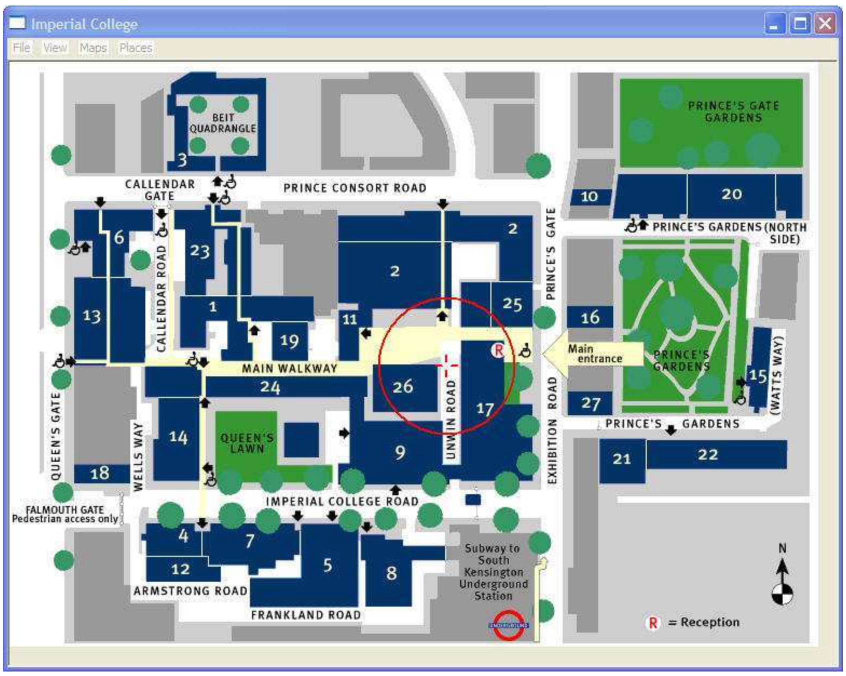


Figure 7, where am I?

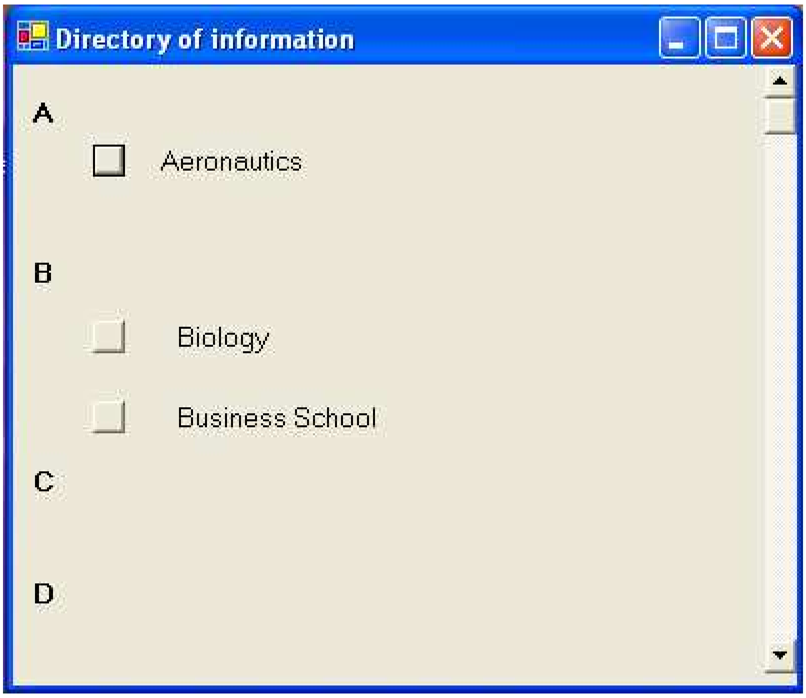


Figure 8, Directory of information

Figure 9 below shows the information screen one would receive when pressing the queen's tower button in the directory of information screen. The form is inherited from base_information.frm.

Figure 9, queens_tower.frm

6.6 Final Implemented System

During the course of designing and implementation, an aerial map location tracking feature has been developed. This will need to be added to the user interface. In testing the prototypes developed, a few interesting and certainly vital issues arose. Firstly, the major concern was the interface did not look professional enough and the colour scheme was not quite right. Secondly, the navigation of the system is not perfect; it does not flow very well. The text on the user interface should be more descriptive to guide the user around the interface. However, the main features i.e. integration with PlaceLab all worked excellently. Here is a summary of problems (mostly aesthetic) which need to be addressed before the final version is released.

- The look of the system needs overhauling
- The navigation structure is not fluid enough
- The map needs a key
- Building information needs to have more information i.e. nearest points of interest and building numbers
- Add aerial map location tracking feature
- The text within the user interface needs to be more descriptive

Pure components

In tackling the look of the system and the navigation structure, we used piece of software designed for Visual Studio .Net called Pure Components Here is a quick summary of pure components, "PureComponents specializes in development of unique and advanced .NET components. PureComponents's mission is to create unique components that meet some important criteria. These mainly include easy usability and high utility gain. Components must have a simple and comprehensible object model and must expose all required functionality."

The next few pages will show the final implemented user interface.

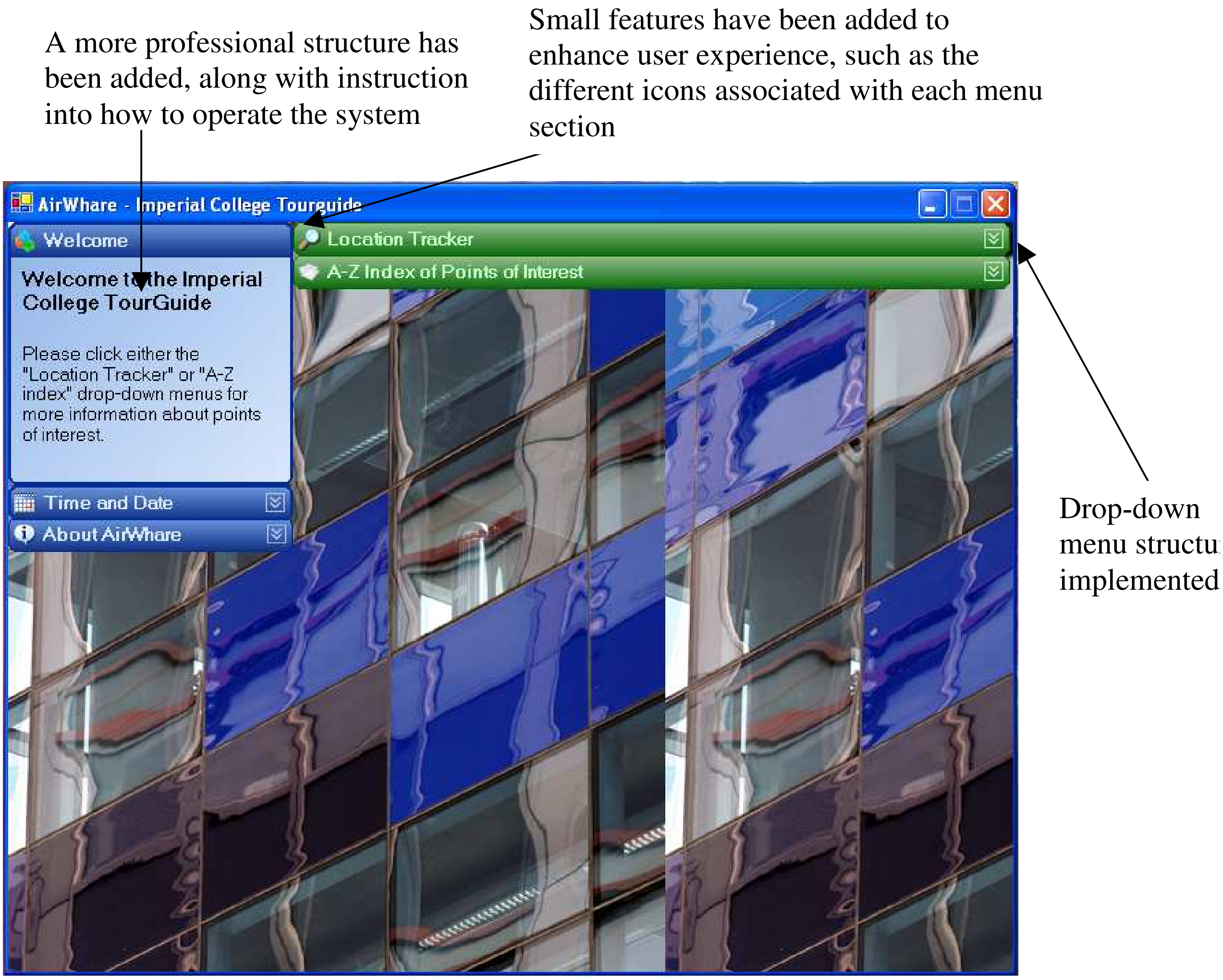


Figure 10, main screen

There have been a vast number of changes to the main screen. Firstly, the overall colour scheme and "look" of the user interface has changed. The second major change is that the main screen, instead of having buttons linking it to different parts of the system, it now has a drop-down menu structure. Not only does this make it more compact, it is useful for the user as not all the features are fully exposed from the offset.

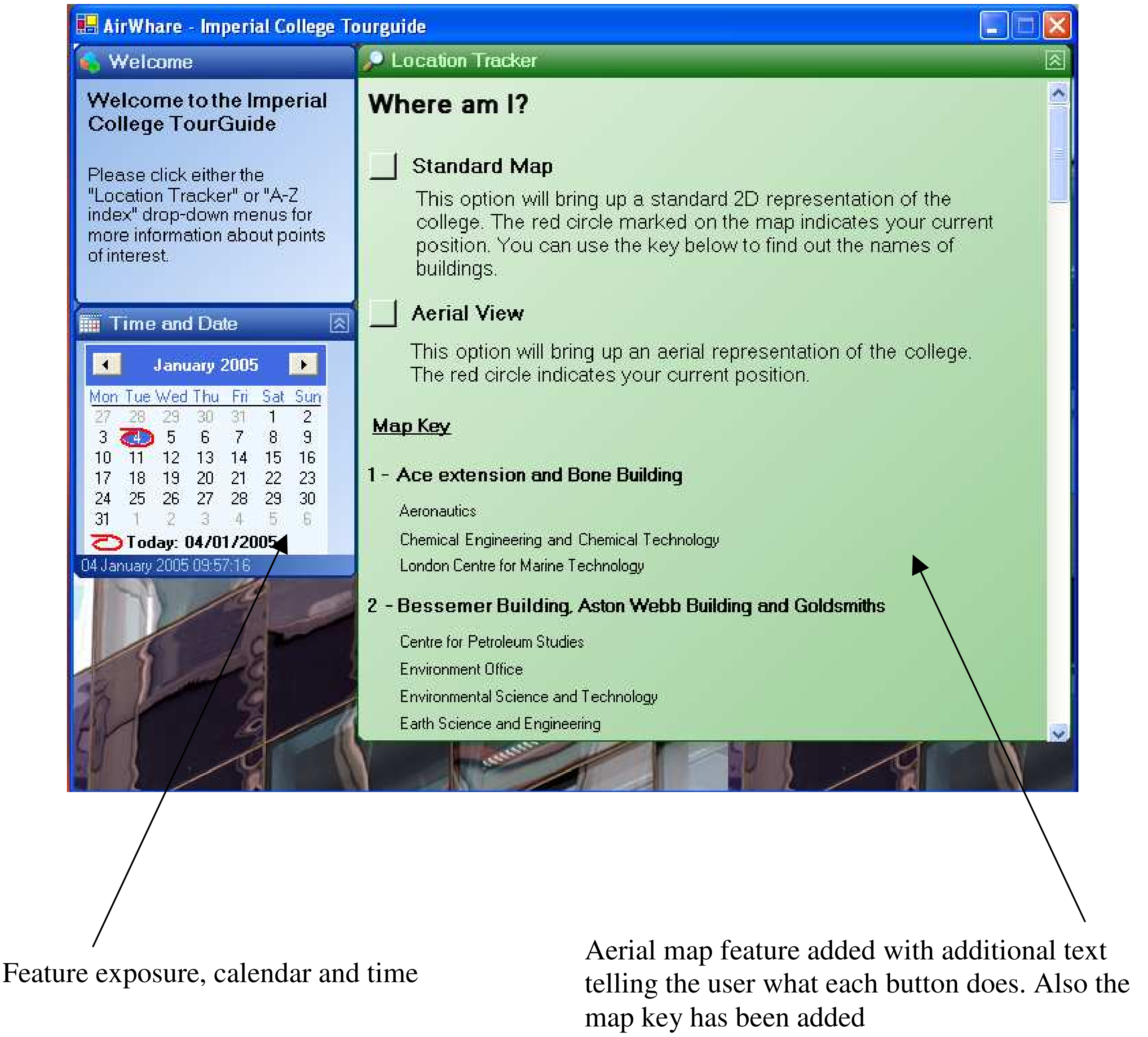


Figure 11, main screen

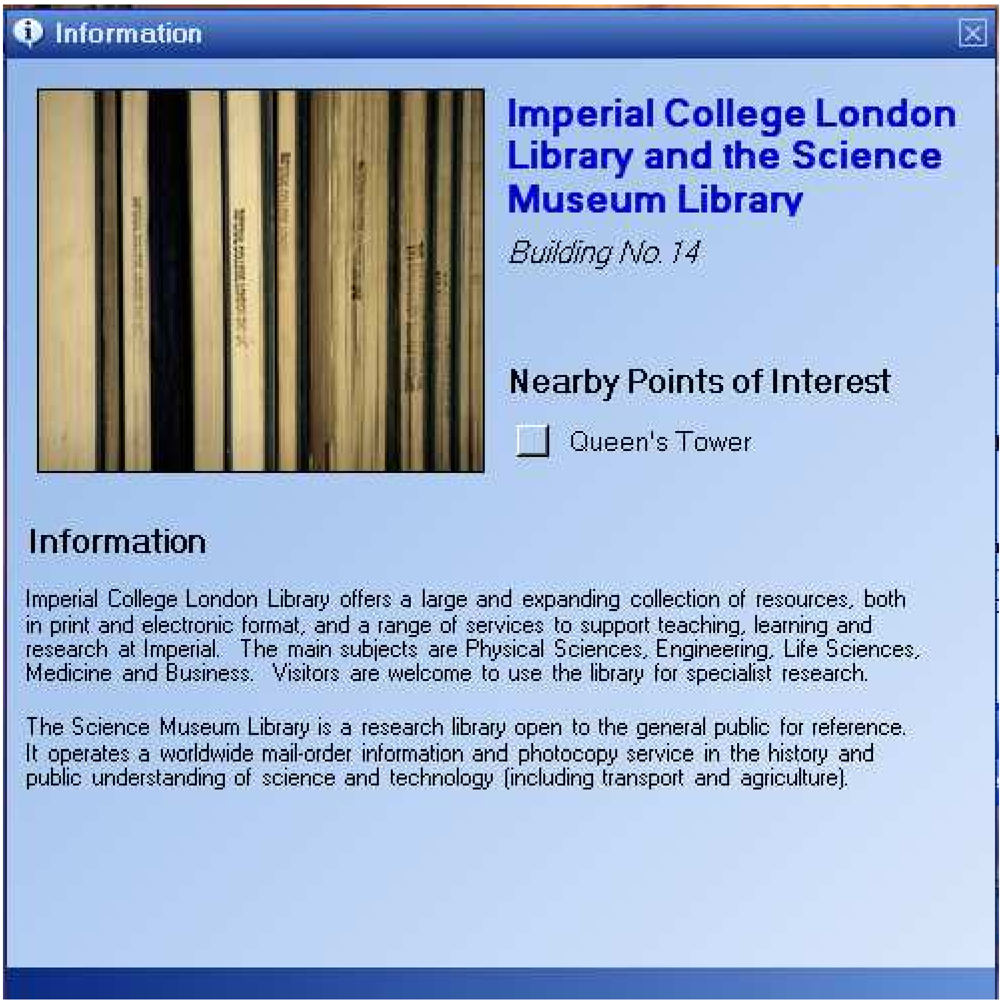
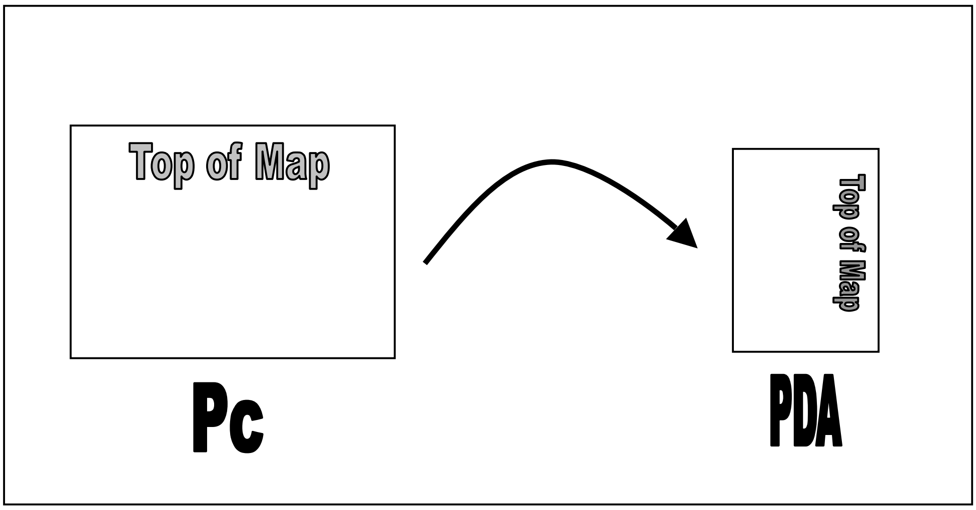


Figure 11. building information interface

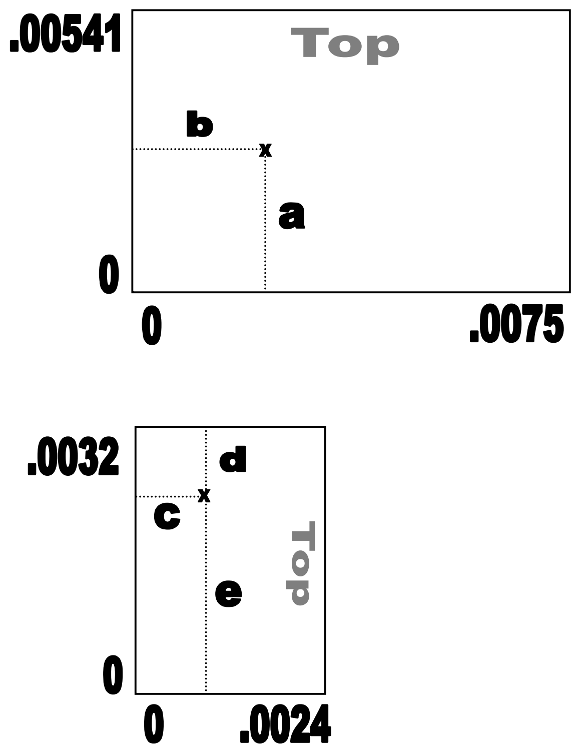
The old "directory listings" has changed its name to A-Z index. A nearby points of interest has been implemented, along with the building number is shown.

7 Rescaling the Map

After inputting all of the access points into the large map of the college, the same access points needed to be transferred into the PDA-sized map. This proved to be much trickier than was first thought. The map had to be re-sized, rotated and cropped, as the viewing area on a PDA is much smaller than on a monitor. The resolution is also much reduced, so the map had to be slightly rearranged to enable the writing to be readable. This again caused more problems when transferring the access points to the smaller map.



As the orientation of the map was changed from portrait to landscape, the coordinates of the wireless points had to be changed.



$$c = a \times \frac{0.00240}{0.00541}$$

$$d = b \times \frac{0.0032}{0.0075}$$

$$e = 0.0032 - \left(b \times \frac{0.0032}{0.0075} \right)$$

B

$$(\alpha,\beta)\to\left(\left\{0.0032-\beta\times\frac{32}{75}\right\}+0.00005,\left\{\alpha\times\frac{241}{540}\right\}-0.00002\right)$$

8 Collating the Tour Guide

There were a variety of places from where we obtained the information for our tour guide. The Internet was used widely as there are many different websites dedicated to the university. We also looked at the tour guide recently installed in the Tanaka business School which became a key source of inspiration for our own project. This gave as an insight into what other people use, and what is needed. We also found out that Peter Gillings was the creator of that tour guide so we emailed him for further help

During our meeting with Mr Gillings he gave us many ideas and even more pictures that he has used in the different pamphlets, tour guides and posters that he has designed. The information was very useful, but all of the pictures needed to be resized and cropped as they were of a much too high quality for us. Also all of the text was in picture format, so that needed to be extracted from the images before we could use it.

One of the problems with the pictures that we obtained from the Internet and Mr Gillings is that many of them were 'artistic pictures'. These look very good in brochures, but we felt that they were of little use to someone using a tour guide as they would want a picture that they could relate to while walking around. This meant that we had to take our own pictures that looked more realistic.

Deciding what type of information to include in the tour guide became a difficult process. It was clear that different users would have different general interests and would be interested in different aspects of the campus. However due to the resource limitations of the PDA it would be impossible to include so much data. Overloading the tour guide could easily clutter the system and severely damage the usability of the system. The group decided that further specific information could be included to the system with minimal extra costs and when it was though appropriate or necessary.

9 Testing and Evaluation Overview

During development the separate components of the system underwent constant testing in order to guarantee the integrity of those parts (see previous). Further testing would be necessary in order for us to assess the overall quality of our product and compare it to our requirements made in the specification.

The final testing phase concentrated on three main areas:
How the system compared to the initial specification? Did our team achieve all that was required or were there failings in our project?
The usability testing. How does the system stand up to general use? Would the product suit the needs of its potential users and what changes may be necessary in order for it to do so?
Do the separate parts of the system fully integrate without the occurrence of bugs and errors? This would involve finding the limitations of the system and how far it may be pushed.

Ideally we would be able to test our system with similar, existing products and make a comparative analysis. This would have allowed us to shown how our team competed with its industrial equivalents and produce an excellent measurement for the success of our product. However due to the necessity to locate near the mobile tour guide area (which were largely in other countries) it was impossible for the team to use and make quantities evaluations of other products.

9.1 Testing the initial specification

The first and most important test of our system was made against our own criteria – the initial specification (see section **3 The System Features**). It was essential that our system meet these standards; the following checklist shows how our final product compared to those initial intentions:

BASIC REQUIREMENTS	
The system must act as a guide to the university campus, thereby holding relevant information to key areas of interest on the campus	✓
The system must hold a graphical representation of the campus thereby allowing a user to easily select information relating to their position	✓
The system must include locating functionality in order to track the user's position over map. This must operate over a minimum of ten access points for a basic accuracy.	✓
The system must operate on a mobile PC of the maximum weight of a laptop	✓
EXTENSIONS	
The system must operate on a mobile PC of the maximum weight of a palmtop	✓
The tracking map must have high enough resolution to accurately show the user's position.	✓

9.2 Usability testing

In order to evaluate such a system it is important to test how potential users respond to it. To this end we developed a series of tests which would allow us to make a qualitative assessment of our product.

9.2.1 The Test Plan

Our main concerns with the respect to the usability were:
Is the information contained comprehensive, relevant and easy to access.
Does the system allow a person unfamiliar with the college to easily and quickly navigate the campus

To this end we gave the product to a group of third year Imperial College Students (from various degree disciplines) and requested they use the system to find certain information on College sites. Although these test subjects could not reliably test the quality of the navigational tools (being far too experience with the layout of the campus) they would be able to supply a useful opinion regarding the worth and accessibility of the information content.

In a second round of testing we gave the product to a number of persons who have had no significant visual contact with the college. This included relatives and friends of our team of varied computing skill. This was to test the location and navigational qualities of our system.

The two groups were given tasks to perform using Airwhare. The time taken to complete these tasks were measured and recorded. We expect to complete some basic statistical testing on these results thereby forming an assessment of the usability of the product.
Other qualitative questions will be posed to the test subjects to suggest where there are failings in the systems and how it may be improved.

9.2.2 The Test Results

The tests described above were organised. The following sites in Imperial College were used:
A - The Computing Department
B - Tanaka Business School
C - The Queen's Tower
D - Beit Hall

As described above, the first test involved measuring the time taken for an experienced Imperial College Student to access information on the above places (to the nearest second). The test subjects were given a palm top with the Airwhare application open.

The following results were obtained:

Table 1

TEST SUBJECT	Time Taken to Find Information (seconds)				Total
	A	B	C	D	(seconds)
6	6	5	5	5	21
7	5	2	3	2	12
8	6	6	6	4	22
9	5	4	3	3	15
10	7	5	6	4	22
AVERAGE	5.8	4.4	4.6	3.6	18.4
Test Case	3	3	2	2	12
Difference	2	1	2	2	6
% Difference	45	47	53	80	53

Included are the results of the Test Case, the average results of the Test Subjects, the absolute difference between the average time and the Test Case's time, and this value as a percentage.

The second test required novices to be timed as they attempt to navigate a route around those locations (to the nearest five seconds). The test subjects were given a palm top with the Airwhare application open at the entrance to the Department of Computing.

The following results were obtained:

Table 2

TEST SUBJECT	Time Taken to Navigate Route (seconds)				Total
	A to B	B to C	C to D	D to A	(seconds)
A	200	265	305	420	1190
B	285	325	300	335	1245
C	165	230	210	275	880
D	185	245	225	295	950
E	190	285	200	280	955
AVERAGE	205	270	248	321	1044
Test Case	140	225	195	270	830
Difference	65	45	53	51	214
% Difference	46	20	27	19	26

Included are the results of the Test Case, the average results of the Test Subjects, the absolute difference between the average time and the Test Case's time, and this value as a percentage.

Other relevant comments made are quoted below:
"The map is confusing when it jumps around"
"The tracking dot is erratic"
"The information is rubbish"

9.3 Testing the system's integrity

The system went through a process of constant testing during its development. This allowed problems and bugs in the system to be recognised locally and rectified before further work was completed. Additionally we could be confident that the final system had a minimal risk of developing a fault.

The final system however went through rigorous usage by our team and external users of the system. No technical faults have thus far been reported.

9.4 Conclusions and changes made to the system

9.4.1 Initial Specification

We may conclude that the system fulfilled its requirements as seen in the previous section.

9.4.2 Usability

The results table (Table 1) shows that information in the tour guide is relatively easy to access. Although there was a large percentage difference between the times taken by the test case and the test subjects (53% on average for all queries), in absolute terms this makes a difference of only six seconds for each query. Given the small number of queries made and the simplicity of the operation performed this does not seem to indicate any significant problem in the use of the system. It must be further appreciated that the test case was highly experienced in this system's use and his time on this exercise was close to the minimum possible.

The results table (Table 2) shows that on average the test subjects took only 26% longer to navigate the route. This in not a discouraging value considering the experience the test case had of the campus. Another encouraging sign is that the final leg took on average only 19% longer which may indicate that the users are quickly becoming familiar with the system.

After an analysis of the questionnaire responses we undertook the task of editing the system in accordance to some of the comments made. This involved two main problems – the first with the tracking map and the second with the quality of information in the tour guide.
To solve the first issue we increased the number of access points listed in the PlaceLab software to fifty. This would allow the PDA to assess its position with an increased accuracy as it has a larger number of points to reference from. We also reduced the size of the map so the majority would fit on the screen at once without the need for it to automatically move around. We could appreciate how for someone unfamiliar with the campus, this could be confusing and irritating.

In an attempt to respond to the criticisms of the quality of the content we took advice as too what particular information would be important in such a system. It was clear that different the groups of users would be interested in different aspects of the campus and to make our system universally useful would have to contain a vast amount of information. Due to limitations in data storage and time we therefore declined to increase the information content significantly unless a particular user was to express a strong interest.

9.5 Evaluation of testing methods, the product and the team

It has been shown that our project has met its requirements laid out for it in the initial specification. Therefore in this sense we may call the project a success.
Our other methods of testing also proved positive – showing good usability and integrity of the system. However these were by no means conclusive tests, having only a small number of test subjects and limited time for assessment. A suggested further study would be to give the product to fifty prospective students and measure how useful a tour they receive from Airwhare.

Improvements may be made with the accuracy of the tracking system. Whilst scanning for access points around the College campus over two hundred base stations could be found. Theoretically these all could be used by a PDA to pinpoint its location and improve its accuracy. However because these access points may be other mobile devices or routers which may not be in a fixed position it would be unwise to calculate locations from them without confirmation they are stationary points. Otherwise a GPS system could be included with the product, although they too have their own flaws such as not usable indoors and being a much greater cost.

Further work could be carried out to improve the information content. This could be by the way of including audio and video clips or increasing the amount of information held in the system. However a much better solution would be to produce specific tours for the different types of users who may wish to be guided around the campus. In this way the PDA would only have to access the information relative to its user at any one time. Additionally with this, an extension could be made to produce specific routes for each type of user, for example a prospective computing student could a have tour made for him that guides him around the Department of Computing and other relevant locations, displaying the relevant information.

Problems arose often caused by the limited resources of the PDA. To run complex tracking software or interfaces often stretches this type of PC to its limits. We believe therefore that with every year bringing quicker computing power such systems will be much easier and therefore worth implementing.

We believe that as a team effort this project has been an overwhelming success. We have mastered new technologies such as Intel's Place Lab (which is still very much experimental) and overcome the challenges that arise from working with limited resources. We have been well organised in our work allowing us to accomplish many tasks efficiently without having to dismiss a good idea because we found it too challenging. Further more we have enjoyed this project which has enabled us to create a product we are truly proud of.

10 Bibliography

[1] Place Lab: Device Positioning Using Radio Beacons in the Wild.
Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello and Bill Schildt.
IRS-TR-04-016. - http://www.placelab.org/publications/

[2] Intel Place Lab - http://placelab.org

[3] http://faq.wardrive.net

[4] Jeode - Java Virtual Machine for Pocket PCs http://esmertec.com

[5] NSICOM CREME - Java Virtual Machine for Pocket PCs http://nsicom.com

[6] Microsoft .NET Homepage - http://www.microsoft.com/net

[7] Microsoft Visual Studio Homepage - http://msdn.microsoft.com/vstudio/

[8] Temples of Taipei a handheld tour guide for pocket pc By Chia-Yu Tai - http://www.charlietai.com/taipei/temples_of_taipei.pdf accessed 12/12/04

[9] Tate Modern Trial - http://news.bbc.co.uk/1/hi/technology/2225255.htm accessed 15/11/04

[10] New Scientist articles - http://newscientist.com/article.ns?id=dn6058 accessed 10/11/04

[11] Cornell University - http://www.hci.cornell.edu/projects.php?id=3 accessed 10/11/04

[12] Museum in Taiwan using PDAs - http://csdl.computer.org/comp/proc/eedings/wmte/2004/1989/00/19890195.pdf accessed 12/11/04

[13] User Interface Design Compromise, http://www-306.ibm.com/ibm/easy/eou_ext.nsf/publish/1120/$File/paper1120.pdf accessed 15/10/04

[14] Designing Usable Systems, http://www.dray.com/articles/usablesystems.html accessed 15/10/04

[15] Principles of User Interface design, http://www.sylvantech.com/~talin/projects/ui_design.html accessed 16/10/04

[16] Personal Digital Assistants, http://www.chin.gc.ca/English/Digital_Content/Tip_Sheets/Pda/tip_sheet11.html accessed 24/10/04

[17] Tog On Interface, Bruce Tognazzini, Addison-Wesley, 1991, ISBN 0-201-60842-1

[18] Electronic Guide Book Research Project http://www.exploratorium.org/guidebook/

[19] Wireless Network Visualisation - http://www.ittc.ku.edu/wlan/images.shtml

[20] Sourceforge hosted Place Lab mailing list - http://sourceforge.net/mailarchive/forum.php?forum=placelab-users

Appendices

Appendix 1 – Contrasting Design Decisions

Contrasting Design Decisions

Design	Benefit	Cost
Shallow Info Arch	Less clicks to find info	More clutter
Deep Info Arch	Clean, reduced clutter	More clicks to find info
Small font	More information per screen above fold	More difficult to read for some users
Large font	Easier to read	Less information per screen above fold
Drop down box	Selection amongst many choices using limited space	Hidden choices
Radio Buttons	See all selections at all times	Additional space required, clutter
Icons	Quick recognition once learned, aesthetically pleasing	Must be learned
Text links	Always understood	Must be read, do not stand out as actionable items as much from other text
Abbreviations	Save space	Must learn or recognize
Full text	Easily understood	Requires additional space
Keyboard shortcuts	High speed of data entry	Must be learned
Point-n-click	Intuitive	Additional time required for interaction due to increase motor skills required

Appendix 2 - User Guide

The user guide will enable users to setup and use AirWhare, the Imperial College Tour guide System.
AirWhare is divided into two applications:

- Laptop
- Pocket PC (hp iPAQ h5500)

Laptop

Setup

1) Download the .NET Framework from http://microsoft.com/downloads and following the online instructions to install it.

2) Download Place Lab from http://www.placelab.org.

3) Install Place Lab in C:\Program Files\ or it will not work.

4) Download AirWhare.zip from
http://www.doc.ic.ac.uk/project/2004/362/g0436240B/

5) Unzip AirWhare.zip and follow the Readme file moving the files to their relevant locations:

- data folder - copy this folder, replacing the data folder in C:\Program Files\placelab-win32\placelabdata\

- MapMain.bat, MapAerial.bat, imperialapsmapped.txt, imperialapsaerial.txt - copy these files to C:Program Files\placelab-win32\run\

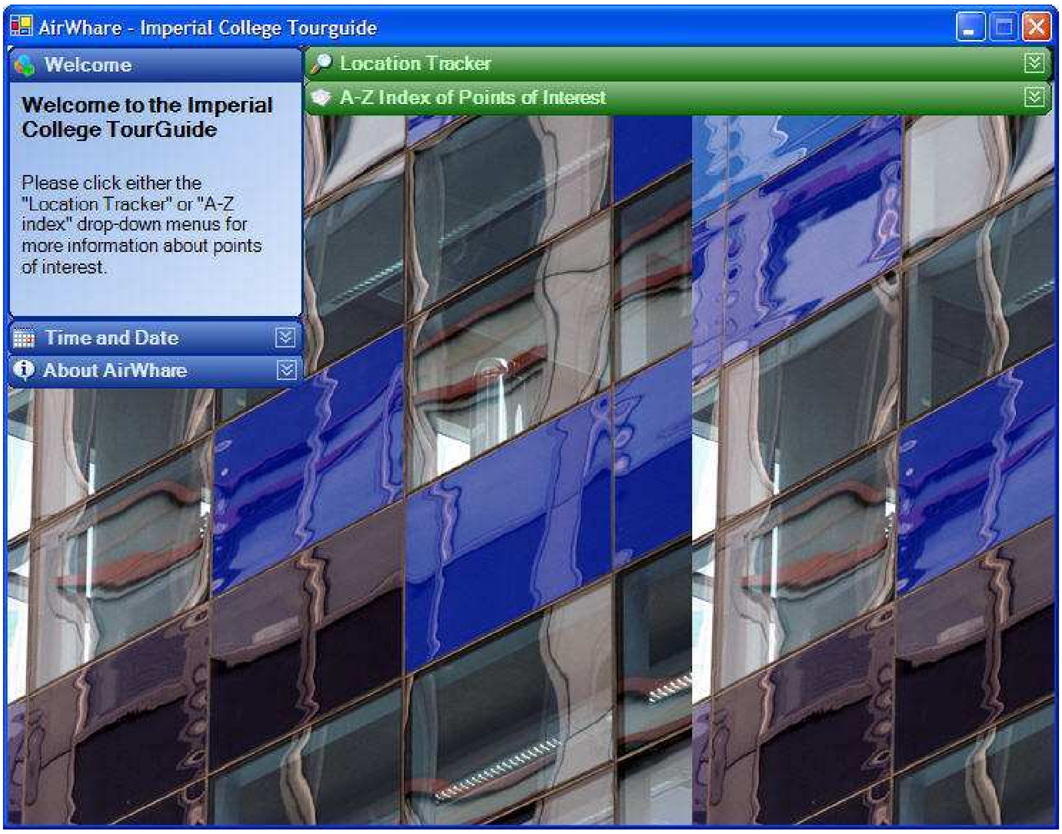
6) Download AirWhare.msi from
http://www.doc.ic.ac.uk/project/2004/362/g0436240B/

7) Install this program following the on-screen prompts.

Using AirWhare

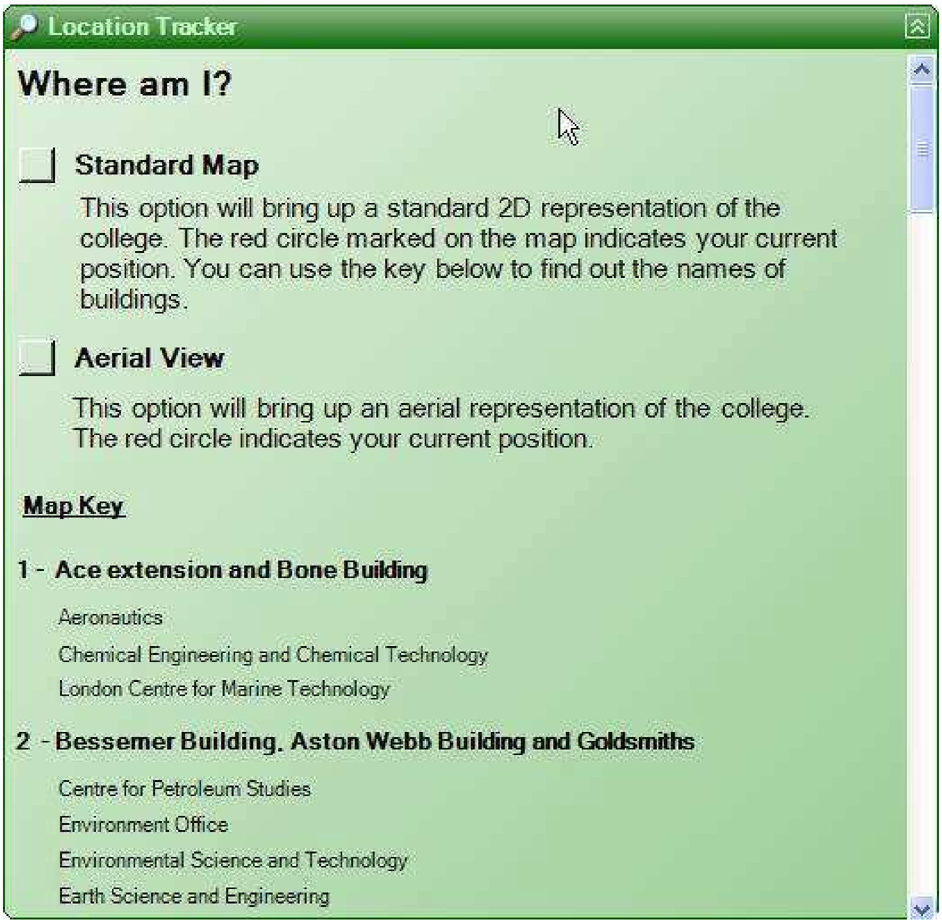
1) Ensure your wireless card is switched on.

2) Launch the application AirWhare.exe by clicking on the desktop shortcut 'AirWhare'.

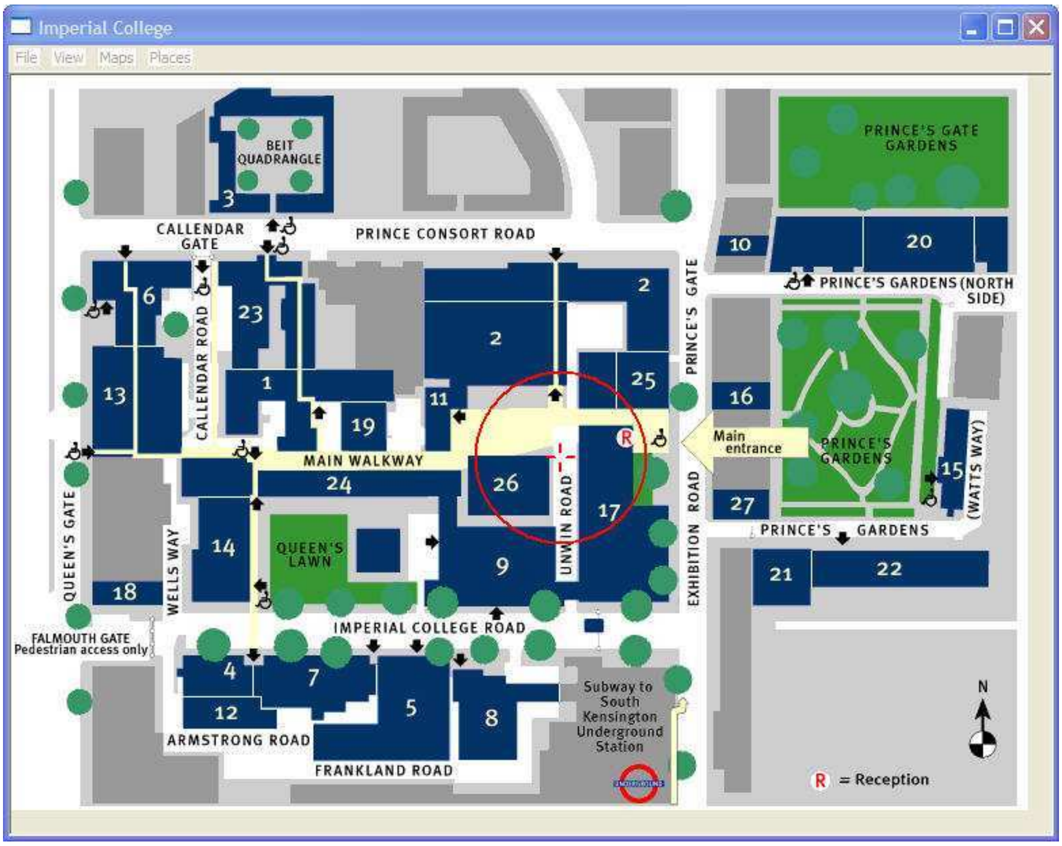


Interacting with the interface

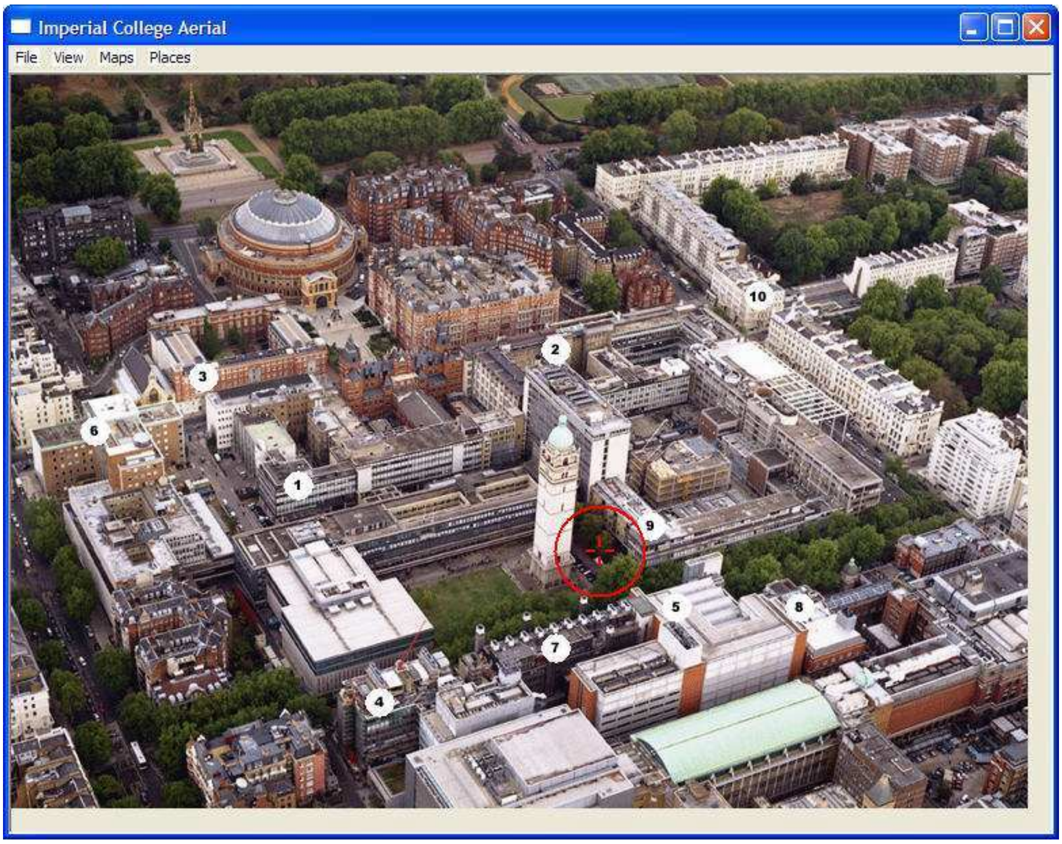
There on two main options that can be selected, 'Location Tracker' and 'A-Z Index of Points of Interest'. The 'Location Tracker' helps you find where you are.



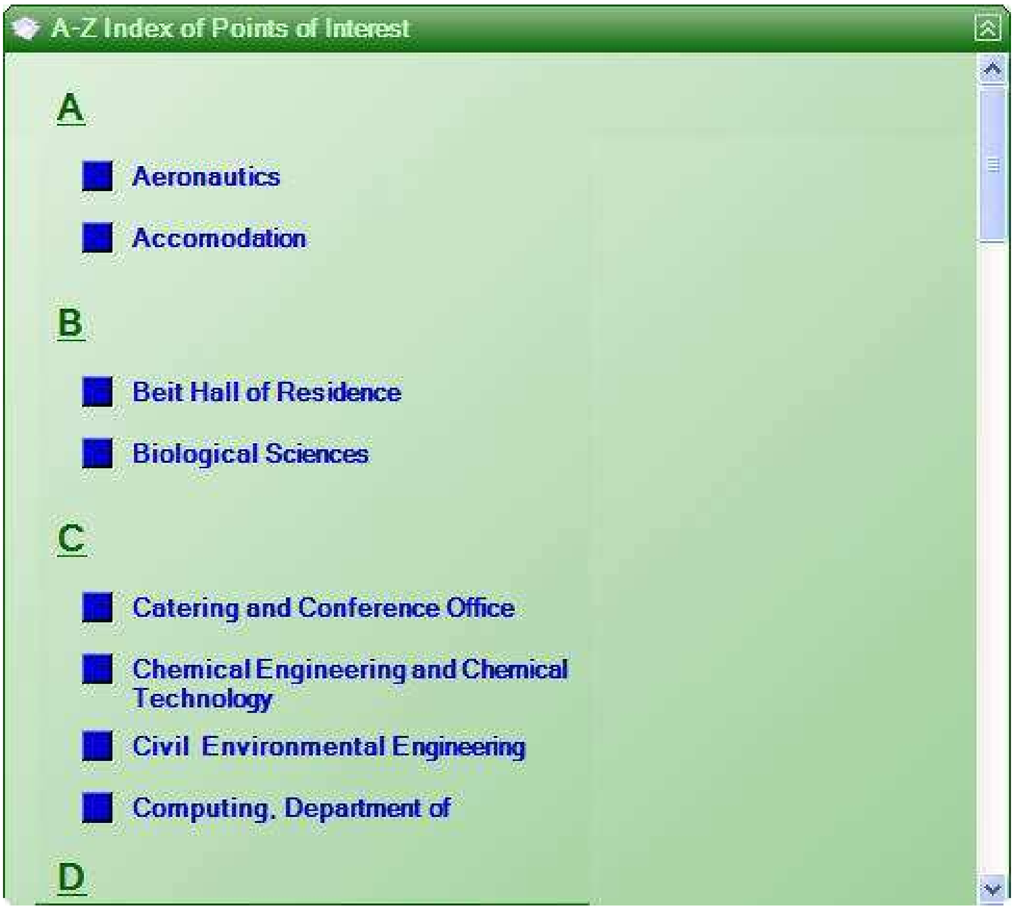
Clicking on the Standard Map button will bring up a detailed map of the Imperial College South Kensington campus. If you are near one or more of the wireless access points held in our database a red circle will appear on the map giving you an estimation of where you are. As you move around the campus this circle will continually update your current position.



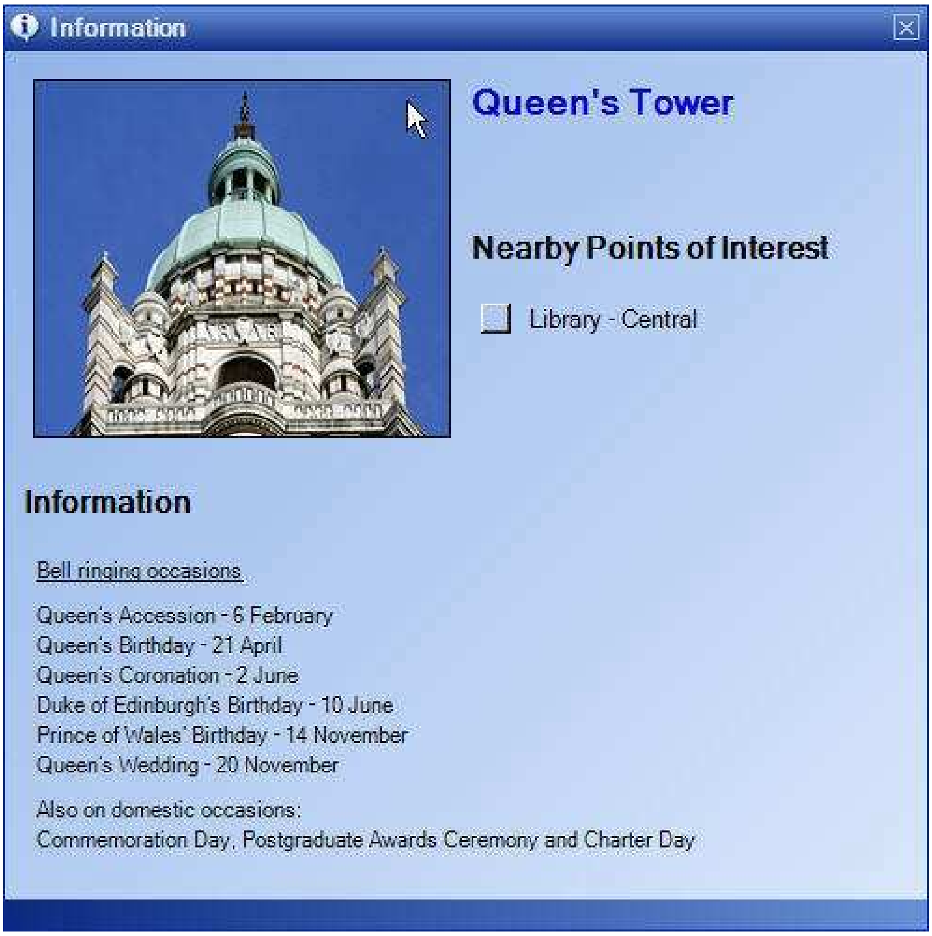
Clicking on the Aerial Map button will display an aerial map of the College campus. Again if an access point held in the database is detected a circle will indicate where you are on the map. This second map further helps you position where you currently are on the campus. You can match landmarks seen on the aerial map with the actual landmarks around you.



The 'A-Z Index of Points of Interest' allows you to search for a specific building or department alphabetically.



Clicking on a buildings button will cause a pop up window to appear with information pertaining to that particular building, possibly also showing nearby points of interest if relevant.



To close the application simply click the X button in the top right corner of the window.

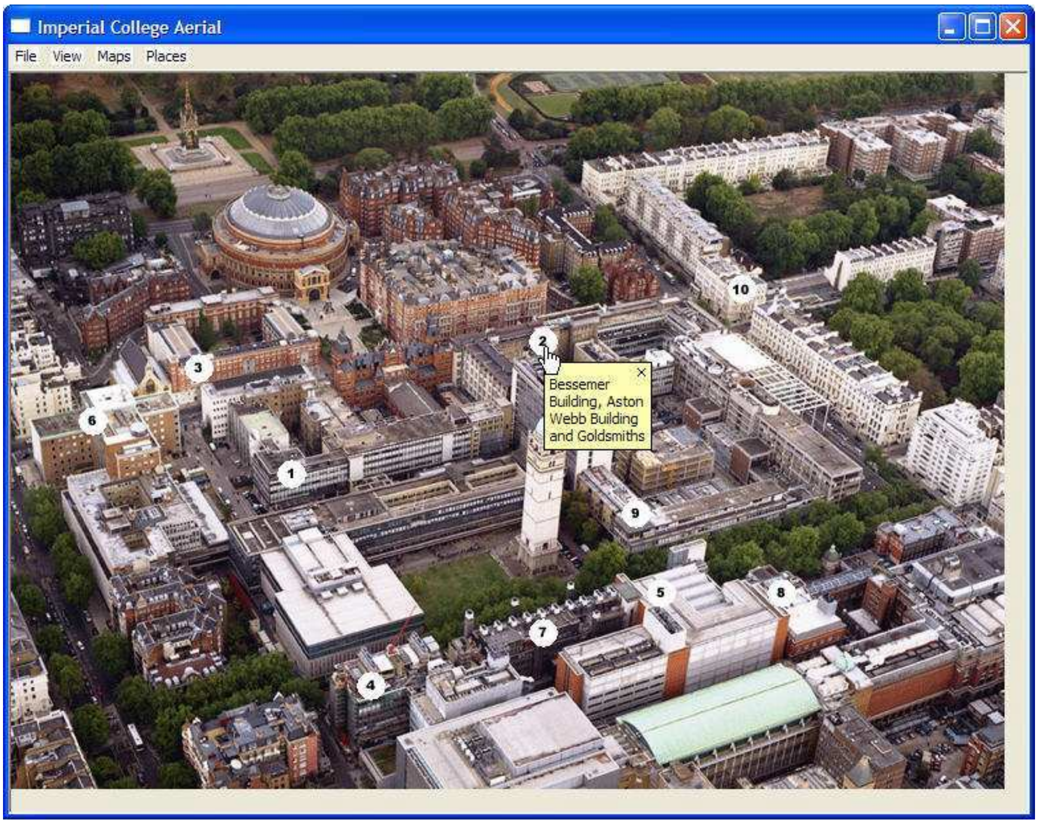
The Standard Map

Buildings are depicted in dark blue and are numbered 1 to 27. This number corresponds to the index where a user can find relevant information on any building, sorted by their number. The roads in and around the campus are clearly labelled with an arrow showing the main entrance. Wheelchair access has been indicated on the map by small wheelchair icons and black arrows. There are four menu options at the top of the map: File, View, Maps, Places.

Clicking on File and selecting Exit allows the user to close the map. Clicking on the X button on the top right hand corner of the map window will also close the map.

The Aerial Map

Prominent landmarks include the Royal Albert Hall which is the round structure on the north side of the campus and the Queens tower which is the tall building with a green dome roof in the centre of the map. There are 10 white circles with black numbers (numbered 1 to 10) over buildings. On clicking on these circles information pops up giving the name of that building. By selecting the Places menu users can switch on or off the overlaid circles giving information about each building. This may be useful if the map becomes too cluttered.



Uninstalling AirWhare

1) Click on Start -> Control Panel -> Add or Remove Programs
2) Select 'AirWhare' from the list of currently installed programs and click Remove. Follow the on screen prompts to uninstall AirWhare.

Pocket PC

Setup

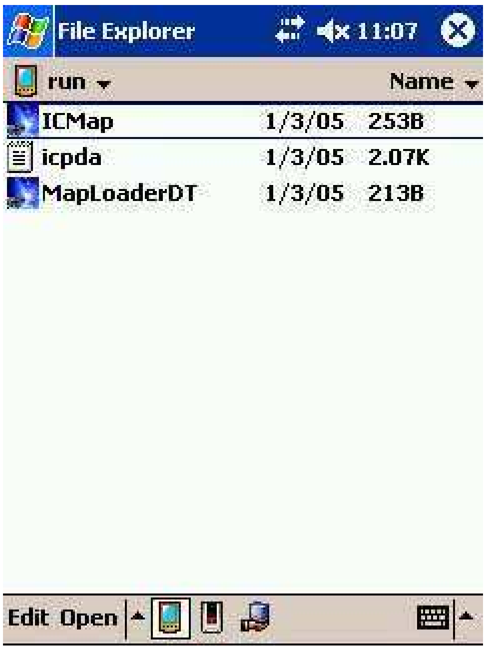
1) Download Place Lab for Pocket PCs from http://www.placelab.org.

2) Install Place Lab for Pocket PCs. Ensure you install placelab in the root directory of the Pocket PC or it will not work.

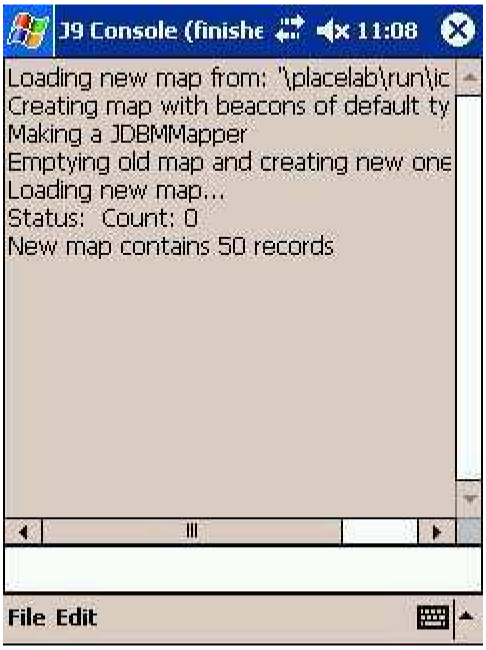
3) Download AirWharePDA.zip from http://www.doc.ic.ac.uk/project/2004/362/g0436240B/

4) Unzip AirWharePDA.zip and follow the Readme file moving the files to their relevant locations:

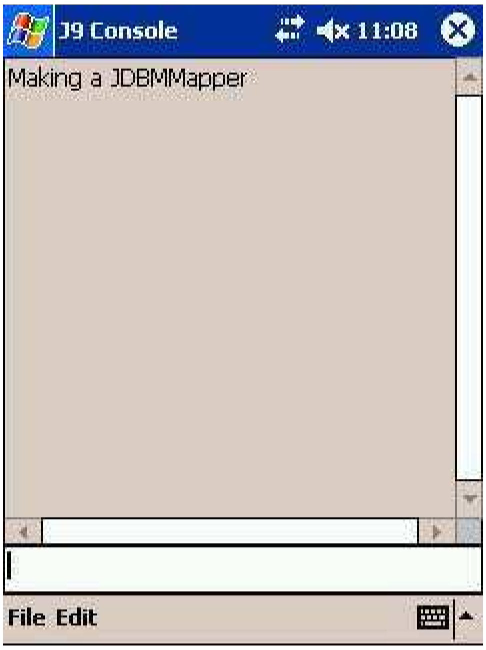
- icmap.zip- copy this file to \placelab\
- icpda.txt - copy this file to \placelab\run\
- ICMap.lnk and MapLoaderDT.lnk - copy these files to \placelab\run\

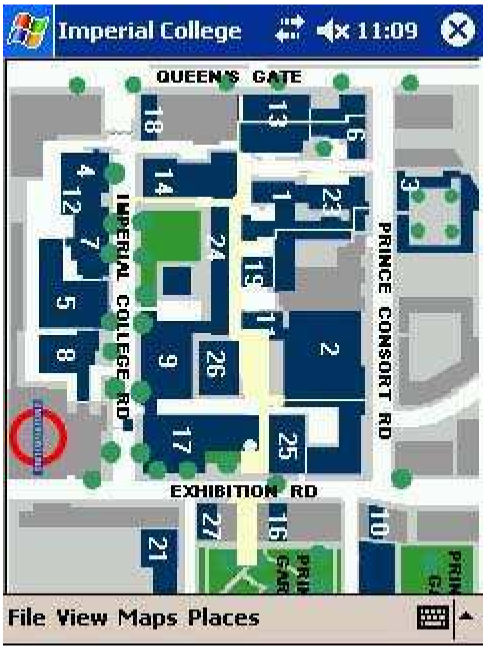


5) Using File Explorer open the placelab\run folder and tap on *MapLoaderDT*. This will load the access points into the database and only needs to be done once. The file will run and on completing successfully will show a count of the number of records (wireless access points) loaded and will display 'J9 Console (finished)' on the taskbar. Select File and tap Close to close this window. Note: DO NOT click the X to close this window as this will keep the window running in the background reducing the Pocket PCs already limited memory.



6) Tapping on *ICMap* in placelab\run will load the main map of the college. Again click File and tap Close to close this window. Select File and tap close on the J9 window that appears to close the application and not the X for the same reasons as explained above.





7) Creating a shortcut to IC Map:

- Open the folder \placelab\run\
- Hold the stylus down over ICMap and select Copy from the menu that appears.
- Open the folder \Windows\Start Menu\ and tap Edit. Select Paste Shortcut. This creates a shortcut to the IC Map which can now be easily accessed by tapping Start -> Programs and then selecting IC Map.

Appendix 3 – Ten Usability Heuristics

by Jakob Nielsen

These are ten general principles for user interface design. They are called "heuristics" because they are more in the nature of rules of thumb than specific usability guidelines.

- **Visibility of system status.** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- **Match between system and the real world**. The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- **User control and freedom**. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- **Consistency and standards**. Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- **Error prevention**. Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
- **Recognition rather than recall**. Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- **Flexibility and efficiency of use**. Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- **Aesthetic and minimalist design**. Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- **Help users recognize, diagnose, and recover from errors**. Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- **Help and documentation**. Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Appendix 4 – The Project Website

The project website, which contains the group's log-book can be found at:

http://www.doc.ic.ac.uk/project/2004/362/g0436240B (accessed 2 January 2005)
A copy may also be found on the CD accompanying this report.

The website includes:

- General project information
- The group's publications (reports)
- Minutes of group and supervisor meetings
- Relevant project links (external links to sites related to our project)
- Download Section (to install the AirWhare program)