Studienævn for Planlægning, Geografi og Landinspektøruddannelsen
Fibigerstræde 16
DK - 9220 Aalborg Øst
Tlf. 99 40 93 09
lft@m-tech.aau.dk
www.ses.aau.dk

AALBORG UNIVERSITET
STUDENTERRAPPORT

Title:	**[Train Station Proximity Finder]**
Semester:	**[Semester 1, 2016]**
Project period:	**[1/09/2016 – 12/01/2017]**
ECTS:	**[20]**
Supervisor:	**[Henning Sten Hansen]**
Project group:	**[2]**

_____
[James Ormond Fethers]

_____
[Nijole Makovskaja]

_____
[Vlad Mihai Rosca]

_____
[Mark Thomas Takacs]

SYNOPSIS:

In this report, a web application is presented to provide a new way for prospective property buyers to search for properties for sale online. This web application utilizes web GIS technologies to allow the user to query spatial data in regards to a pre-specified point of interest.

In order to achieve the desired functionality a proof of concept application is created with a focus on helping buyers find properties for sale within walking distance of train stations in Brisbane, Australia.

Number of duplicates: 2
Number of pages: 80
Finished: 12. January 2017

By signing this document confirms each group that all have participated equally in the project and that all such jointly liable for its content.

[THIS PAGE HAS BEEN INTENTIONALY LEFT BLANK]

Abstract

In this report, a web application is presented to provide a new way for prospective property buyers to search for properties for sale online. This web application utilizes web GIS technologies to allow the user to query spatial data in regards to a pre-specified point of interest.

In order to achieve the desired functionality a proof of concept application is created with a focus on helping buyers find properties for sale within walking distance of train stations in Brisbane, Australia. The application implementation utilizes several GIS and web software technologies for filtering and pre-processing data and for implementing the data into a final, client facing web application.

The results of the report are discussed about the development process of the application, outlining theory related to WebGIS systems, the methodology and implementation process of the application, and the project management methodology utilized during the development process. Future directions and potential improvements are also discussed.

The resulting application can be interesting to government and local planning departments as more population residing next to public transport nodes can reduce traffic and pollution. The resulting application can also be of interest to commercial real estate search portals as it provides a novel search for interested buyers to search for properties, thus providing a point of difference from the existing competition.

[THIS PAGE HAS BEEN INTENTIONALY LEFT BLANK]

Train Station Proximity Finder

Table of Contents

List of Figures

List of Tables

Abbreviations

CCPM – Critical Chain Project Management
GIS – Geographic Information System
GML – Geographic Markup Language
GRASS – Geographic Resource Analysis Support System
OSRM – Open Source Routing Machine
SAD – System Architecture Design
SRID – Spatial Reference System Identifier
TOD – Transit Oriented Development
TSPF – Train Station Proximity Finder
XML – Extensible Markup Language

1 Introduction

Geographic information systems (GIS) are systems that allow users to visualize, interpret, manipulate, and manage spatial data. These capabilities are important for many disciplines ranging from agriculture to astronomy to real estate, and with an ever-growing population, real estate is more now than ever a vibrant industry.

This report outlines the process of creating a web based GIS solution to provide alternative search capabilities to the traditional property search options by utilizing spatially enabled datasets; and performing spatial queries relative to a buyer's choice of public transport hub.

A multitude of factors affect a buyer's decision when purchasing property. These include financial, lifestyle changes and environmental factors. Whilst financial factors remain a major aspect in the property purchase decision process, lifestyle and housing mobility issues brought on by life cycle changes, and environmental factors are becoming increasingly important to buyers.

For most households, the purchase of a property tends to be the most valuable purchase in their lifetime (Tsatsaronis & Zhu, 2004). As such, property purchases tend to be an investment, and a capital gain is usually expected at the end of the holding period. Better access to public transport can positively affect house prices, and proximity to public transport hubs can have a greater impact on capital gains than proximity to quality schools (Gibbons & Machin, 2008).

Lifestyle and housing mobility issues, brought on by changes in an individual's lifestyle tend to be an important factor when deciding to upgrade or downgrade to a new home. These changes can include moving out of a parent's home, starting a family or changing workplaces (Scheiner, 2006).

Environmental factors are also becoming increasingly important during purchase decisions (Pickett-Baker & Ozaki, 2008). A highly important factor in the usage of more environmentally friendly types of transport, such as trains, is the availability of such facilities in the immediate area.

The application designed in conjunction with this report aims to utilize spatially enabled train station locations to facilitate citizens in finding properties for sale in close proximity to a transport hub of interest.

The current online property search options are limited. Despite over 90% of purchasers using an online medium to search for properties, current real estate search portals do not provide all adequate search functionalities required by property buyers. The primary search factor in an online search involves the user searching for a property based on single suburb, with filtering options available for pricing and house size, as shown in figure 1 overleaf.

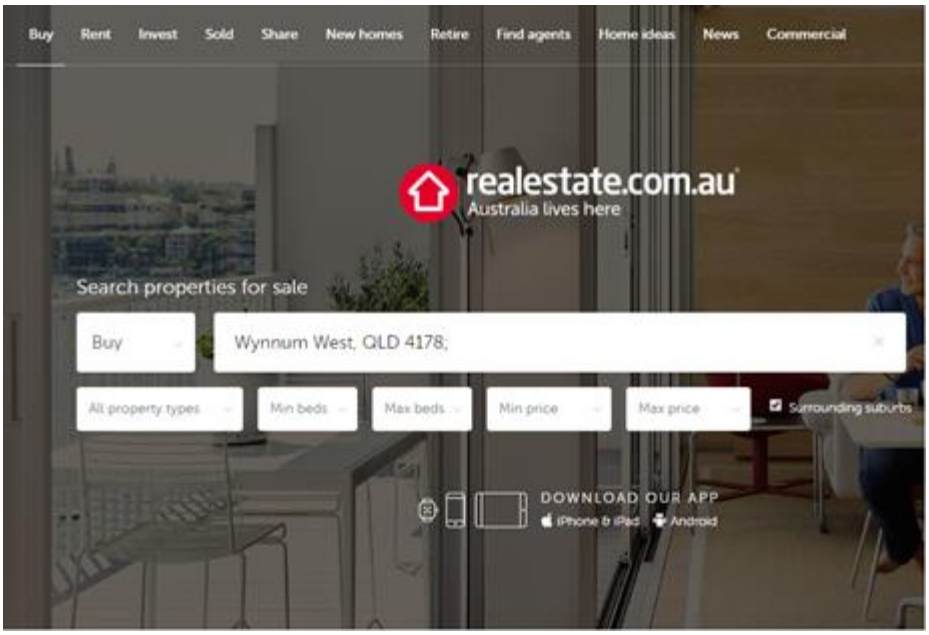


Figure 1: Example of a property search engine (realestate 2016)

While this is adequate to some, this search process, and results, do not provide the prospective buyer with enough information about the available amenities in the area. Some websites often display the proximity of public transport to individual properties for sale, however these do not show travel times or display results on a map, as demonstrated in figure 2.

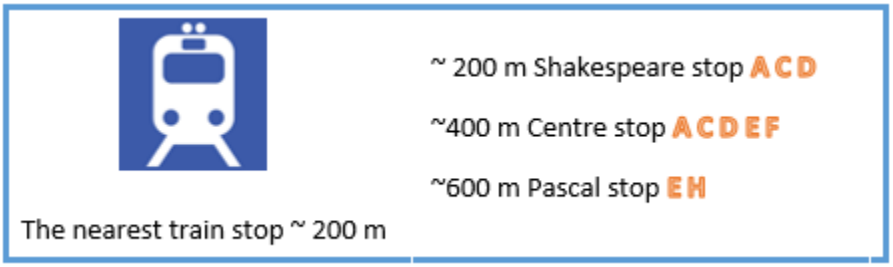


Figure 2: proximity of public transport to property for sale (Aruoda 2016)

The disadvantage of this method is it does not allow the user to visualize the journey and does not provide buyers complete peace of mind about the reliability of the information.

Other sites display travel times based on a specific destination and include travel times and a map, however this does not allow for properties to be searched by proximity to a public transport hub, as demonstrated at figure 3 overleaf.

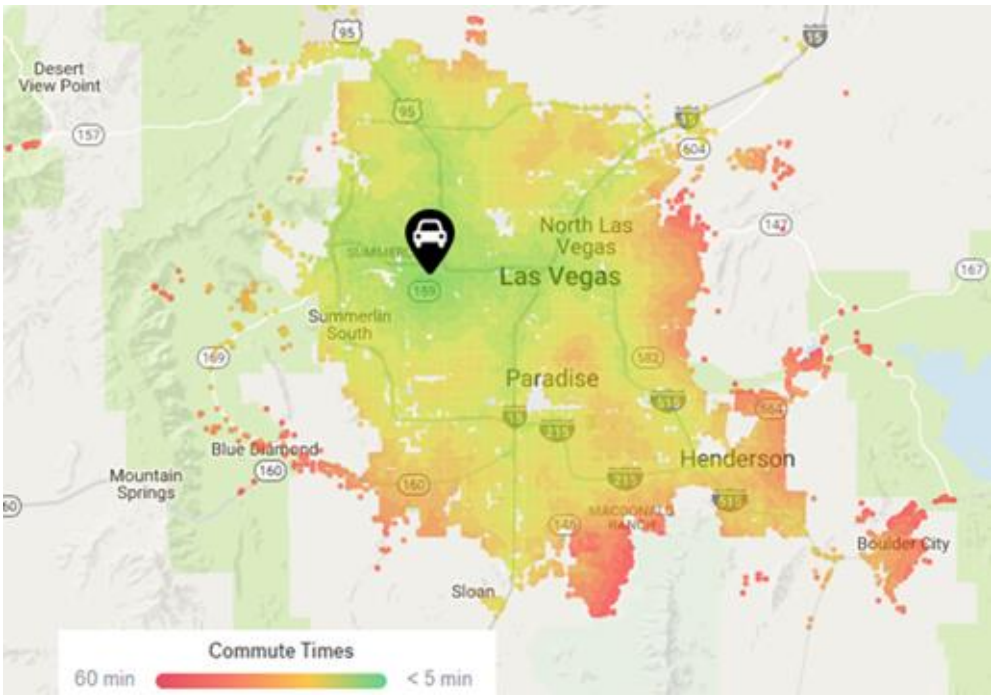


Figure 3: Commute times to a chosen destination (Trulia 2016)

Google Maps can provide a routing solution, as the property of interest can be entered as an origin, the public transport hub as the destination, walking selected as mode of transport, and the route and travel time is displayed as demonstrated in Figure 4.

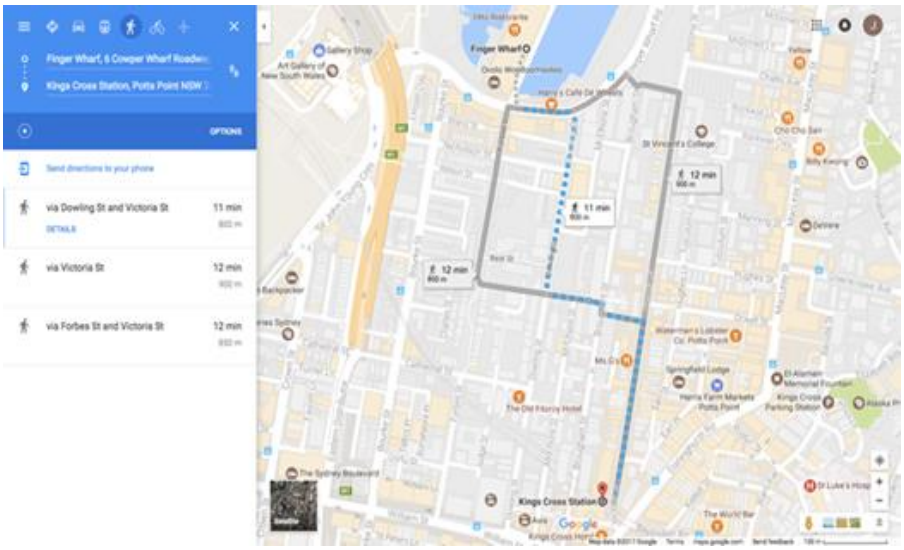


Figure 4: route and commute time from origin to destination (Google Maps 2016)

Although Google Maps can provide the necessary functionality, it requires the user to have prior knowledge about this function, as well as the requirement for the user to navigate away from the real estate website, and manually enter the necessary information.
Collectively, the examples above enable a user to find properties in proximity to the nearest public transport hub of interest, however we have been unable to find a single tool within a property search website we believe adequately performs this function.

We believe this application is necessary for several reasons. Firstly, from the perspective of climate change and town planning policy at a micro level, the proposed webGIS solution would facilitate finding a property in close proximity to public transport. Public transport usage is shown to be highly attractive due to reduced costs and travel times, and usage is shown to increase when amenities are available in the immediate area. Secondly, from a business perspective, the application can provide a point of difference to existing property search portals.

The following sections will explain the current research, introduce the research questions that are to be answered throughout the report, and finally, provide an outline of the report.

1.1 Current research

Several studies conducted in the early 2000's, a period in America when decentralization and urban sprawl was at a high level, demonstrated that public transport is critical in high density urban areas and a strong correlation exists with property and rental prices:

- 2007 study in Buffalo, New York, demonstrated that for every foot closer to a train station property values increased on average by $2.31 (Hess and Almeida, 2007).
- 2001 study demonstrated the positive factors of decreased commuting costs and the negative factors of rising crime rates caused by increased mobility of criminals, resulted in these factors having a critical influence on the relationship between house prices and train stations. Furthermore, the level of importance these effects have alternates depending on the area's distance from downtown and the average income of the neighbourhood (Bowes and Ihlanfeldt, 2001).
- 1996 study at Washington DC, Metrorail stations demonstrated a significant decline in property value as distance increases from the Metrorail station – 2.50 % decrease for each tenth of a mile (106m) (Benjamin and Sirmans, 1996)
- In 1994 a study at Boston, MA demonstrated the influence of the commuter rail service on single family residential property prices before and after a train station is constructed. The study concluded an increase in residential property values by 6.7% due to a local train station being present (Armstrong, 1994).

These studies demonstrate that the proximity of train stations is important for a variety of socio-economic factors, with follow-on influences on determining property values.

1.2 Problem statement

The aim of this report is to develop a webGIS application that enables citizens to determine properties for sale within walking distance of train stations.

1.3 Research questions

1. **Data Selection** - What data and data quality could be used in the design and implementation of a web based application to determine the closest railway station to a property for sale?

2. **Technologies** - What technologies could be used to design and implement a spatially enabled property search engine?

3. **Network Analysis** - What is the best route finding method to find the nearest train station to properties for sale?

4. **Remote Group Work** - How to best coordinate group work while working remotely?

1.4 Report Structure

This report has been structured to address a problem. Research questions have been identified based on the problem; with the title of each question displayed in **bold** to identify it in the report.
The report has been structured in the following way:

- **Theory** - validates problem statement then discusses theories for each research question to demonstrate understanding and identify options;
- **Methodology** - describes the procedure for developing the application;
- **Implementation** - describes how the application was developed using an iterative procedure;
- **Discussion** - discusses each research question in the context of the TSPF application;
- **Conclusion** - describes solution to problem statement.

2 Theory

2.1 Climate Change and Planning Policy

The science of climate change is now clearly established and the majority of western countries are taking measures to mitigate issues arising from rising sea levels, rising temperatures and increased extreme weather events. (Intergovernmental Panel on Climate Change, 2015). At a micro level, town planning policy and sustainability goals set out by city councils can help in reaching a country's sustainability goals.

Policies tend to vary, however Jabareen (2013) identified that land use planning, and building and design codes make a highly beneficial contribution to the resilience of a city.

Sustainable transport has also been identified as a feature in designing a sustainable urban form. This concept focuses on a micro level and on urban design (Jabareen, 2013). Access to sustainable transport should promote usage of public transport, should reduce traffic congestion and commute times. This can be achieved through the encouragement of Transit Oriented Development (TOD) (Carl, 2000). Cervero (2006) and Schlossenber & Brown (2004) noted that TODs are a highly viable model for increasing integration between public transport usage and land-use. However, when developing for transit oriented communities, it is important to consider other key factors that influence the house purchasing decision. These factors are related to housing mobility and include major life events, availability of funds, and lifestyle and environmental considerations.

2.2 Housing Mobility and Life Changes

Housing mobility refers to an individual's long-term mobility, including housing type and choice of location (Scheiner & Kasper, 2003). The term defines how an individual interacts with the physical space in relation to their residential housing choices.

Housing mobility is strongly correlated to several events in the life cycle, e.g. Moving out of parents' home, starting a family, changing workplaces, or retirement (Scheiner, 2006).

Scheiner also noted that the mode of transport used for travel after relocation is highly dependent on the available facilities in the area. Increase in the use of public transport and foot trips were noted when residents relocated to an inner-city area which was known for the quality of its public transport. Smets (2000) also noted similar increases in public transport usage when residents relocated to areas with better connectivity.

2.3 Factors Influencing Purchasing Decisions

2.3.1 Financial

In general, the trend in most western countries remains focused on 'upward housing mobility' (Han et al, 2016). This refers to most the population being interested in moving from renting to owning property. However, in recent years this has proven to be more difficult. The global financial crisis, an influx of investor spending post recovery, and an undersupply of housing in most major metropolitan areas has forced new homeowners into the outer suburbs instead of inner city. These outer suburbs are generally less connected, with little to no reliable public transport available.

For most households, the purchase of a house tends to be the most valuable purchase of their lifetime (Tsatsaronis & Zhu, 2004). As such, most new buyers consider property purchasing an investment, and a capital gain is usually expected.

Better accessibility to public transport can positively affect house prices. This is primarily noticed due to reduce commute times (Gibbons & Machin, 2008). Gibbons & Machin concluded that accessibility to a metro station could have an even greater impact on house prices than proximity to quality schools.

It is of note however that once a purchaser establishes a budget, housing in several locations would still be available for purchase. At this point in the decision-making process other factors, such as lifestyle and proximity to daily activities are then considered.

2.3.2 Lifestyle

Bauer, Holz-Rau, Scheiner and Walther (2003) noted that there is evidence that growing up in certain environments can have an influence on future locational decision. This leads to several adults preferring to live in the same place, regardless of changes in their daily commute or spatial activities. Kalter (1994) notes that this leads to long-distance commuting becoming the preference over migrating to new areas. This is of importance to town planning, as it requires future planning for public transport facilities to provide interconnectivity between towns and suburbs, as residents are choosing to live farther away from workplaces, for lifestyle choices and not just financial.

2.3.3 Environmental

In recent years, individuals have shown an increase in environmental concerns. In the developed countries, especially, climate change, energy saving and environmental benefits have begun impacting purchasing decisions (Pickett-Baker & Ozaki, 2008). This trend is seen in housing purchases and housing choices where energy saving and environmental benefits are highly valued by consumers (Banfiet et al, 2008).

It has been hypothesized that an increased popularity in car usage can be highly difficult to reverse (Kramer-Badoni & Kuhm, 2000). This was mainly attributed to the high degree of autonomy that private car usage provides, and eliminating private car ownership would mean social disintegration (Kramer-Badoni & Kuhm, 2000). However, Scheiner notes that this hypothesis does not consider the users' shifts towards more environmentally friendly modes of transport in recent years. However, a highly important factor remains the availability of public transport facilities in the immediate area.

2.4 Property Search

A 2012 study from the National Association of Realtors found that over 90% of purchasers searched for property online during the buying process, with real estate related searches having grown 253% between 2008 and 2012.

The current search process is limited. The primary factor in an online search is to search based on a single suburb, with filtering options available for pricing and house size. This simple process however is intuitive and less tech-savvy persons can easily utilize the search engine.
The web GIS solution proposed in this report is of a higher complexity and may require additional technical knowledge from the end user.

The following chapters outline the theory and methodology utilized in implementing a location based property search web GIS solution, with a focus on proximity to train stations.

2.5 Data

Open Government Data (OGD), Public participatory geographic information systems/science (PPGIS) and Volunteered geographic information (VGI) have emerged as important data contributors over the past decade (Hansen et al, 2013). These data sources provide the place component for built and natural environment data, however due to the methods by which this data is generated challenges arise regarding data reliability and therefore usefulness.

2.5.1 OGD

OGD is spatially referenced data made available for open use and can be freely used, reused and redistributed. Production is taxpayer funded and does not follow traditional pricing models where revenue is generated by selling data; therefore, benefits are realized through improved efficiency and cost savings to society (Hansen et al, 2013). It is the most reliable data source in most developed nations due to open data initiatives such as the INSPIRE directive and the EU Directive for the Re-use of Public Sector Information (Hansen et al, 2013), as it conforms to standards and contains reliable metadata. Such data would therefore be most reliable for developing an application for communicating spatial information.

2.5.2 PPGIS and VGI

Where Spatial Data Infrastructures (SDI) are inadequate to enable OGD to be a spatial enabler, as in many developing nations, PPGIS and VGI can fill this void.

PPGIS is a web based GIS service where citizens participate in data collection and legally binding policy creation alongside government. The concept started in poor areas and developing countries, where indigenous and local leaders were consulted to extract their knowledge. It is a bottom up process to collect data, and top down to disseminate it.

VGI is user-generated content that has increased significantly with advances in ICT, such as GPS enabled mobile phones and car navigation systems (Enemark and Rajabifard, 2011). Users contribute georeferenced material on a voluntary basis, with the incentive to make a contribution that benefits society through spatial enablement. The success of OpenStreetMaps and Google Maps highlight the demand for such services in the private and public sectors (Hansen et al, 2013).

PPGIS and VGI go beyond traditional tenure and cadastral methods that are often poorly established and maintained in developing countries. They can fill the data void to provide data for communicating spatial information where such data is not available from OGD sources. These sources should be used with caution however, as they often do not have complete metadata and the source cannot be verified.

2.6 Technologies

2.6.1 Software

One of the first considerations when developing an application is what type of software to use - commercial or open source.

Open source software is freely accessible with the ability for the user to modify the source code and thereby tailor the program to their needs by adding new features or fixing existing bugs. Open source is often favoured by students because it offers a freely available platform to practice and thereby gain knowledge about real world programming; as well as gain feedback, comments, critique, reviews, discover mistakes. It can provide greater security and stability compared with commercial software, as well as remain up-to-date, because programmers can rapidly update, fix and correct code, resulting in faster growth. Another advantage is stability, especially for long-term projects (OpenSource, 2016).

Users of commercial software tend to value reliability and good customer support, they have no interest in working on problems that aren’t related to their work. So, in a sense commercial software can be looked at by some as having a guarantee that all of the user's working hours will be spent on finances not programming (Chubirka, 2014).

2.6.1.1 GIS Software

A Geographic Information System (GIS) is designed to store, retrieve, manage, display, and analyse all types of geographic and spatial data. Spatial data is stored as points, lines, or polygons; or as raster images. In the context of a city map, buildings could be stored as points, roads stored as lines, and boundaries stored as polygons; while aerial photos or scanned maps could be stored as raster images (Chang, 2008).

2.6.1.2 QGIS and ArcGIS

The two most popular and capable GIS software applications are ArcGIS and Quantum GIS (QGIS). Both applications are similar in terms of functionality - each can intuitively join tables, process a wide variety of data types, supports a wide variety of coordinate reference systems (CRS), and offer extensive plugins to expand capability (GIS Geography, 2016).

The major difference between the applications is that QGIS is an open source application and is free to use, whereas ArcGIS is commercial software from ESRI and requires an annual licence fee. Depending on the subscription however, ESRI provides an extensive suite of GIS services in addition to ArcGIS, such as ArcCatalog and ArcGIS for Server. Another difference is that QGIS is available on all 3 major operating systems (windows, Linux, mac), whereas ArcGIS is only available on one operating system - Windows (GIS Geography, 2016).

2.6.1.3 Grass GIS

Geographic Resource Analysis Support System (GRASS) is open source multi-platform GIS software developed by the U.S Army Construction Engineering Research Laboratory (CERL) in the late 20th century, in the early days of computer technology. The original reason for creating this landscape analysis tool was to manage U.S department of defence installations. These installations include millions of square kilometres of land required for military training and testing. Like a lot of software created by the military the software was eventually brought into the public sector. In the early days of GIS technology GRASS focused on raster data analysis, which competing software did not provide at the time (Neteler & Mitasova 2013).

GRASS can produce geospatial data, analysis and mapping. It be work with 2D and 3D raster data, includes a topological 2D and 3D vector engine with SQL based attribute management, and vector network analysis functions. GRASS provides multiple modelling algorithms, 3D visualization and image processing capabilities relevant to LIDAR and multi band imagery (Neteler et al. 2012).

2.6.1.4 OGC Standards

Web map service (WMS) and Web feature service (WFS) are standards defined by the Open Geospatial Consortium (OGC) for sharing geographic data (Michaelis & Ames, 2008). Both services use eXtensible Markup Language (XML) for the requests, due to its extensive compatibility for client and

server requests. The choice of service depends on the required speed of the service, and whether the end product is solely for end user visualisation, as is the case for WMS, or whether geospatial operations need to be performed on the data, as is the case for WFS (Michaelis & Ames, 2008).

WMS is a service which a client requests a map image from a server. The image is returned without any geographic information, as the service is solely for end user interfaces (Michaelis & Ames, 2008). Several request types are supported by the server which are communicated in XML:

- **GetCapabilities** - returns the available data on a server (bounding box and coordinate reference systems);
- **DescribeLayer** - provides details on a specific layer; and,
- **GetMap** - returns an image of the map in raster format.

WMS advantages:

- Much faster at loading because images are pre-rendered;
- Does not require a GIS platform to view data, therefore easier to use and accessible to a broader spectrum of society; and,
- Uses CSS which can be used to influence the appearance of the end product.

WMS disadvantages:

- Cannot change the data or resolution; and,
- No access to raw data, so not possible to conduct analysis.

Web feature service (WFS) delivers vector data as a 'streamed' feature, and requires a GIS platform to view data. Similar to WMS, it uses XML to communicate requests and returns the same information for the GetCapabilities and DescribeLayer requests, however for the GetMap request vector data in Geographic Markup Language (GML) is returned, rather than an image. GML is a type of XML, which is readily parsed by machines and easy to read by humans which makes debugging easier, however also results in the transfer of large amounts of XML text which can be data intensive (Michaelis & Ames, 2008).

WMS advantages:

- Raw vector data allows for analysis and manipulation of features;
- Transactional version allows modification of data - add, change, delete, lock; and,
- Works at different resolutions and on different devices.

WMS disadvantages:

- Requires a GIS platform to view data which can be complicated to use; and,
- More data intensive.

2.6.1.5 GIS servers

A GIS server stores geographic data remotely, processes spatial queries, generates maps and delivers responses to a web client (Peng et al, 2003). It is capable of a variety of GIS functions such as translating geometries into visual representations, and sending data to the end client in a variety of formats such as feature data or a rendered map image.
Open source and commercial GIS servers are available to host a web map, which are described below.

2.6.1.5.1 ArcGIS Server

ArcGIS Server is a commercial software from ESRI, and is part of the ArcGIS suite of software. It is suitable for advanced WebGIS services and has multiple functionalities such as geodata capabilities, server assist for teamwork, WebGIS distribution, compatible with a variety of databases (eg. PostgreSQL) and capable of processing multiple users simultaneously.

In addition to use with ArcGIS Desktop, ArcGIS Server can be used in web applications and combined with GIS content from consumer maps such as Google and Bing (ESRI, 2016).

2.6.1.5.2 GeoServer

GeoServer is an open source java based software. The functionality of the server enables users to view and manipulate geospatial data, create maps and share data (Geoserver, 2016). Key functionality of GeoServer includes:

- Implements both WMS and WFS standards. WMS allows creation of maps in a variety of output formats, and WFS enables the sharing and editing of data used to generate maps;
- Integrates OpenLayers, a free mapping library, which facilitates easy map creation;
- Integrates with existing mapping APIs (eg. Google Maps), and can connect with traditional GIS architectures (eg. ArcGIS); and,
- Open source means rapid bug fixes and feature improvements.

2.6.1.6 API

An Application Programming Interface (API) is a set of protocols and tools for developing application software. It assists the developer by proving the building blocks for an application. In the context of GIS applications, APIs can provide access to functionality such as base maps and various buttons and tools (Blanchette, 2008).

2.6.1.6.1 Openlayers

Openlayers is a JavaScript API which allows a developer to insert a dynamic map in most modern web browsers. Since it is a purely client-side application, it is independent of any server, and is capable of

supporting various mapping APIs such as Google, Yahoo, OGC WMS, OGC WFS, and Text layers (OSGeo, 2016).

OpenLayers can be integrated into a web page by embedding the OpenLayers API inside any block level element of HTML script, and including a script tag pointing to the OpenLayers library (OSGeo, 2016).

Key characteristics of OpenLayers include:

- Only a basic knowledge of JavaScript is required for use;
- Provides OpenLayers base maps;
- The JavaScript library is open source and therefore free for any user;
- It is a pure client-side application and can therefore be driven easily with any server language such as PHP, and PERL; and,
- Maps can be embedded directly into an HTML file.

2.6.1.6.2 Leaflet

Leaflet is a popular open-source JavaScript API for web map creation and development. It is widely used due to its simplicity, flexibility and compatibility with other devices.

Leaflet provides the capability to zoom, pan and select feature layers. It is capable of basic tasks such as converting data to map layers and mouse interactions; and is extendable with API plugins (Woodruff and Mullins, 2016).

Key characteristics of Leaflet include:

- Only a basic knowledge of JavaScript and HTML is required for use;
- Does not provide data - only provides a framework for displaying and interacting with data, but user has to provide it; and,
- Does not provide GIS functionality, however can be integrated with GIS tools.

2.6.1.7 Database Management System (DBMS)

A Database Management System (DBMS) is a system or a software for managing and creating databases. The DBMS software allows users to view, edit, create and organise data. This is obviously vital to the construction of any application and so picking the ideal type of DBMS software is important. Two of the most widely used and successful geospatial databases are Oracle spatial and PostGIS (Shukla et al, 2016).

- **Oracle and Oracle Spatial:** Oracle spatial is a geospatial database which is created by Oracle. It provides a schema known as MDSYS. This determines required syntaxes and semantics of data types supported by oracle spatial. It supports all major operating systems, but requires a subscription fee to use (quite a substantial fee). In an experiment performed by (Shukla et al,

2016) in the Institute of Technology Nirma, found that oracle when compared to PostgreSQL is considerably slower when loading data.

- **PostgreSQL and PostGIS:** PostGIS is an open source software program that adds geospatial capabilities to the PostgreSQL relational database. PostGIS abides by the SQL specifications laid out by the geospatial consortium. PostgreSQL and PostGIS are both completely free, incredibly reliable and user friendly (Shukla et al, 2016).

2.6.2 Web Application

A web application is a client-server software application, in which the client runs in a web browser and accesses data from a server. The client interprets the user's request and sends a request to the server, which sends a return to the client that displays in the browser. Web applications typically have similar functionality to desktop software applications, and include many examples such as office software, project management, and computer aided drawing (Kavourgias, 2014).

Web applications are developed in two stages: front-end and back-end. Front-end describes 'what the client sees', and combines languages such as HTML, CSS and JavaScript. Back-end describes the 'server side' work by the developer and involves updating and modifying the site (Kavourgias, 2014).

Web applications have the following advantages over traditional client-server models:

- Relieves the developer of building a client for a specific operating system, since the client runs in a web browser through the use of web documents that are compatible across a wide range of operating systems;
- Use a combination of server-side script, such as ASP or PHP which is responsible for the presentation of information; and client-side script, such as HTML or JavaScript, which stores and retrieves information and enables web applications to perform quickly; and,
- Client updates may occur each time the web page is visited.

2.6.2.1 Python

Python is a high-level, object-oriented and interpreted language. High-level means that before the program starts, the compiler converts code into low-level language (knows as assembly language). It is more time consuming, but has several advantages. Firstly, it is easier to program on high-level language because it is more human readable than low-level, which assists for faster, less confusing and correct code. Secondly, it is portable, which means it works on different computers without, or with very small, changes to the code.

The working scheme is demonstrated in Figure 5. Source code goes to compiler, which interprets the code and the converts it to object code, which is known as executable. After compilation, it is possible to execute it many times without compilation (Elkner et al, 2010).

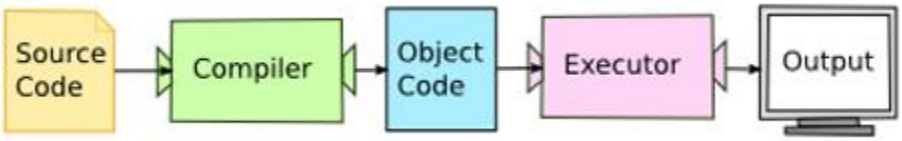


Figure 5: The Python process - Converting high-level language into low-level language (Elkner et al 2010)

Python is very popular because of the easy syntax and large library. The example at Figure 6 demonstrates how to calculate all odd numbers from 1 to 100 and their sum. Python uses indentation, which differs from others languages which use brackets and keywords.

```
sum = 0
for i in range (0,100):
    if i % 2 != 0:
    sum = sum + i
print i, sum
```

Figure 6: Example of Python code (Python software foundation 2016).

A number of Python web frameworks are available to support the development of web applications. These web frameworks usually provide libraries and promote code reuse in order to assist and simplify web application development. (Docforge, 2016) Some notable python web framework examples are Flask, Django and web2py.

2.6.2.1.1 Flask

Flask is one of the most popular Python micro web frameworks. Its functionality can be improved and expanded on with the various flask extensions available.

Flask can be easily integrated with an open source technology stack as it provides accessibility to a PostGIS database. Spatial database querying can be achieved using the GeoAlchemy object relational mapper (ORM). The ORM provides a more python friendly way to query the database than through direct SQL queries.

2.6.2.2 PHP

Personal Home Page (PHP) is a general-purpose scripting language used by web developers that can be implemented into HTML. The syntax is similar to C++, Java, and Perl. An example of how PHP appears in HTML is illustrated at Figure 7.

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>Example</title>
    </head>
    <body>
        <?php
            echo *Hi, I'm a PHP script! *;
        ?>
    </body>
</html>
```

Figure 7: PHP in HTML (Cowburn 2016)

The same code will be with a lot of commands and inputs if we will use C or Perl. But PHP has starts and ends brackets <? php and ?> they show where to go for PHP mode. The difference between PHP and JavaScript is that the code is made on a server, which generates HTML and then reaches the user. This way the user gets results, but without knowing the underlying code. PHP is appealing for a new programmer, because it is easy to learn and provides a large variety of advanced features.

2.6.2.3 HTML

The foundation of World Wide Web is based on Hypertext Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS). This computer language is used to create web pages. HTML is a mark-up language rather than a programming language. HTML is an abbreviation for Hypertext Markup Language (Fajfar I., 2015). Hypertext is an approach which is responsible for using webpages. Hyperlinks is a text which links user to another webpage (Shannon R., 2016). Markup assist for annotations which are categorizing and characterizing from plain text, which is combined by variety of annotations in a document. Markup hides instructions which controlling the program which displays text or actions. The instructions are not visible for the user. User sees only consequences of the code (Fajfar I., 2015). HTML allows us to create website and these webpages can be seen by anyone who use internet. So, HTML mark-up language is very powerful (Shannon R., 2016).

HTML structure is necessary to communicate with web browser (Duckett J., 2010). The main structure idea is shown on Figure 8.

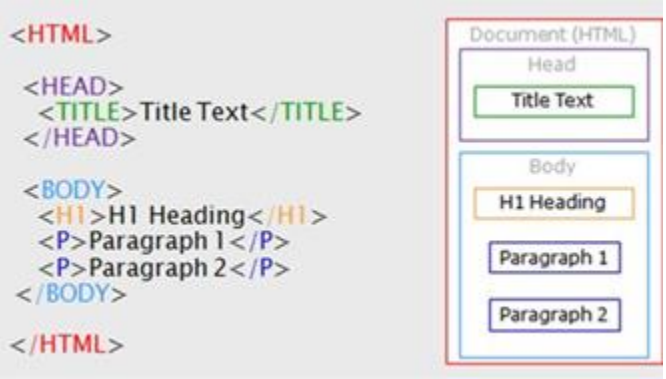


Figure 8: HTML structure (Akash, 2016)

In the structure there are four main attributes: <HTML>, <HEAD>,<TITLE> and <BODY>. Each document should begin with open <HTML> tag and finish with close </HTML> tag. <HEAD> element appeared after <HTML> and as well ends with closing tab </HEAD>. It contains all head elements. In this example in <HEAD> it is one element <TITLE>, but it can contain more than one. <TITLE> is visible in the top of the pager, as well, as bookmark of browser page, moreover, it helps indexing pages. <BODY> is one part after <HEAD>. This part has all attributes which are possible to use from attribute part. (Duckett J. 2010).

In the next chapter, we will go more detailed to cascading style sheets, which is responsible for styling.

2.6.2.4 CSS

CSS stands for cascading style sheets. Originally it was created for HTML styling, but now are using for more languages, SVG, XForms (Pihkala K., 2003). CSS is responsible for page style control, like font and lines size, width, colour, space and a lot of other components which makes webpage appealing for user. The CSS is continuously progressing by taking care by the World Wide Web Consortium (W3C). W3C is organization which is responsible for internet standards (Olsson M., 2014). CSS specification provides instruction/rules which tells how the content of document have to come out. For instance, it is possible to specify, that background colour is white, the content of all <p> elements is black using the time new romans typeface and all <h1> elements ought to be in green using the Calibri typeface. In other words, CSS accomplish rules with elements, which shows up on a webpage.

Figure 9 gives a short understanding how CSS rules work.

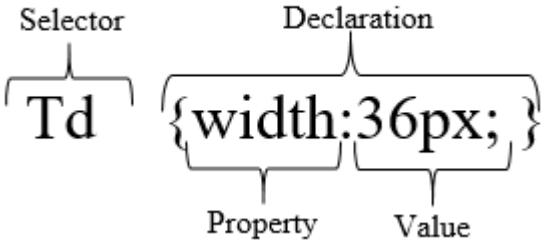


Figure 9: CSS rule (Duckett 2010)

This rule made from two main parts: the sector and Declaration. The sector provides which elements are applied for. The declaration is responsible for element styling. (Duckett J., 2010).
In the next chapter, we will focus on Java Script – language, which is used for web browsers.

2.6.2.5 JavaScript

JavaScript is Object – orientated (OO) interpreted program language (Flanagan, 2006). JavaScript is a language which is used for web browser. This makes JavaScript one of the most famous languages. D.Crockford in his book "JavaScript: The Good Parts claims, that this is the most desired language in the world. (Crockford, 2008).
JavaScript can be used for solutions like:

- Better and faster web page reactions according the user interactions to elements and hyperlinks
- Share small parts of database information with user friendly interface
- In relation to user needs, it regulates multiple – frame navigation, plug-ins, Java applets in the HTML document
- Assists data pre-processing before submission
- responding to user interactions dynamically and instantly by changing content and style. (Goodman, 2007)

2.6.2.6 Databases

A "Database" refers to a group of related data and they in which it is structured. In order to access and manipulate this data a "Database management system" or DBMS is implemented (Robbins,1994). DBMS's are described in 2.1.5.6 of this report. The following chapter will discuss spatial databases, structures, querying, indexing, and ER diagrams.

2.6.2.7 Spatial Database

A spatial database is a database system which is a large collection of data organized to allow for rapid exploration and retrieval of data, a spatial extension to this database system allows the database to interact with spatial data (Güting, Ralf Hartmut 1994). GIS spatial databases store data which

represents the real-world e.g. trees, buildings. The database represents this real-world data in three different models as seen in Figure 10 overleaf; conceptual, logical and physical (Sameth 2002).

These data models are an abstraction of the physical world. The conceptual model models the user's view, defines objects and their relationships and selects the geographic representation of this data (discrete or continuous data) (Sameth 2002).

The logical model attaches the data to relevant database types such as points, polylines, polygons etc. The structure of the geographic database is also laid out, detailing the necessary relationships such as regulations, topology and setting coordinate systems.

The Physical model deals with the database schema or the actual layout of the database (fields, attributes etc.) It is described in a formal language (SQL) which is supported by a data management system (Rigaux et al 2003).

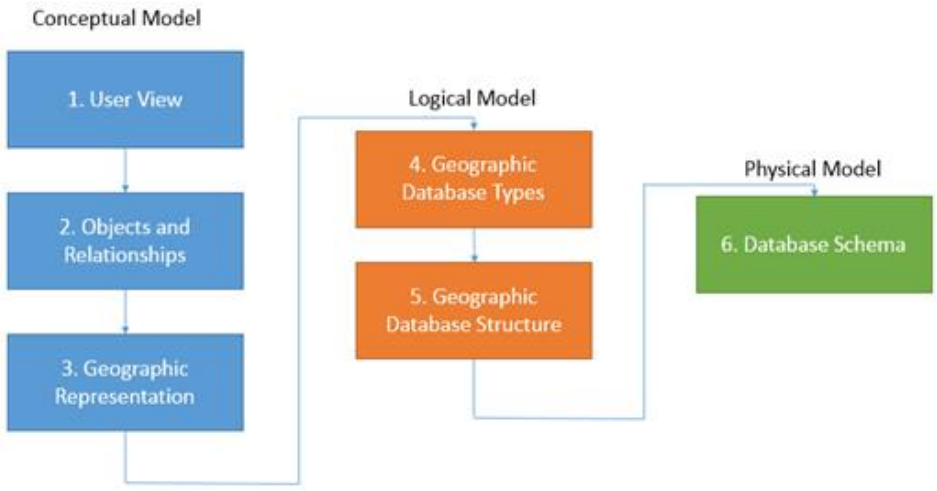


Figure 10: The different Models that make up a spatial database (Inspired by Kessler 2016)

2.6.2.8 Flat-file

The main concept of the flat file and relational model is a table(s) in which all data is contained. The smallest unit of data in a table is called a field (see below), fields are grouped together to form records, subsequently records group together to form records. Flat file databases store all the data in one table. This is suitable when you have a low number of records linked to a single topic, such as a person's name and phone number. But as the level of records increase, the database base gets a lot more complicated and unwieldy.

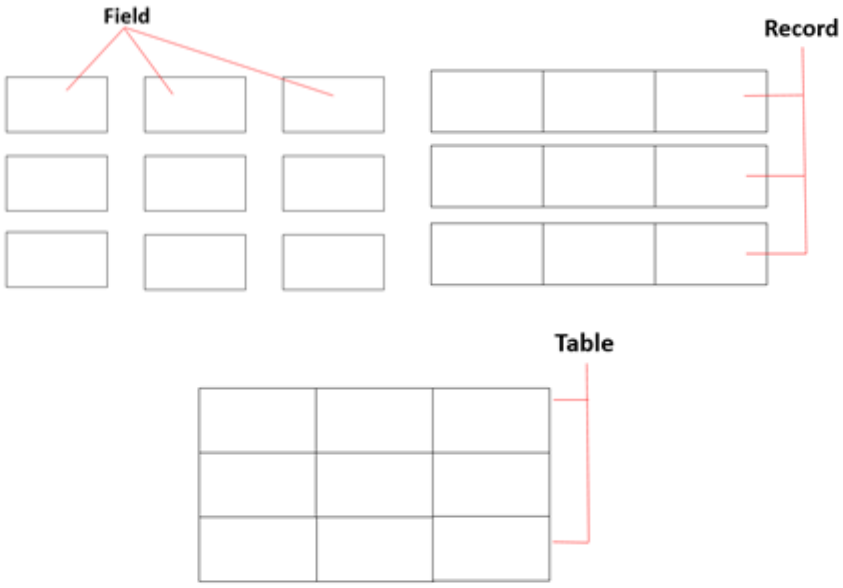


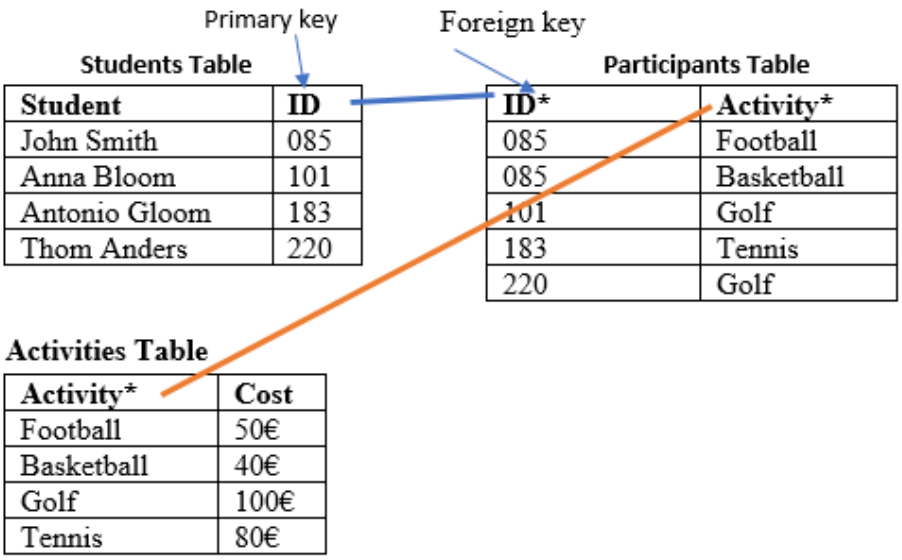
Figure 11: Breakdown of the Table structure (Inspired by IBM 2016)

2.6.2.9 Relational Database

Relational databases split a large amount of data into multiple tables. Each column in a table should be related to one topic such as user information which would contain columns with information such as height, age etc.

The numerous tables in a database are connected through keys (see Figure 12 below). Each table has the capacity to contain more than one foreign key and one primary key. Foreign keys are simply a primary key placed in one table to another.

The location and method of how the tables of data are stored has no relevance in adequately finding the data. Each table can be found using a unique identification, that the database then uses ID to locate the table in the background.



Student	ID
John Smith	085
Anna Bloom	101
Antonio Gloom	183
Thom Anders	220

ID*	Activity*
085	Football
085	Basketball
101	Golf
183	Tennis
220	Golf

Activity*	Cost
Football	50€
Basketball	40€
Golf	100€
Tennis	80€

Figure 12: Relational database and labels identifying primary and foreign keys

2.6.2.10 Indexing and querying

In order to find a specific unit of data the database must be queried or simply asked for certain information e.g. how many employees named ‘Mark’? The data base software will then search the entire database for any record that has the name ‘Mark’, this is done one at a time taking a considerable amount of time. To speed up this process an index is created. Database indexes are data structures that minimizes the length of time it takes to retrieve data in a database, at the expense of creating additional code and storage space to maintain the index structure, essentially the index cuts down the number of records that needs to be searched for the name ‘Mark’. B - trees are the most popular type of index structure, they allow for speedy searches and edits in logarithmic time (Comer, Douglas, 1979). The diagram below is an example of a b - tree index structure, if you were to search for all numbers less than thirteen you would look to the left side of the structure and so on.

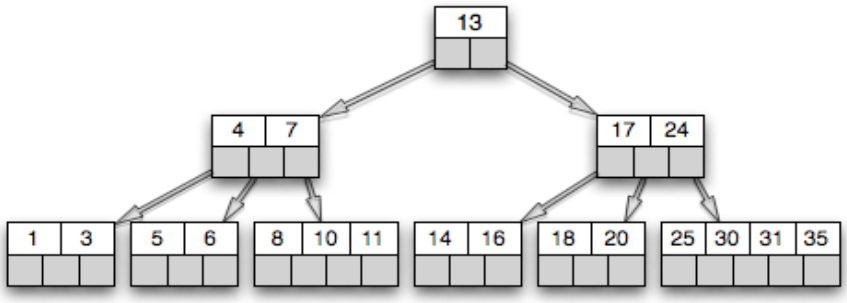


Figure 13: B-tree scheme (Farmer 2016)

2.6.2.11 Entity Relationship diagram

An Entity Relationship (ER) Diagram is type of flowchart that depicts the relationship between 'entities' within a system, entities can represent concepts, people etc. ER diagrams most common use is to design or repair relational databases. The ER diagrams consist of a multitude of differently shaped symbols, the shape of each symbol determines the main entity, its attributes and the type of relationships it has (ERD Tutorial, 2016).

ER diagrams can have different relationships. The main are:

- One to one (1:1) – one person have one passport
- One to Many (1:M) – one author may have write many books. The person is related to products, but all entities have only one specific connection
- Many to Many (M: N) – when some rows are related to many other rows in a table (Aberle, 2016).

The example of entity relationship diagram is in Figure 14. The diagram can be read as "Store owns a Video". On the diagram the relationship is "owns" and is marked by diamond shape, store and video here are entity class. "Purchase data" and "cost" are attributes.

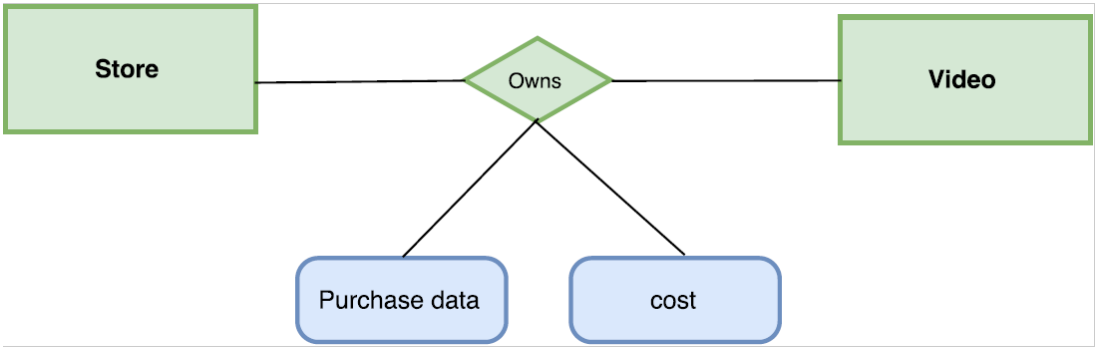


Figure 14: ER diagram example (Data Modelling with ER Diagrams 2016)

To be more precise we can show that relationship is one to many (1:M). One store can own a lot of videos, but the video can be owned only by one store. Now the diagram and the database is spatial, because of the geometry attribute, which contains spatial information about the store's location (Figure 15).

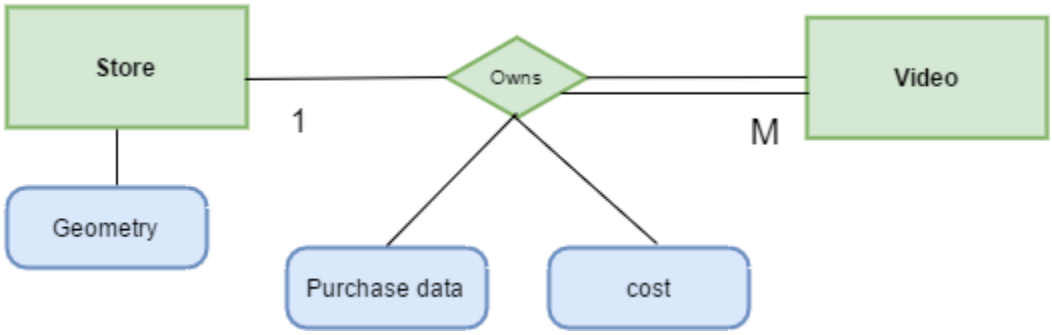


Figure 15: Spatial entity relationship (Data Modelling with ER Diagrams)

Spatial information is information about real world objects. Geometry usually can be simple or complex. Simple geometry stands for point, lines and polygons. Complex (Geometry collection) stands for multi point, multi line string (multi curve) and multi polygon (multi surface) (Figure 16). The point has 0 dimension (has only position), line - 1 (has length, but not thickness) and polygon - 2 (length and width). (Kessler, 2016)

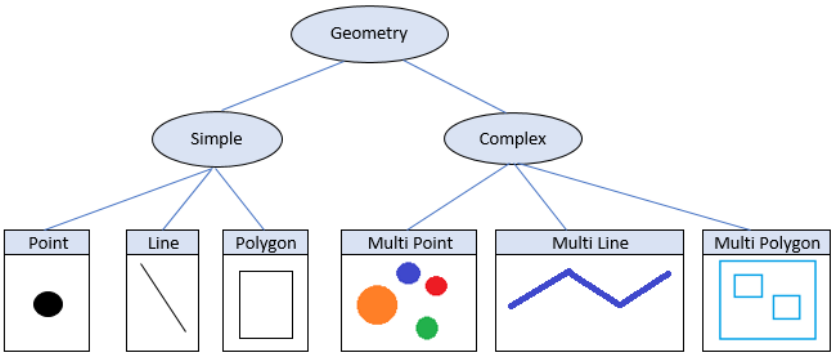


Figure 16: Types of geometry (Inspired by Kessler 2016)

2.7 Network Analysis

Network analysis is used to calculate the best route between two points, and may be expressed as physical distance or cost distance (Chang, 2008). Is conducted using software acting as an extension to a postGIS database, such as pgRouting or ArcGIS.

The analysis may be conducted on raster or vector data, with both types capable of determining a best route through minimising costs or shortest Euclidian distance, however only vector analysis is able to

determining the shortest distance for a journey on an existing network. Raster analysis generates results faster than vector analysis, but is not capable of providing true shapes of routes or route directions. Analysis of both data types use Dijkstra's algorithm to determine a path of least cost.

2.7.1 Raster

Raster analysis determines the path of least cost between two points. It requires a source raster, a cost raster, cost distance measures, and an algorithm for deriving the least accumulative path (Chang, 2008). Since cost can be assigned a variety of values, such as environmental or monetary factors, it is a useful planning tool for projects such as roads, pipelines, and canals (Chang, 2008).

The source raster defines the source cell and allocates a cost to that cell. All remaining cells have no value initially. The cost raster defines the cost to move through each cell, and may be an actual or a derived cost (such as aesthetic or environmental value). The cost distance measure is the cost of moving between nodes in each cell, either laterally (1.0 cell) or diagonally (1.414 cells). The cost to travel from one cell to another is determined by the equation:
D X [(Ci + Cj)/2], where D is the cost distance measure (1.0 or 1.414), Ci is the cost value at cell i, and Cj is the cost value at neighbouring cell j. Determining the least cost path between adjacent cells requires a simple calculation, however determining the least accumulative cost path is more complex and requires an iterative process using Dijkstra's algorithm (Chang, 2008).

In the first stage of Dijkstra's algorithm, cells adjacent to the source cell are activated, and the cost distance measure is calculated. The cell with the lowest cost distance is selected, and its value is assigned to the output raster. This process repeats, however each time a cell is reactivated (meaning it is accessible to the source through a different path) the accumulative cost must be recomputed. The end result is a raster output with the least accumulative cost to the source cell.

Four different types of output are possible:

1. A least accumulative cost raster;
2. A direction raster, showing the direction of the least cost path for each cell;
3. An allocation raster, displaying the relationship of each cell to a source cell based on cost distance measures; and,
4. A shortest path raster, which shows the least cost path from each cell to a source cell.

2.7.2 Vector

Vector analysis requires a vector based network that consists of lines connected by nodes. Examples of networks include roads, railways, bicycle paths and streams. Each line and node has attribute information for traversing each feature, such as speed limits on roads for lines and average turn time at intersections for nodes. This information is stored as an impedance matrix, which represents the cost for traversing between two nodes. If a node is not directly accessible from another it is assigned a value of infinity. The value assigned in the impedance matrix is dependent on the desired outcome of the

analysis; for example, emergency services require the quickest route so time is used as the measure, whereas a delivery driver may prioritise fuel consumption so distance is the measure.

Shortest path analysis finds the path with the minimum cumulative impedance between nodes on a network (Chang, 2008). Similar to raster analysis, Dijkstra's algorithm is used to determine the path of least cost using an iterative process, however uses an impedance matrix instead of a cost distance as in raster analysis.

2.7.3 Route Finding Applications

Open source and paid network analysis applications are available as an extension to PostgreSQL and PostGIS; such as pgRouting, ArcGIS extensions, and MapQuest.

2.7.3.1 OSRM

Open Source Routing Machine (OSRM) is a open source high spec routing engine for locating the shortest route from A to B (Luxen & Vetter 2011). It is tailored to be highly compatible with OpenStreetMaps (OSM) data allowing for OSM data to be imported and used easily (De Silva et al. 2014).

OSRM implements contraction hierarchies which creates shortcuts in the pre-processing stage. Contraction hierarchies can generate shortest path routes at a faster rate than Dijkstra's algorithm or various other routing techniques (Delling et al. 2009).

2.7.3.2 pgRouting

pgRouting is an open source vector based application that provides routing functionality to a PostGIS database. It is capable of conducting shortest path analysis with a variety of constraints, and has the ability to update data and automatically change attributes through the built-in routing engine (Singh et al, 2015).

It utilises a variety of different algorithms, such as various Dijkstra functions, One to Many Shortest Path, Traveling Sales Person, and Turn Restriction Shortest Path (TRSP); the choice of which must be done wisely to ensure the query returns the desired result quickly with minimal resource consumption such as memory and processing time (Singh et al, 2015).

2.8 System Design Strategies

Agile, Waterfall and Critical Chain Project Management are common methodologies for software development. Each have strengths and weaknesses that need be considered when deciding the appropriate method for a project, such as whether the project has a firm deadline and a clear scope. Each method is discussed below.

2.8.1 Waterfall

Waterfall is a software design methodology whereby development is conducted in a sequential, step-by-step process. It typically consists of 8 stages (conception, initiation, analysis, design, construction, testing, implementation, and maintenance), and requires each step is completed before moving on to the next. It is a deliberate, sequential process that produces a reliable result. The steps of the Waterfall process are illustrated in Figure 18.

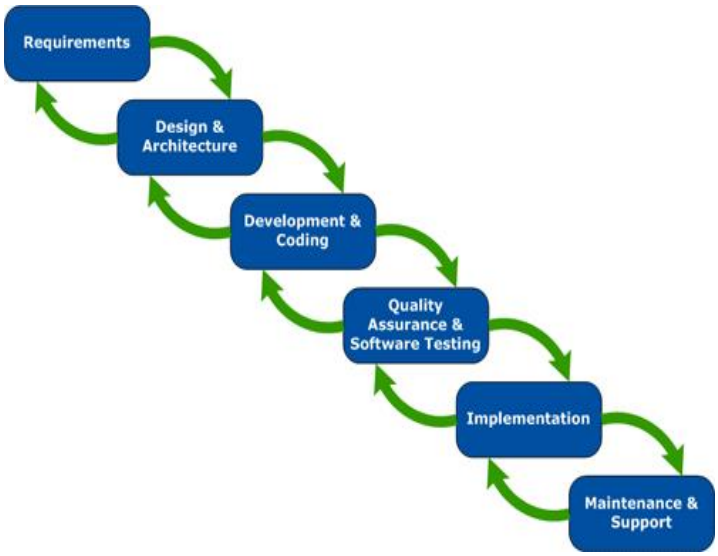


Figure 17: The waterfall model (Rising 2009)

Advantages:

- Requires a clear end goal and a structured plan, so time and cost can be accurately planned; and,
- Requires accurate record keeping, that provides solid documentation to build on for future projects, and new members can be join mid project.

Disadvantages:

- Expensive (time consuming) to return to previous stages to make changes;
- Requires a clear project scope, and expensive to make changes on the fly; and,
- Errors/problems discovered at the end.

The waterfall methodology should be used when there is a clear idea about the end goal, and precision rather than speed is the priority.

2.8.2 Agile

Agile software development utilises an incremental approach, whereby software is developed in small version during ‘sprints’. With each sprint session project priorities are defined and tests are run, which allows for detection of errors and to incorporate client feedback.

Advantages:

- Facilitates on-the-fly modifications;
- Easier to incorporate new features;
- Enables client input at the end of each sprint, thereby ensuring end product matches client requirements;
- Errors detected and resolved early; and,
- Product could be launched at any stage during development.

Disadvantages:

- Requires strong project management to ensure project stays on time and within budget; and,
- Project may be largely different from initial scope.

Agile methodology should be used when a client is flexible regarding the scope of the end product, and when a product is intended for an industry with rapidly changing standards, so they can be implemented. It should be implement in conjunction with a team that consists of a strong project manager, and skilled and creative developers. A basic methodology schema how does Agile work is illustrated in Figure 19 overleaf.

Figure 18: Agile method (Timothy 2016)

2.8.3 Critical Chain Project Management

Lawrence P. Leach on his book "Critical Chain Project Management" says, that for a project goal requires three necessary conditions: Time (schedule), Cost (Budget) and Scope (Quality) (Figure 20.)

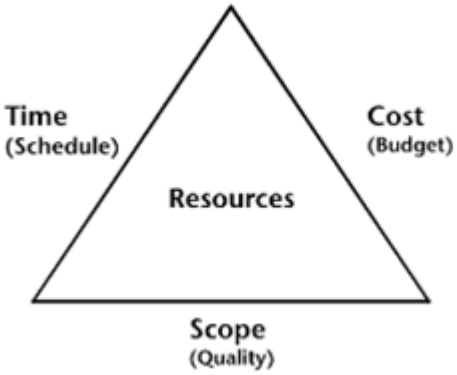


Figure 19: Satisfying the project goal requires three necessary conditions (Leach 2014)

For our project work we decided to use CCPM (Critical chain project management) methodology (Figure 20)

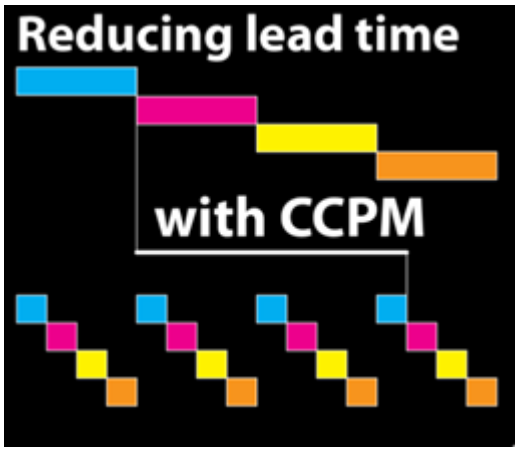


Figure 20: Critical chain project management methodology (Sian 2016)

It was developed by Eliyahu M. Goldratt. The strength of this methodology is attention to people, equipment and physical strength. CCPM allows to control all project with dividing it to smaller parts which are flexible with each other.

The key points are:

- Helps to avoid multitasking, which often is not effective. People have to focus just on one project task and finish it before moving further.
- You have to be aware of information and material unavailability. Ensure that all needed information and materials are available and can be taken without any restrictions.

We will work with both project task and non-project work tasks. This methodology enable us to focus only on one of them and helps to understand which one is more important and have to be done before (Goldratt, 2002)

2.9 Remote Group Work

With improvements in internet connectivity, working remotely has become increasingly common amongst students and businesses (Barber et al, 2009). Although this has facilitated sharing of information, working remotely presents several challenges.

2.9.1 Communication

The benefits of face to face group meetings become obvious when working remotely, especially when several stakeholders are involved. Where a conversation between two people involves back and forth dialogue, conversations between more than two people requires more formal processes to ensure no one speaks out of turn. Here visual cues become important, and emphasises the value of virtual group meetings which can be conducted on platforms such as Skype.

2.9.2 Coordinate writing report

Although effective communication can alleviate the complexities with multiple group members drafting a report, complications arise when integrating the content in a single document, which typically involves installation of client and server side applications. Due to the complexities involved with setting this up, email exchange of documents with annotated changes is a popular method to collaborate (Dekeyser & Watson, 2007).

The emergence of online collaboration platforms such as Google Docs however, have alleviated these problems by providing a simple to use application that runs in a web browser, and automatically updates changes to display the most up to date version.

2.9.3 Sharing large files

When developing an application with a large data sets, issues arise when sharing the data with all group members as traditional sharing methods such as email attachments are typically unable to support files over 2MB.

Many online file hosting services are available for such purposes, which distribute links to all stakeholders and allow data sets to be downloaded. These services differ through the storage space provided, upload/download speeds, helper applications, and paid/free.

Dropbox and Google Drive were considered for this project, as both are free services and provide adequate data storage.

3 Methodology

'Proper GIS planning is the most important investment any organization can make in building a GIS' (Peters, 2014). System Architecture Design (SAD), illustrated at Figure 22, is a process developed by Esri to facilitate the development of GIS solutions. It defines several critical stages to determine existing and projected user needs, and hardware and network solutions (Peters, 2014).

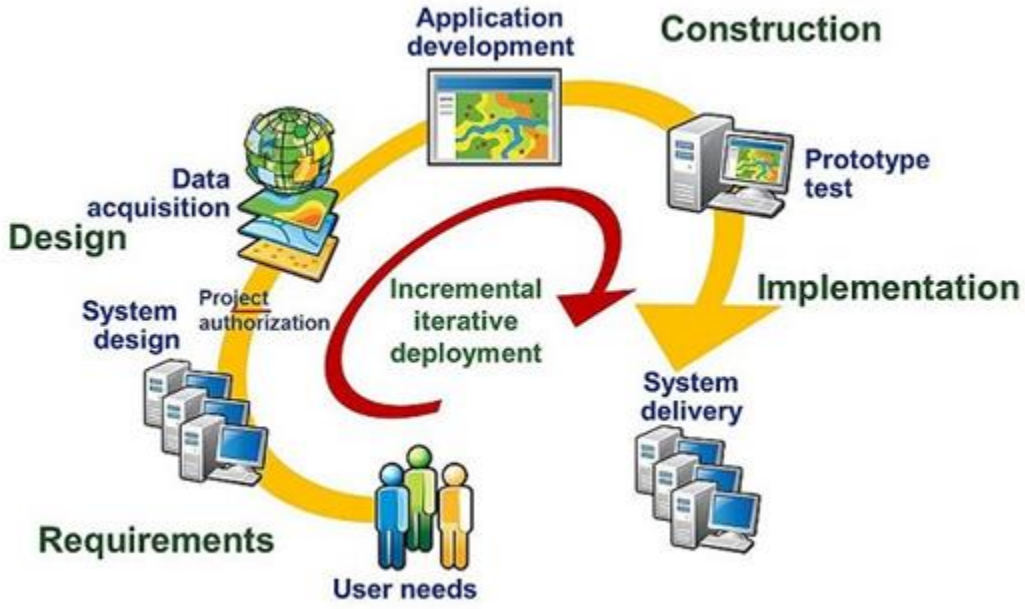


Figure 21: System Architecture Design incorporates several critical deployment stages to support successful implementation of a GIS solution (Peters 2006)

Initially requirements are defined, which are based on user needs and the anticipated system design requirements to meet these needs. System design is then developed, based on anticipated infrastructure upgrade requirements, network communication requirements, hardware and software procurement requirements, and anticipated data requirements. Typically, at this stage of a project, business decisions based on funding and procurement are required to proceed to the next step, the construction phase. During the construction phase data acquisition and database design commences, as well as prototype testing and determination of delivery targets. The final stage is implementation, where the system is delivered and the training and maintenance cycles commence.

However, certain stages of this process, such as determining user needs as demonstrated at Figure 23, are ‘exceptionally complex and very cumbersome to use for even a skilled and accomplished GIS professional’ and ‘the ESRI recommended preparatory course represents significant time and expense (3-days and $1605)’ (GeoNet, 2016), therefore it is proposed undertaking this exact process is out of the scope for this project.

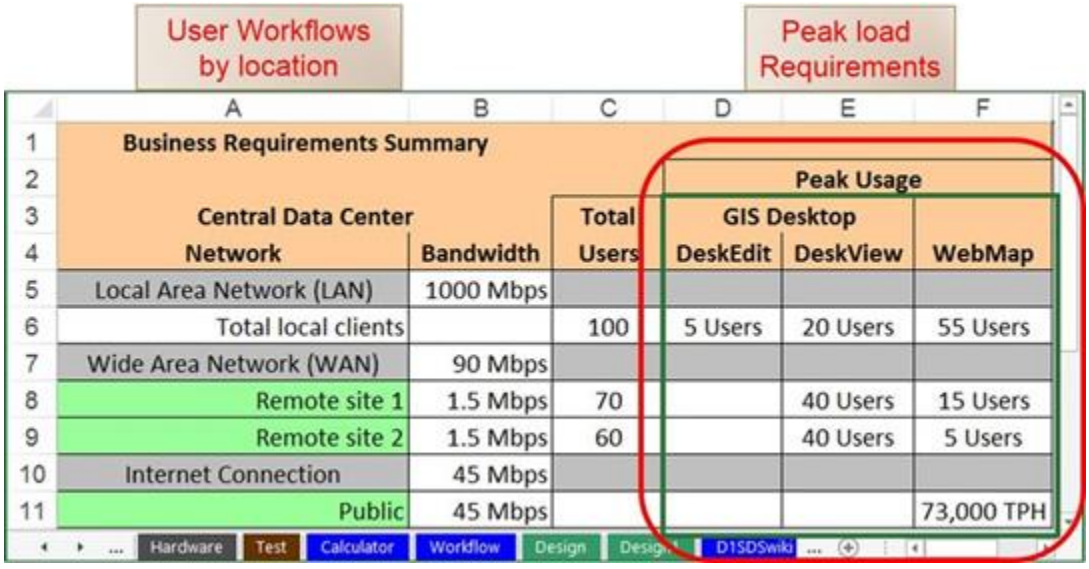


Figure 22: User Needs Template, as part of the Capacity Planning Tool, is recommended by Esri for determining system requirements (ESRI 2016)

Instead the structure of the Esri SAD has been followed, but the process itself has been re-defined to suit the scope of the project, which is to construct a prototype operating on a local system to determine the nearest train station to a property for sale.

The requirements and design phases of the TSPF application are described below. Construction and implementation is outlined in the Implementation chapter.

3.1 Requirements

Requirements for the application were determined based on user requirements and anticipated system demand which are outlined below.

3.1.1 User

‘User needs must be identified before completing the system architecture design’ (Peters, 2016). The first stage of this process was to identify the user, which was accomplished through group discussions and literature reviews, and then establish user needs.

The following characteristics were hypothesised for users of the application:

- Home buyers or renters;
- Frequently use public transport;
- Possibly in lower income brackets, as the cost savings associated with public transport are more appealing;
- Likely to be employed rather than unemployed, as employment provides income to acquire a property and public transport can reduce commute times to work;
- Younger demographics were hypothesised to walk further to public transport than older;
- Less likely to be reliant on a motor vehicle for transport; and,
- Assist for ecological and environmental friendly lifestyle.

Based on this assessment, the following user needs were hypothesised:

- Determine the nearest train station to a property of interest, including travel time by walking;
- Access information about the nearest train station, such as the number of train lines;
- Access to the application through an existing real estate website;
- Process requests quickly;
- Intuitive to navigate around the application.

To service these needs, it was determined the application shall have the following functionality:

- Function as a web GIS application in a web browser, as the majority of users are not expected to have advanced computer skills;
- Be available as an extension to an existing real estate web page, to be convenient and accessible;
- Be intuitive to use, therefore consist of basic functions such as:
    - Zoom in/out
    - Pan
    - Layer switcher
- Consist of clear formatting.

Data requirements were then determined by first selecting a city that was representative of our problem statement, and then focusing on a region of that city to base the application on. It was assessed this was an appropriate method given that access to public transport has relevance globally, and given the wide availability of data, an application could be developed anywhere in the world.

Brisbane, Australia was selected because it has experienced, and is still undergoing, expansion in the housing market due to an increasing population as well as construction of new train networks. It therefore provides an appropriate case study to demonstrate our problem statement as there is a strong demand for housing, and many train stations are within walking distance of the properties.

Two suburbs in Brisbane were selected - Wynnum and Manly, as indicated in Figure xx, which contains 4 train stations. A small sample was selected to create a prototype, which can then be expanded in scope or region. The following data requirements were established:

1. Wynnum and Manly suburbs train stations (Wynnum, Wynnum Central, Wynnum North and Manly);
2. Railway lines;
3. Real estate data for properties for sale;
4. Cadastral data; and,
5. Roads data.

3.1.2 System

'The first step in completing a system architecture design is to select appropriate project workflows to represent your business requirements. Your selected project workflows identify the processing loads that must be supported by your selected hardware solution to satisfy your business needs' (Peters, 2009).

As this application is only a prototype running on a local system, system requirements were not a consideration. However, if the application were to be implemented on the internet, the following factors would need to be considered:

- The ability to change location - since our web application is a prototype that uses data from Brisbane suburbs, it is important to ensure data is acquired in the same format when applying it in different locations;
- Ensure the server hosting the application is sufficient for the volume and location of visitors - Alexa (www.alexa.com), a service provided by Amazon with 7-day free trial version which provides information on website traffic and information about unique visitors, could be used to determine web traffic to real estate web sites where the application will be hosted which could be used as a baseline for our application. If the application is hosted at servers such as Amazon Web Services or Heroku, the server will adapt automatically;
- Ensure the real estate webpage where the application is hosted receives sufficient web traffic - the Alexa service described above could be used to determine this information; and,
- The area of interest is in proportion to demand - ensure the application is aligned with the demands of the users of real estate webpages.

3.2 Design

During the design phase, hardware and software procurement requirements were identified, as well as network requirements, and data was acquired. The application was developed using Agile Methodology, with a series of version that introduced successive improvements in functionality. Figure 24 illustrates the software considered to design the application. All software is open source, primarily due to budget constraints, however open source software is also suitable for the scope of this project.

PostgreSQL was selected as the spatial database for all versions, as was GeoServer as the GIS server, with SLD (Styled Layer Descriptor) XML based mark-up language for data styling. To display data in a web page, HTML language containing JavaScript was used. The base map is from OpenStreetMap for version I and II, and Leaflet for version III. A web server was not selected because the application prototype is only intended for a local host, however should the application be accessible on the internet, a web server would be utilised.

Figure 23: System design for TSPF application

Hardware and network requirements were deemed outside the scope for the project and therefore not considered. However, if implementing on the internet, the following factors would need to be considered:

- Identifying where the GIS users are located in relation to the associated data resources (site locations);
- What network communications are available to connect user sites with the GIS data sources; and,
- What are the peak user workflow requirements for each user location (Peters, 2009)?

Data was sourced from different resources, train stations, railways - OpenStreetMap, properties for sale www.domain.com.au, cadastral parcels - Brisbane Council.

An Entity Relation (ER) diagram was developed to provide a conceptual representation of the database prior to implementation, as demonstrated at Figure 25. This is the primary diagram which was changed during the project.

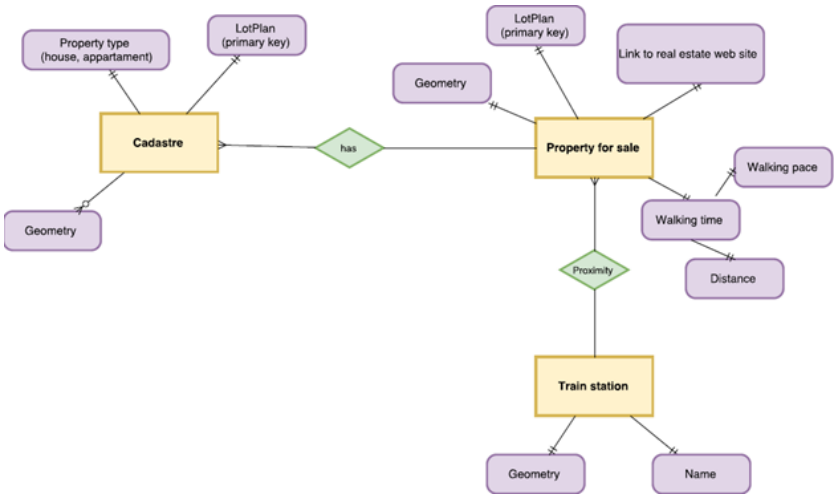


Figure 24: ER diagram for TSPF application

3.3 Project Management

The majority of the group work for this project was conducted remotely, which required an online platform for the group members to conduct online meetings, share information, create the application and draft the report. For this to occur successfully it was important the online platform enabled rapid and easy sharing of data and information, was intuitive/easy to use to allow all members to learn it in a short period of time, and was free to use. The following applications were selected:

1. Skype,
2. OneNote and SharePoint, and,
3. Google Docs.

3.3.1 Skype

Skype is an application that provides internet voice, video, and video conference call services. It was selected to conduct online group meetings because of the following key characteristics:

1. Simultaneous group conference calls,
2. Reliable call quality, and,
3. Free to use.

3.3.2 OneNote and SharePoint

Microsoft OneNote is a software program for the storage and organisation of information. It is complemented by the cloud hosting program SharePoint, which enables OneNote notebooks to be shared by multiple users. These applications were selected for group members to collaborate online.

'OneNote is an exceptionally intuitive application to master' (Barber et al, 2009). It provides simple and clean formatting options such as the ability for the user to type text anywhere on the page and the ability to drag and drop information, and the useful structure with the notebook divided into groups, sections and pages.

When OneNote is hosted on SharePoint, a copy of the entire notebook is stored in the cloud which acts as a central copy, and a local copy is stored on each user's computer. Any changes are first performed on the local copy, and then synchronized with the server whenever a network connection is established (Barber et al, 2009).

One of the greatest strengths and greatest weaknesses of OneNote is the lack of sign-in/sign-out functionality, which improves usability by making it faster and simpler to use, however introduces the risk of multiple users undertaking simultaneous edits (Barber et al, 2009). During the course of this project this was not an issue because it was deconflicted with regular communication amongst the group.

3.3.3 Google Docs

Google Docs was selected to draft the project report. It is application that allows multiple authors to access a document stored on a Google server through a web browser. Changes to a document are communicated to the server at approximately 30 second intervals, and an extensive revision history is maintained. It enables the export of documents in a variety of formats, such as PDF, HTML and Word, making it a useful collaboration platform for a variety of disciplines (Dekeyser & Watson, 2007).

Google Docs has several advantages for document creation:

1. Lightweight - only requires a web browser;
2. Free - requires a Google account; and,
3. Supports simultaneous edits well, with a full revision history and ability to comment.

The following disadvantages were identified:

1. Does not support creation of Figures, bibliography or citations; and,
2. Output layout can change significantly and be difficult to control when exporting to a different format.

3.3.4 Group work

The project was divided into 2 parts - the development of the application and writing the report. The unique differences of these 2 parts required different approaches to project management, therefore Agile methodology was used to create the application in a series of iterations, and the report was written using Critical Chain Management. The principles of Waterfall methodology were used to create a timeline to provide an overview of deliverables and assign responsibilities as demonstrated in Figure 26.

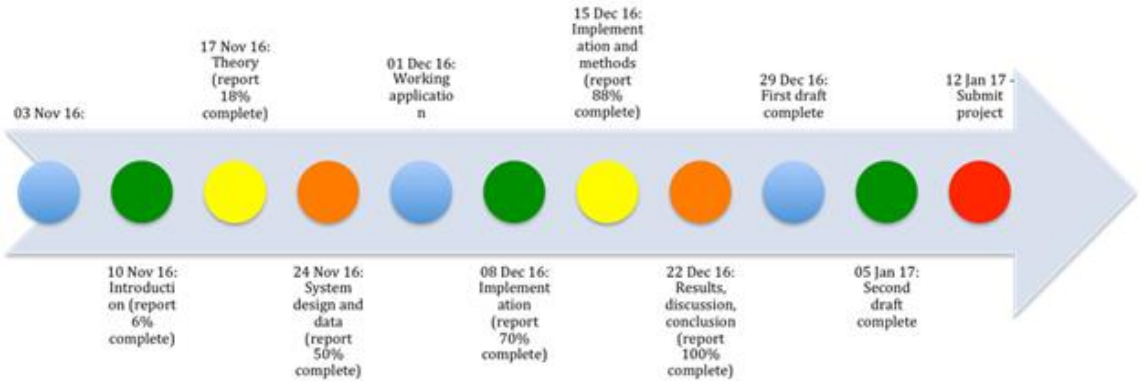


Figure 25: Timeline of the project

A different group member was assigned responsibility of project manager each week, which is indicated by the different colours in the timeline. The project manager was responsible for overseeing all group work, communications with the project supervisor, and planning further project directions.

As the majority of our group work for writing the report was undertaken remotely by group members in different parts of the world it was often the case that members worked independently, which presented challenges to coordinate content and ensure the report flowed logically. To accomplish this, Critical Chain Management was used to divide the report into sections, and then assign components of each section to a group member. Upon completion, the section was reviewed by each group member and amended accordingly. This process was not protocolised anywhere, only verbally during group discussions which occurred weekly in accordance with the timeline at Figure 26.

Agile methodology was used to develop the application, with each group member taking it in turns to develop an interaction based on group discussions about the desired functionality of each version. The responsible group member would work independently on developing the version, seeking assistance from the other group members when required. The more complex version was assigned to the more capable group members.

To ensure a successful end result a good communication system must be in place. For this we used OneNote, Google Doc, Skype and Facebook. OneNote was used for documenting our thoughts, discussions, solutions and future questions (Figure 27).

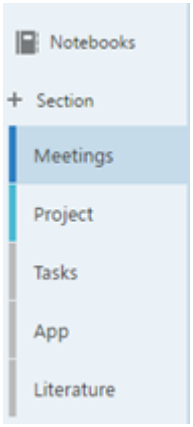


Figure 26: OneNote for project management

The different tabs that where created in one note were: Meetings, Project, Tasks, App, Literature. All chapters have subtabs. The Meetings tab contain information about group meetings, agenda, discussions, decisions, solutions, possible offers and so on. Project - more specific information about project, like software, content, ER diagrams and etc. Task helps to specify and see each member's tasks. The App section covers the different techniques and software used in constructing the app, requirements, design, version, and everything else related to the project is also included. Literature – any literature relevant to the project is shared here; journals, textbooks, and web pages.

4 Implementation

The Train station proximity finder (TSPF) application was developed using Agile methodology to develop a series of web application versions with increasing functionality. The goals of the versions are as follows:

1. Version I: create a database, load the database on GeoServer and present the data within a web browser;

2. Version II: create and display buffer zones representing areas which can be traversed in 5min, 10 min, 15min or 20 min from train stations;

3. Version III: Replacing certain software and increasing current capabilities. Leaflet replaces open layers and Euclidean distance buffers are replaced with isochrones which illustrate travel times from train stations to surrounding properties in 5, 10, 15 and 20 minutes.

4. Version IV: Open Source Routing Machine (OSRM) is implemented to provide routing capabilities for the application, and the Django web framework is introduced for the creation of the final web application

This chapter incorporates the construction and implementation phases of GIS SAD (Figure 22 on page 36).

4.1 Version I

The first Version starts with data collection. Property for sale, railway lines, train stations. Property for sale was taken from Australian real estate agency www.domain.com.au. as csv file. Railway lines and train station were downloaded from OpenStreetMap. Spatial information was acquired by concatenating the addresses and using Excel Geocoding tool and Bing's API key. The database from real estate and database from Bing map were joined within the address column and this is how we get spatial information about properties for sale. The data from OSM was converted to shapefiles with Global Mapper 18. In the OSM we chose bounding box and then download the data (Figure 28).

Figure 27: Exporting data from OpenStreetMap

Due to the large volume of data for the entirety of Brisbane we created a demo version which consists of two suburbs Wynnum and Manly. And 4 train stations: Wynnum, Wynnum Central, Wynnum West, and Manly, each of them representing different living areas. The first step was to select data. For this we used ArcMap 10.4 as the licence was still available at the time. Red pins represent properties for sale, train station and railways were represented by self-explanatory default symbols provided by ArcMap. Data styling and visualization is shown in Figure 29 overleaf.

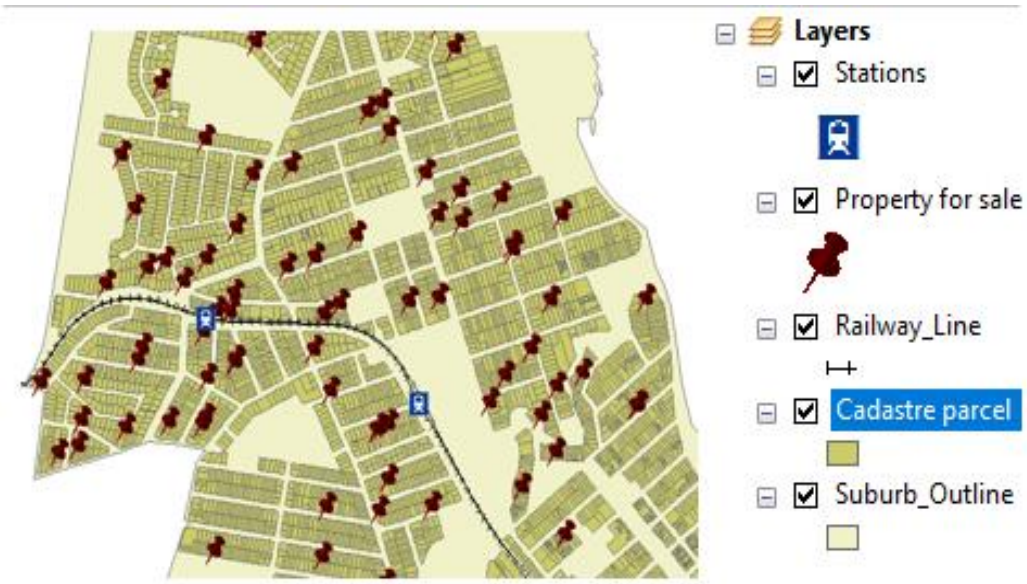


Figure 28: Data visualization in ArcGIS

The data was loaded into PostgeSQL using PostGIS 2.0 and DBF loader. The SRID (spatial reference system identifier) used is 28356, due to Brisbane belonging to zone UTM 56 (Adventurer, 2016). Using GeoServer and OpenLayers the data is presented in our browser. The Red dots are properties for sale, railway line is on its cartographical line and train stations are marked as green triangles (Figure 32). The Base map is from OpenStreetMap. There are scale and zooming functions, which are necessary for any kind of map. The layers: train station, railway line, and for sale, was displayed from geoserver, this part of the code is shown in Figure 30. The title "train station" is what the user will see on the screen. "Visible: true" - indicates that this layer will appear automatically when the map is loaded. The option "source" shows the source for the layer which is geoserver.

```
new ol.layer.Tile({
  title: 'train station',
  visible: true,
  source: new ol.source.TileWMS({
    url: 'http://localhost:8080/geoserver/wms',
    params: {'LAYERS': 'ProjectAustralia:train_station'},
    serverType: 'geoserver'
  })
}),
```

Figure 29: JavaScript code

The styling is created in geoserver with xml, an example of train station styling code is shown in Figure 31. The opacity was chosen 0.2, because we want to see the map behind the sighn, and the colour code is HEX #009900, which is green. The stroke is full black (colour code #000000) and the stroke width is 2. With these parameters, the train station is easy to notice, but the map is not overburdened.

```
<Mark>
  <WellKnownName>triangle</WellKnownName>
  <Fill>
    <CssParameter name="fill">#009900</CssParameter>
    <CssParameter name="fill-opacity">0.2</CssParameter>
  </Fill>
  <Stroke>
    <CssParameter name="stroke">#000000</CssParameter>
    <CssParameter name="stroke-width">2</CssParameter>
  </Stroke>
</Mark>
```

Figure 30: Styling code

The final result is shown on Figure 32. It displays train stations, railway lines and properties for sale where the base map is from OpenStreetMap.

Figure 31: Data in web map

4.2 Version II

For the second version, we made a website with HTML. Here we have the main menu bar, which allows to choose between 4 areas Wynnum North, Wynnum, Wynnum Central and Manly. The code from the version two was upgraded with tables and links to other maps. The one part of the code is given on Figure xx. The align is centre the table width is 183px and the file is taken from computer, which directly path is written. The text which user will see is "Wynnum North" because the link provides you to this train station.

```
    <tr align="center">
      <td style="width: 183px;"><a
 href="file:///C:/Users/Nijole/Desktop/Project/nvu%20map/OpenLayers/Wynnum%20North.html">Wynnum
North</a></td>
    </tr>
```

Figure 32: HTML code

The map displays the whole of Brisbane and the property buyer can zoom to the area in which a customer is interested in, but for faster navigation, a table with all relevant areas has been included. The view of what the map looks like when first opened is shown in Figure 34.

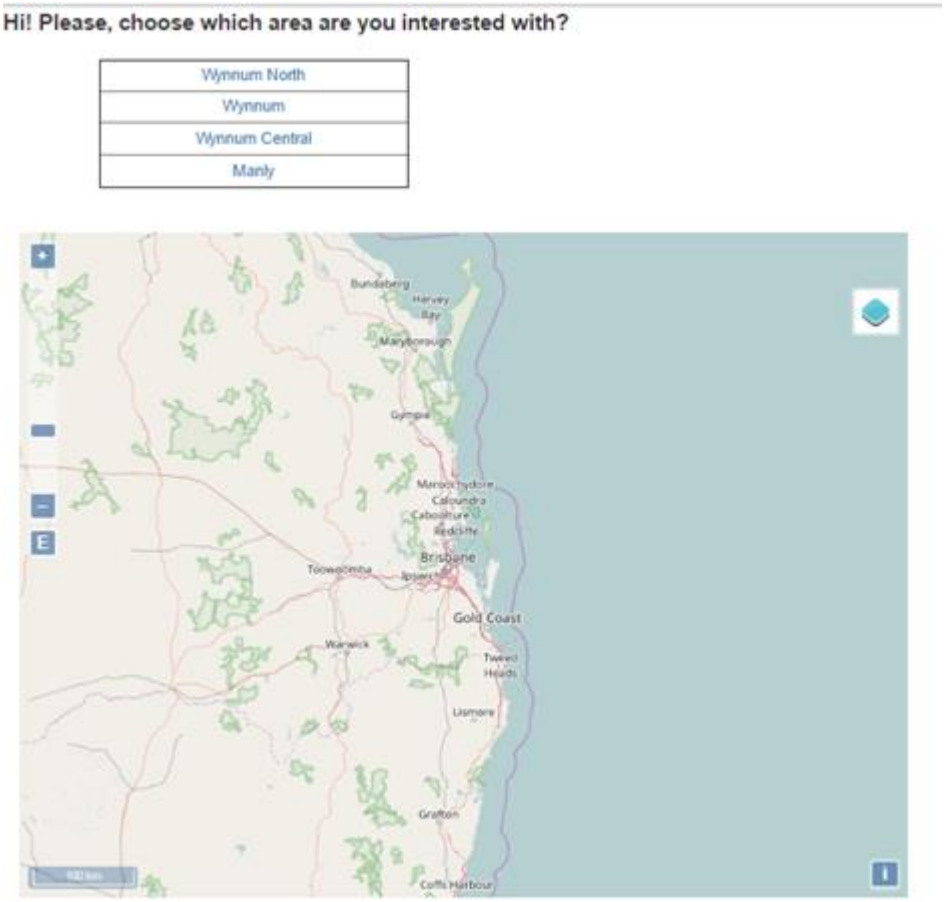


Figure 33: First view of web page

Once the user has selected Wynnum North from the menu, the new map is displayed (figure 35)

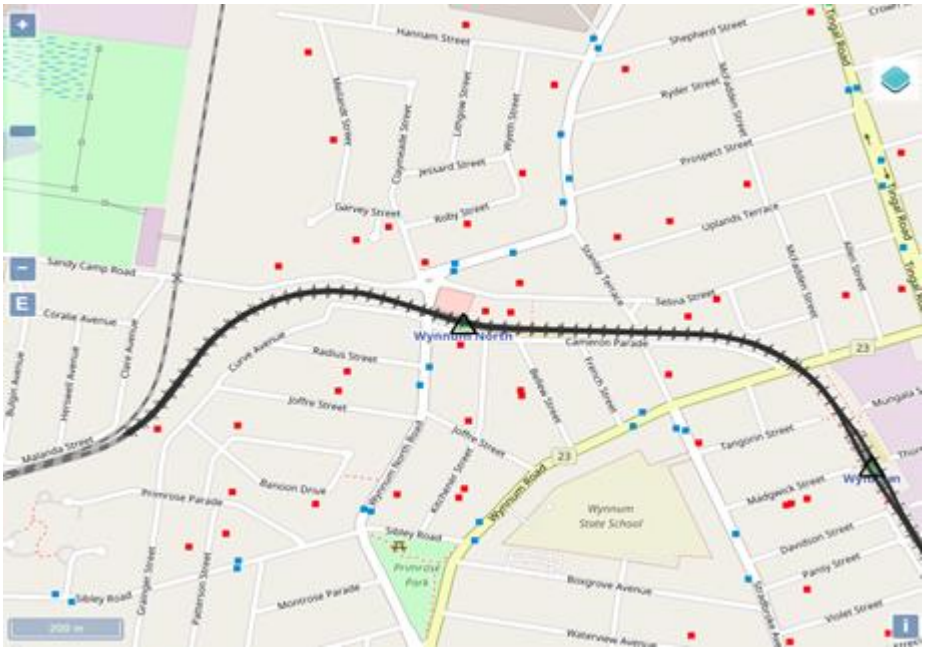


Figure 34: Wynnum North map

On the menu bar on the right it is possible to choose walking distance from 5, 10, 15, 20 minutes. After you've decided which distance you are able to walk, the application provides a new view with buffer zones (Figure 36). It is possible to choose more than one buffer zone.

Figure 35: Buffer zone showing 5 minute walking distance

The problem here is that buffer zone does not indicate walking path but simply Euclidian distance. So, the real walking distance will not be 5 min. Even if the property is close, the walking distance might be a lot longer. Version 3 will address this problem.

4.3 Version III

The third version of the project further enhances the existing features of the application and introduces new features and functionality to the application. The major changes implemented into the third version of the project are:

1. The switch from OpenLayers to Leaflet in order to prepare the application for the routing process which will be added in the fourth and final version
2. The introduction of isochrones in order to calculate road travel distance as opposed to circular buffer zones
3. The addition of the cadastral dataset, which will be used to display properties reachable within 5, 10, 15 or 20 minutes of walking time

4.3.1 Data Collection and Preparation

The third version builds on datasets previously introduced in version one and two. The datasets utilized in this version include:

Dataset	**Provider**	**Usage**
Streets	Openstreetmap (Mapzen)	Creation of Isochrones
Digital Cadastral Database (DCDB)	Queensland Government Spatial Catalogue	Creation and Display of Isochrones
Railway Stations	Queensland Government Spatial Catalogue	Display of Railway Stations
Suburb Boundaries	Australian Bureau of Statistics	Data filtering and preparation

Table 1: Datasets, providers and usage

The application will continue its localized focus on the suburbs of Wynnum and Manly, and the four train stations of Wynnum North, Wynnum, Wynnum Central and Manly.

Firstly, the suburb boundaries dataset is loaded into QGIS (replacing Arcmap due to lack of licencing) and the data is filtered to display only the suburbs relevant to our application. (Figure 37) The suburbs included are Wynnum, Manly and the adjoining suburbs of Manly West, Wynnum West, and Lota.

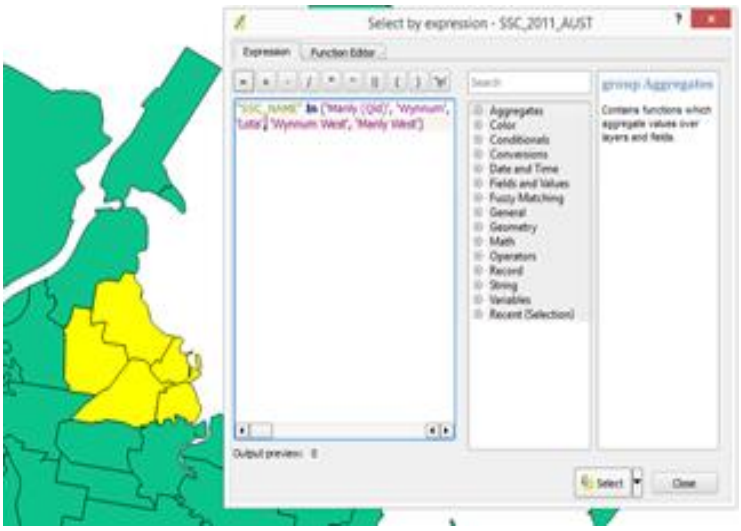


Figure 36: Filtering by Expression

The selection is then saved as a new layer and the Digital Cadastral Database layer is added to the QGIS Project. The DCDB layer is then filtered to include only the data relevant to our project. This is done using the QGIS geoprocessing tool 'Clip'. (Figure 38)

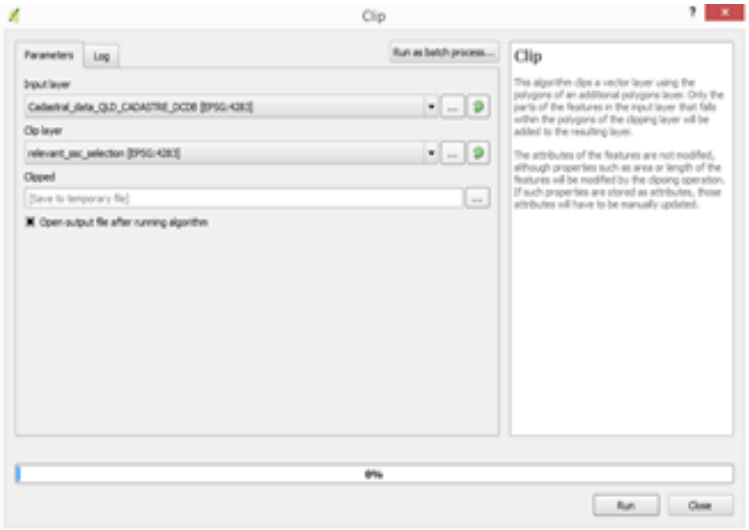


Figure 37: Clip Geoprocessing Tool

The resulting layer includes all the property boundaries that will be relevant to our project. This filtering is done to speed up processing times and to avoid having to work with the entire DCDB layer for the whole state of Queensland.

The railway stations layer is then loaded into QGIS and filtered in a similar manner to the suburbs dataset. This results in a separate layer which includes the four train stations required for our project as points. The four train stations are Manly, Wynnum, Wynnum Central, and Wynnum North.

The resulting map, shown in Figure 39, includes the DCDB and the railway station relevant to our project in two separate layers. These layers will be used at a later step in the process to create the isochrone maps.

Figure 38: Filtered DCDB and Railway Stations Map

Isochrone Maps

An isochrone map is a time-based travel map used to show areas of equal travel time to a predefined point. For the purposes of our application, the isochrone maps will show cadastral property boundaries in relation to the walking distance to each train station.

The isochrone maps are created in GRASS GIS, using the v.isochrones extension. The extension creates a vector polygon map of isochrones based on OpenStreetMap data. In order to create the isochrone maps, the OpenStreetMap data has been downloaded as a shapefile from the third-party provider MapZen. The data has been extracted only for the Brisbane Metro area. (Figure 40)

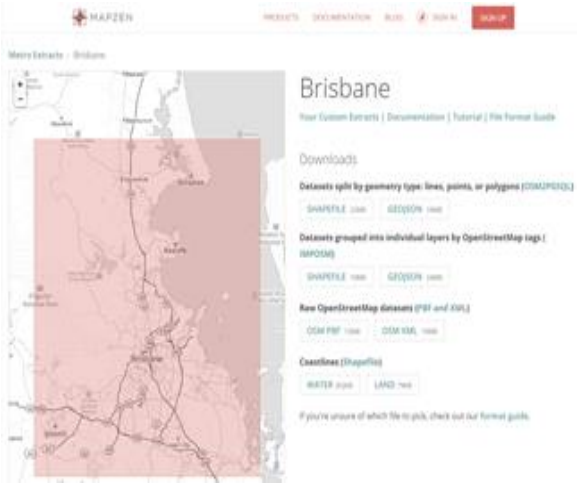


Figure 39: MapZen OpenStreetMap download options

The only dataset relevant to our purpose at this stage is the roads shapefile. This shapefile is loaded into QGIS for further manipulation.

Our final project version will include a routing option which will provide walking paths between the train station and each listing. While this is not yet relevant in this version, for consistency, the walking speeds and the paths used by OSRM are now taken into account when creating the isochrones. The OSRM project generates routes based on OpenStreetMap data and as such provides a walking profile, shown in Figure 41, which lists the walking speeds and paths utilized when calculating a route. By filtering the OpenStreetMap data using this profile, we can ensure that the isochrone creation algorithm utilizes a walking speed that is consistent with the routing algorithm, and will not consider unwalkable paths, such as a highway.

```
walking_speed = 5

speeds = {
  ["primary"] = walking_speed,
  ["primary_link"] = walking_speed,
  ["secondary"] = walking_speed,
  ["secondary_link"] = walking_speed,
  ["tertiary"] = walking_speed,
  ["tertiary_link"] = walking_speed,
  ["unclassified"] = walking_speed,
  ["residential"] = walking_speed,
  ["road"] = walking_speed,
  ["living_street"] = walking_speed,
  ["service"] = walking_speed,
  ["track"] = walking_speed,
  ["path"] = walking_speed,
  ["steps"] = walking_speed,
  ["pedestrian"] = walking_speed,
  ["footway"] = walking_speed,
  ["pier"] = walking_speed,
  ["default"] = walking_speed
}
```

Figure 40: OSRM Walking Profile

Once the OpenStreetMap dataset has been loaded in QGIS, it can now be manipulated to ensure consistency with the walking profile. This is done by first using a SQL query to select all the relevant road types utilized by the walking profile (Figure 42) and then adding a new column titled ‘speed’ which will include the walking speed for each road in our new dataset (Figure 43).

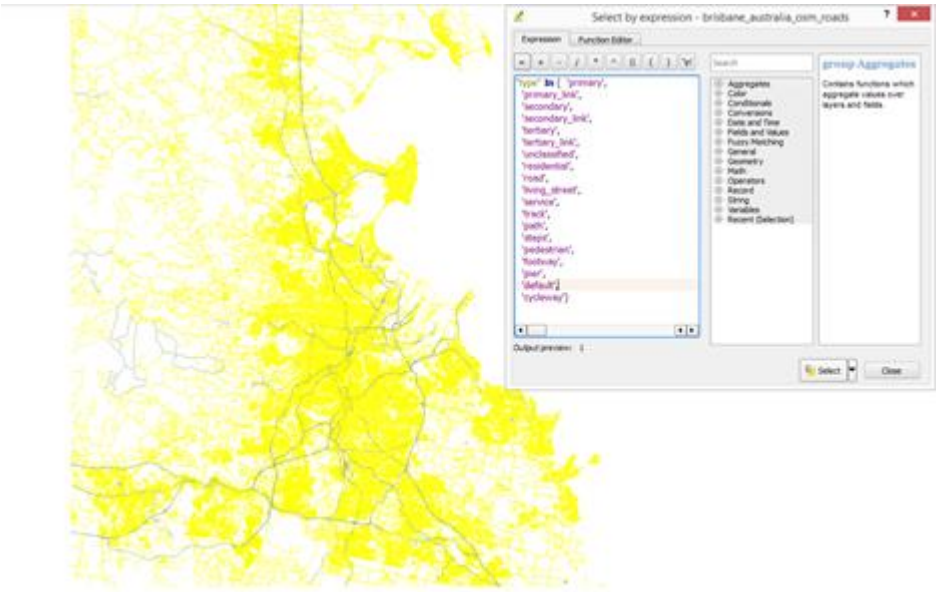


Figure 41: Selecting Roads Based on Walking Profile

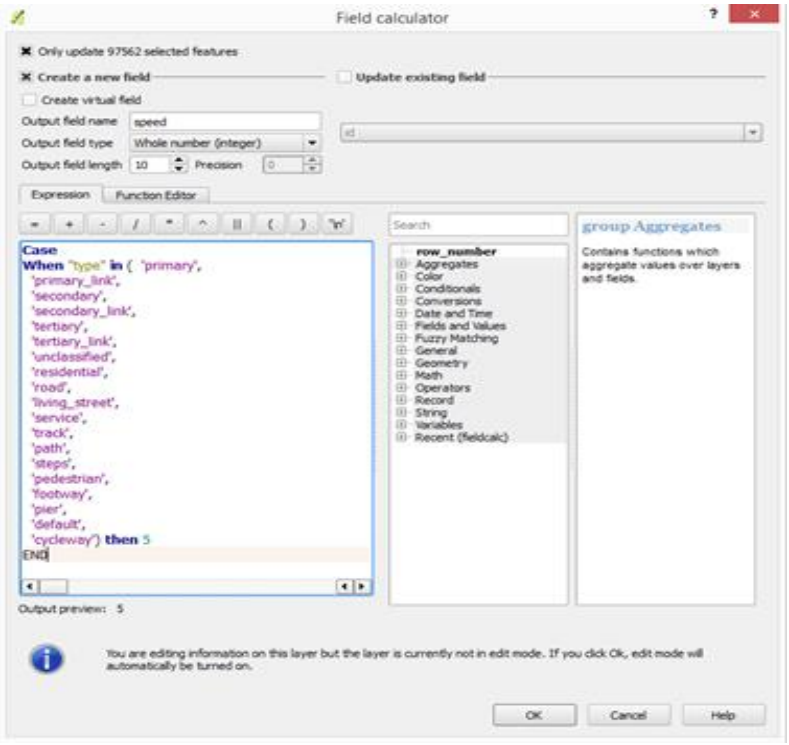


Figure 42: Creating a new ‘speed’ field

The layer is then saved using Australian Albers CRS (EPSG: 3577). This is to ensure distance is not distorted in the projection, as distance is the most important variable when calculating travel time. GRASS GIS is used to create the isochrones maps. After starting a new GRASS session, the vector data which includes the street network, and the starting nodes (train stations) is then imported (Figure 44).

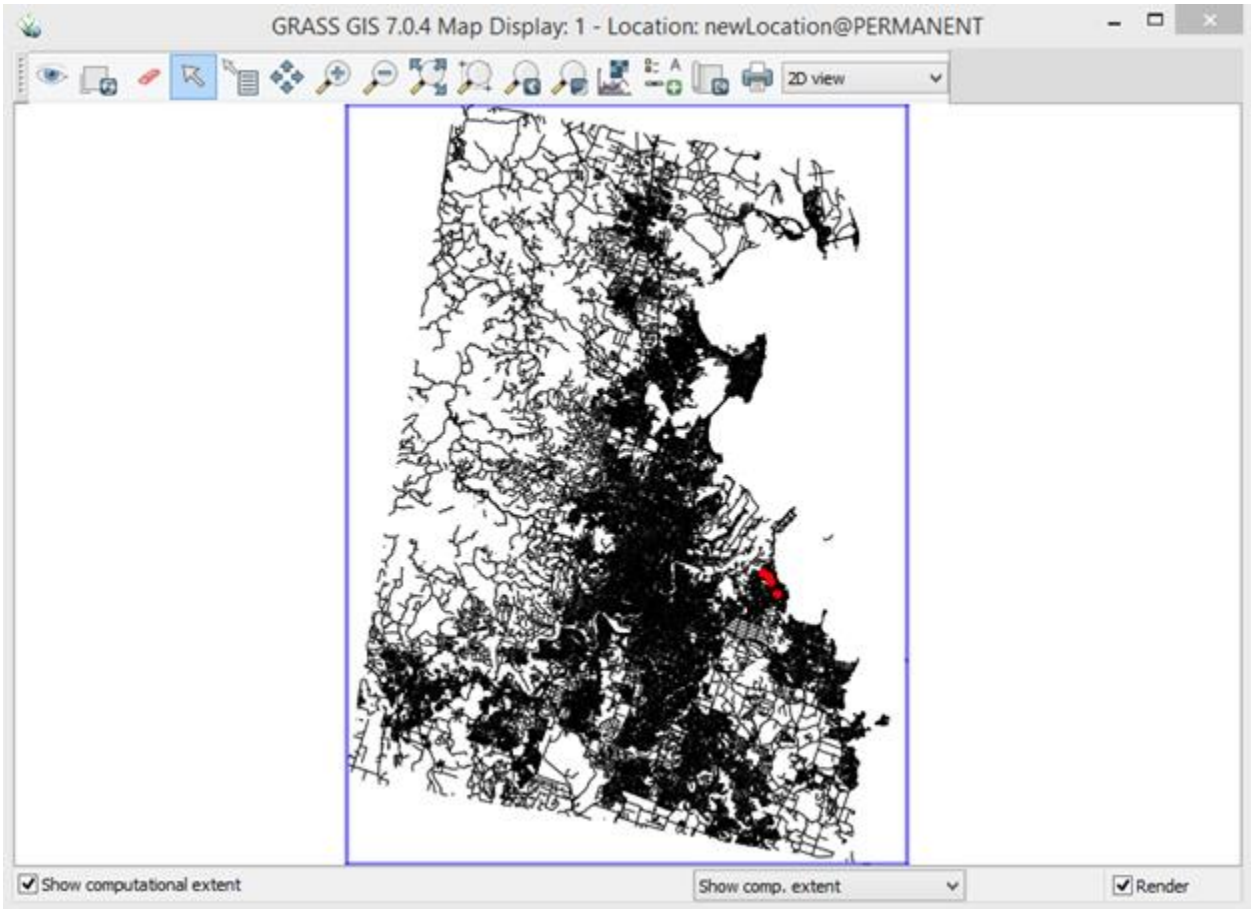


Figure 43: Data displayed in GRASS

In order to ensure data accuracy, the region, bounds and the display settings must be checked and defined for the data set as shown in Figure 45. Once these are correct, the v.isochrones extension is run.

```
(Wed Jan 04 18:03:49 2017)
g.region -p
projection: 99 (GDA94 / Australian Albers)
zone:       0
datum:      gda94
ellipsoid:  grs80
north:      1
south:      -3181087
west:       0
east:       2073922
nsres:      99.99962277
ewres:      100.0010608
rows:       31811
cols:       20739
cells:      659728329
(Wed Jan 04 18:03:49 2017) Command finished (0 sec)
```

Figure 44: GRASS Region Settings

The v.isochrones module is run using the settings shown in Figure 46. The time steps of 5, 10, 15, and 20 minutes are chosen for the walking time travel steps. The method used for the isochrone calculation is r.cost. This method ensure that a proper calculation is performed regardless of the topological accuracy of the roads layer. A raster layer can also be created with this method, however for this project we are only interested in the vector output. The speed for off-road areas in the r.cost tab is set close to 0 as off-road areas are considered to be blocked by buildings and are as such inaccessible to pedestrians.

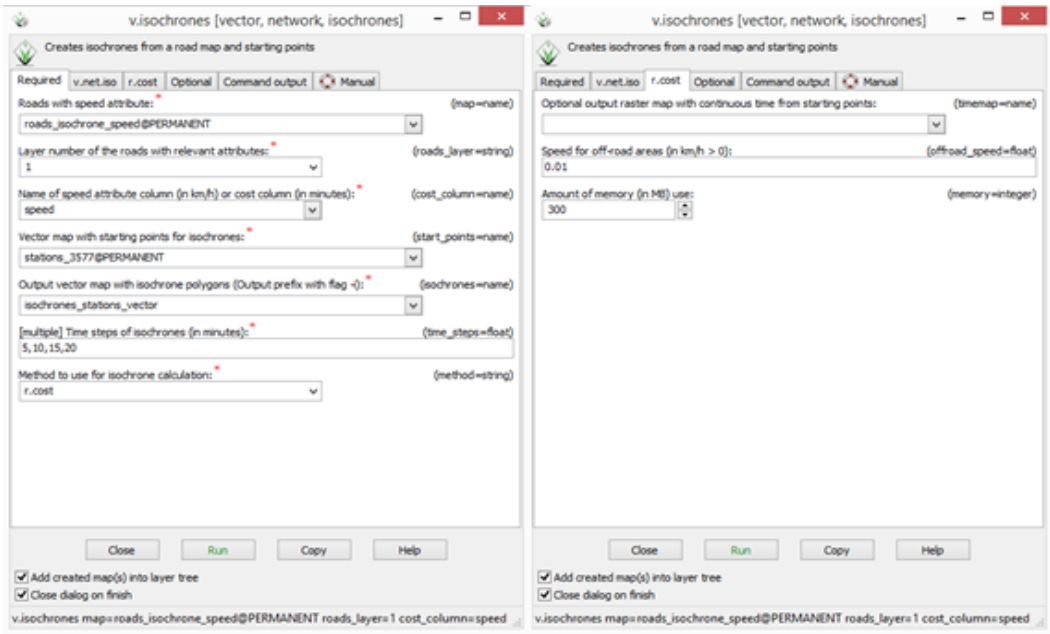


Figure 45: v.isochrones settings

The module computes an isochrone map based on the input parameters and produces a vector layer. This vector layer is then exported as a shapefile and imported in QGIS for further analysis (Figure 47)

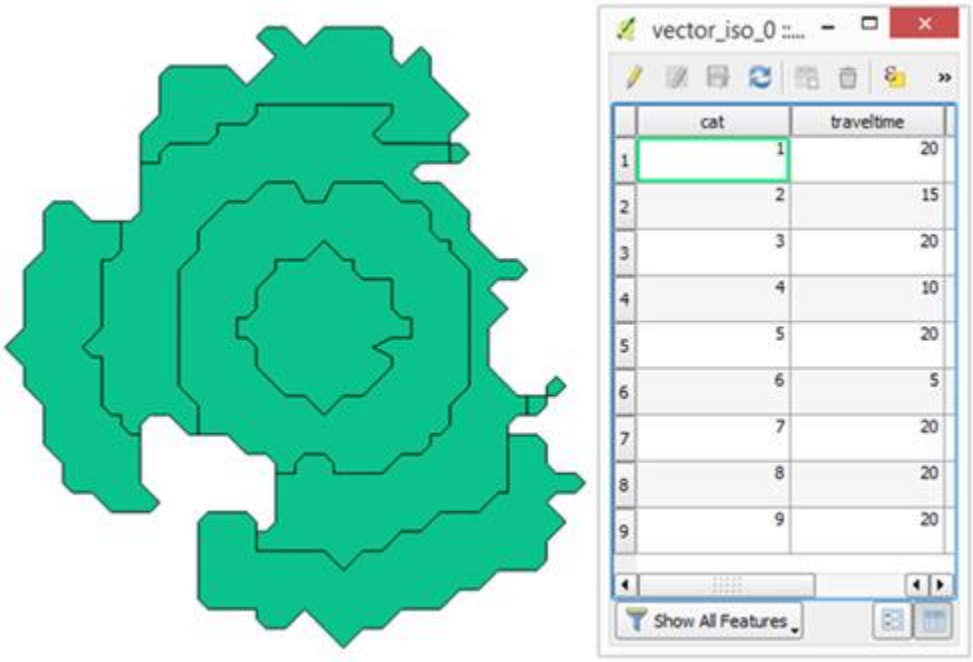


Figure 46: Isochrone imported in QGIS

The filtered DCDB layer saved earlier in the process is then imported into the map and again filtered using the select by location tool so that only property parcels that intersect the isochrone are saved. The resulting layer is saved as a shapefile and then imported into the Postgres Database using the PostGIS Shapefile Import/Export Manager tool (Figure 48).

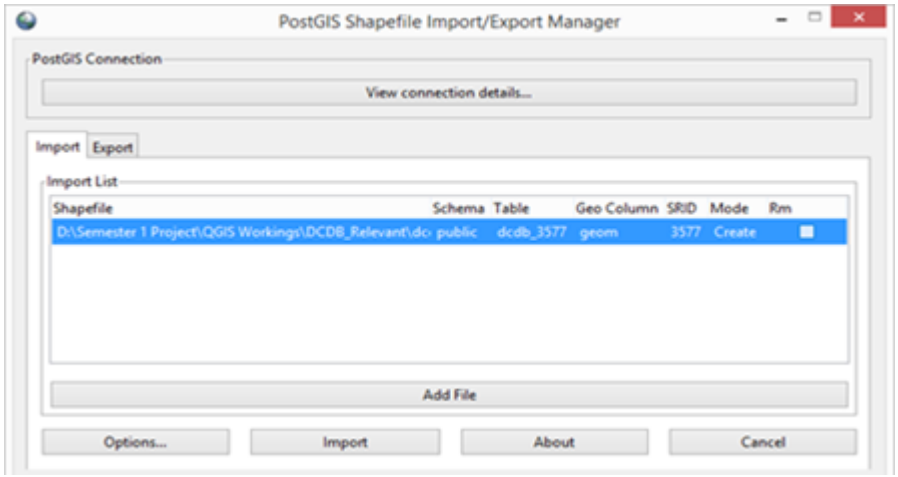


Figure 47: PostGIS Shapefile Import/Export Manager

The resulting cadastral layer is going to be served from Geoserver as a WMS layer to be displayed onto the map. As such, the layer should be saved in a database which has an established connection to the

Geoserver Workspace. The layer (now saved as a Postgres table) is then be published using Geoserver's admin tools and styling is created (Figure 49). The styling is based on the travel time, and each property boundary will be styled based on the area of the isochrone it intersected with during the above analysis process.

```
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/
xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/sld
StyledLayerDescriptor.xsd">
<NamedLayer>
<Name>Attribute-based polygon</Name>
<UserStyle>
<Title>SLD Cook Book: Attribute-based polygon</Title>
 <FeatureTypeStyle>
      <Rule>
        <Name>5minutes</Name>
        <Title>5minutes</Title>
        <ogc:Filter>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>distance</ogc:PropertyName>
            <ogc:Literal>5</ogc:Literal>
          </ogc:PropertyIsEqualTo>
        </ogc:Filter>
        <PolygonSymbolizer>
          <Fill>
            <CssParameter name="fill">#1a9641</CssParameter>
          </Fill>
        </PolygonSymbolizer>
      </Rule>
      <Rule>
```

Figure 48: Example Styling XML

The new isochrone layer can then be displayed in the application as a WMS layer. The screenshot below (Figure 50) Shows the layer displayed as a WMS layer on a leaflet web map with the styling based on the travel time to the station (5, 10, 15, 20 minutes).

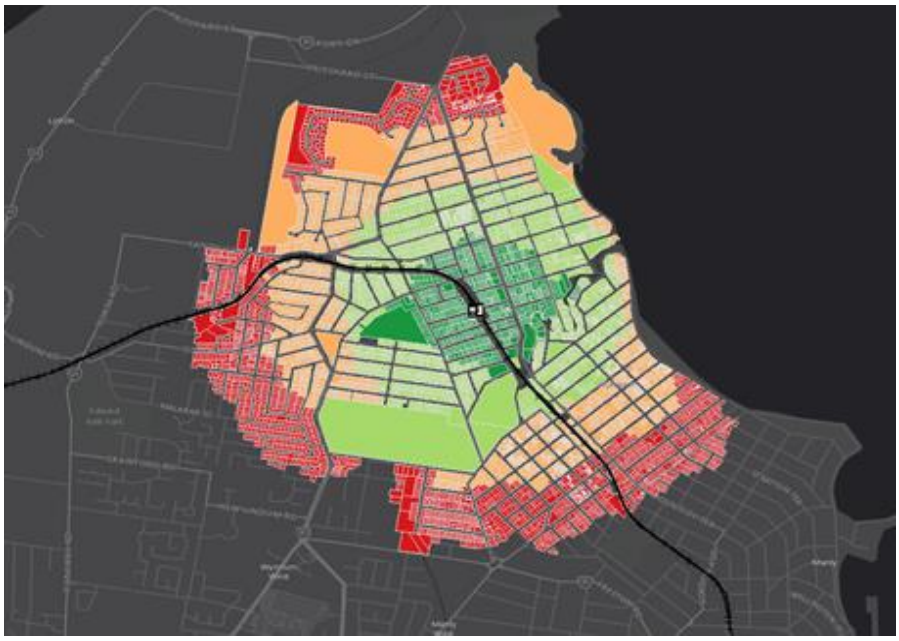


Figure 49: Isochrone as WMS Layer

4.3.2 ER Diagram for version III

The data displayed on the webmap is stored in three tables within a Postgres database, these are: Train Stations, Isochrones, and Listings. When the user searches for properties within a train station, a SQL query is performed on the database. The query returns the train station, its related isochrone, and the listings contained within the isochrone geometry to the map. The train station and the listings are returned as GeoJSON objects which the user can interact with by clicking on them to see more information. The isochrone is returned as a WMS jpg image and is only used to provide the user with a visual representation of how far each listing is from the train station. The ER diagram in Figure 51 below shows a visual representation of the database structure.

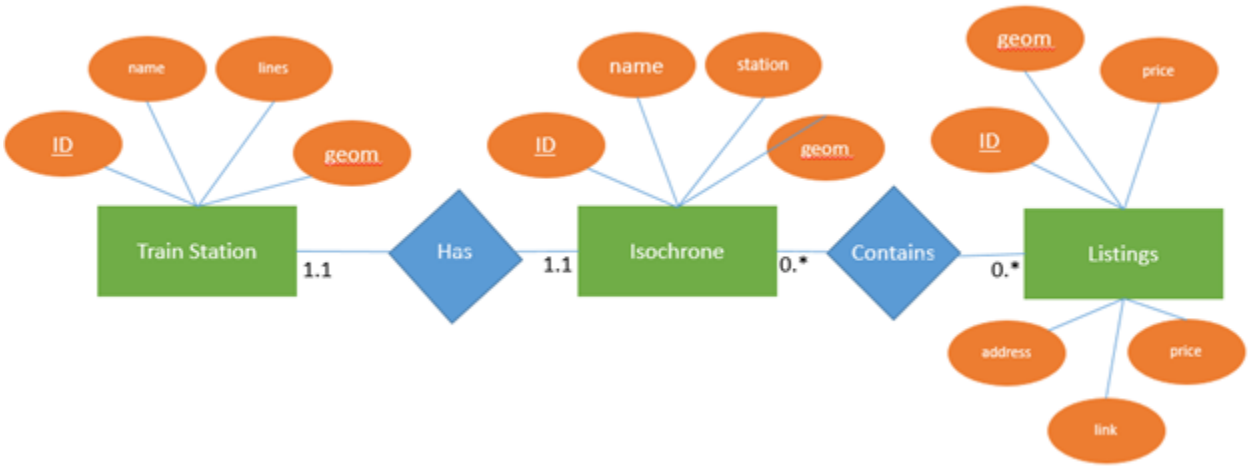


Figure 50: ER Diagram version III

During this version, our application's front end mapping library switched from OpenLayers to LeafletJS. Leaflet is a lightweight web and mobile mapping JavaScript library that allows similar functionality to OpenLayers. Whilst it would have been feasible to keep using OpenLayers for this version, we have decided to make the switch in order to prepare the application for the final version . The fourth and final version of the application will include routing capabilities which will allow the end user to see a walking route from the train station to a property for sale. As we have chosen OSRM to provide the routing algorithm, we were required to switch to Leaflet for an easier implementation on the front end. This easier implementation is provided by the Leaflet Routing Machine plugin for Leaflet.

4.4 Version IV

Version IV of the application further builds upon the data and features utilized in Version III. This version introduces two new major components to the application: Open Source Routing Machine (OSRM) to provide routing capabilities for the application, and a PHP script for connecting the backend database to the frontend Leaflet map.

These components were selected for the following reasons:
1. OSRM was chosen as the desired routing engine due to its accuracy and ease of use 'out of the box' when compared to the alternative, pgRouting.
2. PHP was selected to create the querying parameters between the database and the front end. This was due to the relative simplicity of the script and the wide range of resources available online.

Version IV of the application utilizes the same data and the same database as Version III. However, one major difference is that version IV was established on a separate virtual machine running Ubuntu 16.04., which was done in order to keep the version separate and to provide consistency between the documentation found online for PHP and OSRM with our implementation process.

4.4.1 PHP

In order to connect and display our data to the map, a simple PHP script was created. In addition to PHP, the application makes use of the jQuery and jQuery-ui libraries to facilitate AJAX calls and to provide an autocomplete feature for the search forms.

Figure 52 shows the database connection established in our getdata.php script. These lines of code assign the database credentials as variables, establish a connection to the database using the credentials variables and may return an error if the credentials provided are incorrect.

```php
<?php
ini_set('display_errors', 1);

//database login info
$host = 'localhost';
$port = '5432';
$dbname = 'phpex';
$user = 'postgres';
$password = 'postgres';

$conn = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if (!$conn) {
    echo "Not connected : " . pg_error();
    exit;
}
```

Figure 51: Database Connection in PHP Script (inspired by Sack 2015)

In order to display the data from the database onto a leaflet map, the features will be served as GeoJSON objects, and the SRID will be transformed into WGS 84 (4326) so that it is readable by Leaflet. This is done within the PHP script using the PostGIS queries ST_AsGeoJSON, and ST_Transform (figure 53)

```
27    //get the geometry as geojson in WGS84
28    $fieldstr = $fieldstr . "ST_AsGeoJSON(ST_Transform(l.geom,4326))";
```

Figure 52: SQL Queries in PHP Script

The data can then be queried using spatial querying for search such as within a certain distance of a feature or for features that are within a certain polygon. The results are then processed through JavaScript in order to be displayed on the map. This is done using Leaflet's L.geoJson function as shown in figure 54. Attributes such as marker colour, opacity, style, etc can be changed here.

```
88        var mapData = L.geoJson(geojson, {
89            pointToLayer: function (feature, latlng) {
90                var markerStyle = {
91                    fillColor: "#0000FF",
92                    color: "#FFF",
93                    fillOpacity: 0.9,
94                    opacity: 1,
95                    weight: 1,
96                    radius: 8
97                };
```

Figure 53: L.geoJson function and attributes

Another feature of the application is the web form search functionality. This web form allows the user to perform a query on the database and see only the results relevant to their area of interest (figure 55)

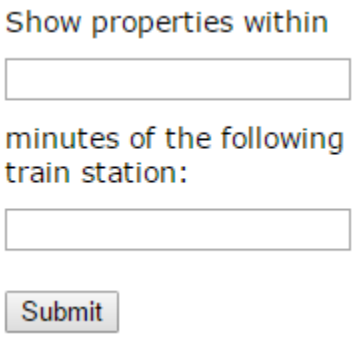


Figure 54: Simple Web Form

The features are then returned on to the map and can be interacted with (clicked on) to provide more information about the property for sale in a popup and to display a route using OSRM.

4.4.2 Routing with OSRM

OSRM provides shortest path routing for road networks, based on data from Open Street Map. As explained in the Theory section of this report, OSRM is based on contraction hierarchies. This allows the routing machine to achieve fast performance and display the route to the client in milliseconds. This is achieved by a long pre-processing stage of the routes on the back end. While this would not be viable for use with data which requires constant updating of traffic information, it is a perfect option for our application.

The OSRM backend has been installed on the same local virtual machine running Ubuntu 16.04. This was required in order to provide access to the walking profile for the routing machine. The OSRM backend was downloaded from the Project OSRM github page using the git clone command (figure 56).

```
git clone https://github.com/Project-OSRM/osrm-backend.git
```

Figure 56: git clone command

The Brisbane OpenStreetMap data was then downloaded in. OSM format from Geofabrik, and the pre-processing of the routes was then completed using instructions provided in the OSRM-backend wiki.

For the purposes of our application, a route is displayed from the train station to the selected property for sale. In order to not overcrowd the display, this route is only visible once the user clicks on the selected property (Figure 57)

Figure 57: OSRM generated route displayed on map

The route is displayed on the map with the Leaflet Routing Machine (OSRM frontend) extension available to the leaflet mapping library. This extension is open-source and provides full functionality for routing and allows full customization to the user interface and interactions.

4.5 Results

The final application utilizes a significant amount of pre-processed and pre-filtered data and makes use of a number of open source technologies. The fourth and final version of the application ties together all the data and technologies in order to provide a user friendly, accurate application where prospective property buyers can search for properties within walking distance of a chosen train station.

The application allows the user to select a time and a train station and query the database through a user friendly form. The results are then returned to the map (figure 58)

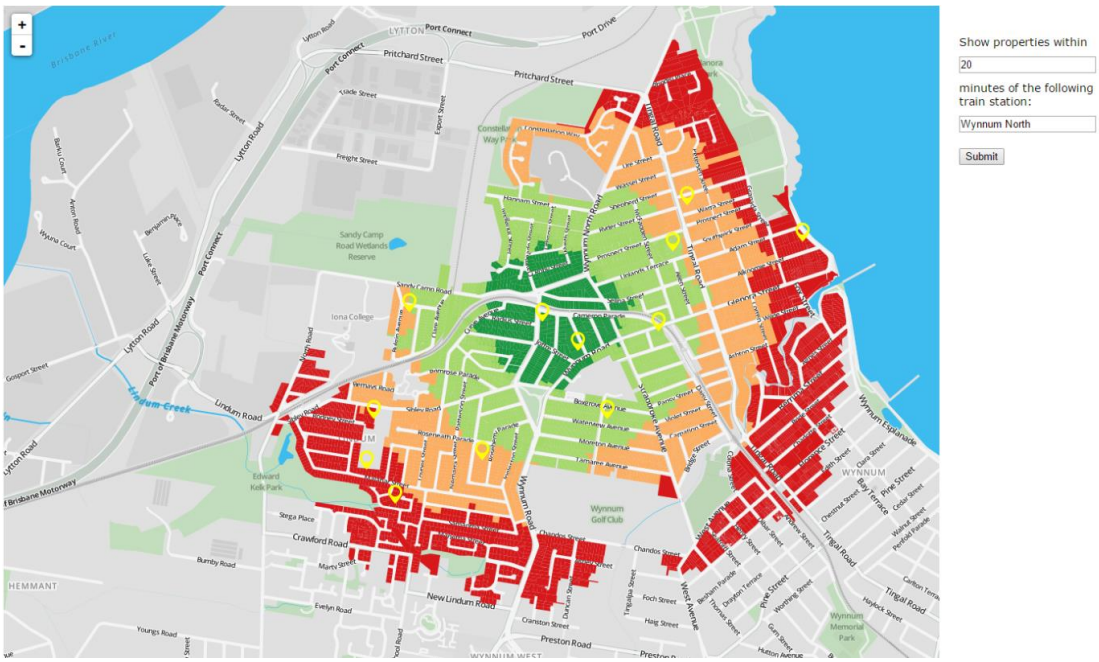


Figure 58: Sales results filtered and displayed on map

The user can then interact with the results by clicking on the marker. OSRM then creates a route between the marker and the train station and displays walking directions and time. A new control is also displayed on the left hand side of the client showing additional information about the property.

The map is also zoomed in for the user to better see the route between the property and the station (figure 59)

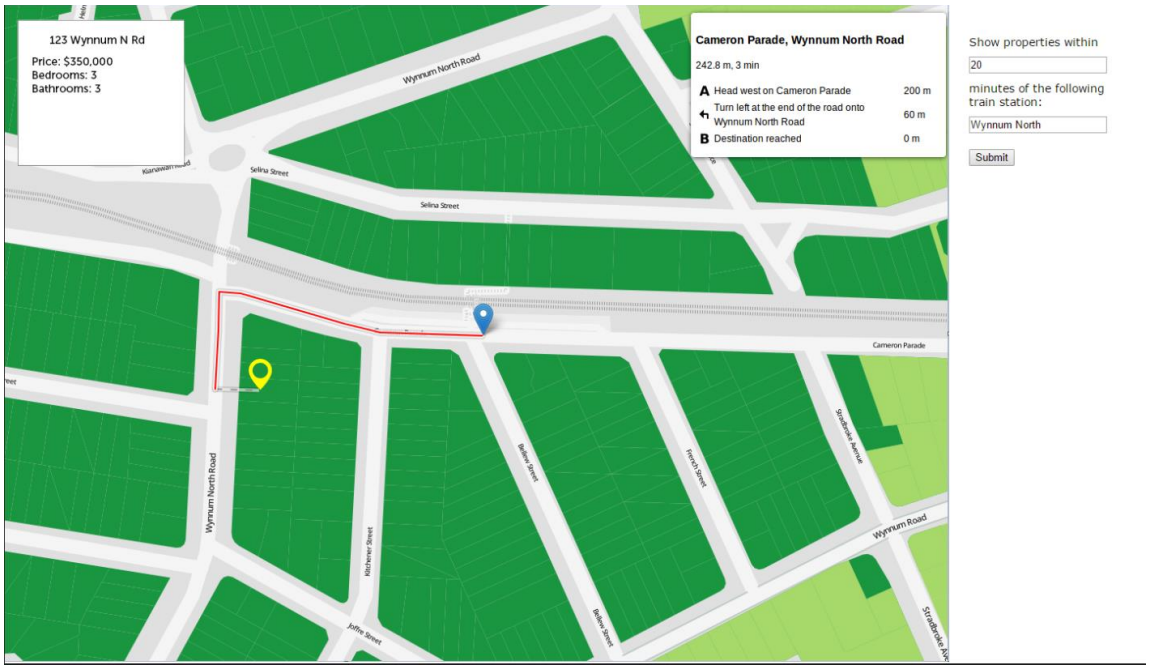


Figure 59: Property information, route, and directions

The final application provides the majority of the functionality that we have set out to implement at the beginning of the project. While the application does have some weaknesses, these can be improved on with time and with additional skill. The application provides a basic proof of concept for the search function and makes use of a large number of GIS and web technologies to provide the desired functionality.

5 Discussion

The aim of this project was to develop a web application that enables interested property buyers to find a property for sale in proximity to a chosen transport hub. In order to achieve, this aim four questions were proposed:
1. What data is required?
2. Which technologies are to be used?
3. Which is the best route-finding method to be implemented in the application?
4. How can we create the application whilst the team is working remotely from different parts of the world?

Discussed below is an overall assessment of the project in the context of the research questions, as well as strengths and weaknesses of the application, and future prospects.

5.1 Data

Research question: What data and data quality could be used in the design and implementation of a web based application to determine the closest railway station to a property for sale.
Data was sourced for the application as illustrated in Table 2.

Data Type	Source	Format	Usage
Sales data	www.domain.com.au	.csv	Display sales data
Street data	OpenStreetMap Mapzen Geofabrik	OSM converted to shapefile, shapefile	Display street info Creation of isochrones
Railway stations	OpenStreetMap, Queensland Government Spatial Catalogue	OSM converted to shapefile, shapefile	Display railway stations Query data relative to train stations
Digital Cadastral Database (DCDB)	Queensland Government Spatial Catalogue	Shapefile	More appealing and map representation Creation and display of isochrones Ability to spatially query data
Suburb Boundaries	Australian Bureau of Statistics	Shapefile	Data filtering and preparation

Table 2: Data for the application acquired from a variety of sources.

The majority of the data was sourced from open government data, from Government departments, and was provided in convenient formats suitable for our application. Sales data however, which was provided by the popular real estate website www.domain.com.au, was provided as properties for sale with addresses but no spatial information (coordinates). The Bing maps geocoding API was therefore used to provide coordinates based on the addresses.

The Queensland government spatial catalogue and Australian Bureau of statistics are both government sources of data and thus should be highly credible. The sales data provided by a commercial website which is a real estate portal where agents upload their sales on a regular basis. The only data that might be worthy of scepticism is OpenStreetMap data, which is public participation GIS, resulting in questionable authenticity. However, the data seems to correlate well with the more reliable data sources. The Data was all free to use and easy to find.

5.2 Technologies

Research question: What technologies could be used to design and implement a spatially enabled property search engine?

The majority of the application was created with open sourced software, except for ESRI's ArcGIS, which was free to use due to a license provided by Aalborg University. Open sourced software was selected primarily due to funding constraints, however it also performed the desired function to the required standard, and in some cases, was superior to the commercial option. With regards to the database, the open source software PostgreSQL was selected as it performs interactions at a much faster rate than the commercial option - Oracle. PostgreSQL, with the PostGIS extension also provides the most querying and integration options when used in conjunction with the Django web framework, and it has become the de facto standard for open source spatial databases.

The GIS server selected was GeoServer due to its adaptability to easily coordinate with OSRM and Leaflet; it also has a very user friendly interface. GeoServer is used in our application to display the pre-processed isochrones generated using Grass GIS for each train station.

The web mapping JavaScript library selected for the final version (version IV) was Leaflet. OpenLayers was selected initially and was used for Versions I and II; however, enabling a routing capability within OpenLayers proved difficult. Leaflet was used instead due to the leaflet routing machine extension. The extension is very streamlined in providing the from end technology for working with OSRM. Leaflet is one of the most popular and easy to use JavaScript libraries making it one of the most obvious choices to provide this functionality.

In order for all these parts to interact with one another a web framework is required. GeoDjango integrates easily with an open source software stack, due to its accessibility to a PostGIS database, and integration with the Leaflet mapping library through the Django-leaflet open source extension.

Due to our lack of expertise in developing web applications, we found this to be quite difficult. We relied mostly on searching the internet for support and documentation. Most of our technological selections were a result of trial and error. Django and version IV proved to be the most difficult to implement due to its complexity. Overall, progress for iteration IV proved to be a challenge, and not all desired functionality was implemented in the final application.

5.3 Network Analysis

Research question: Which is the best route finding method to be implemented to assist in displaying routes to properties for sale from the nearest train station?

For the final version of our application we required the integration of a routing extension to display the shortest walking path from the train station to a selected property sale. Two alternatives were considered for the integration of the routing display: pgRouting and OSRM.

PgRouting involved the creation of additional database tables and the conversion of OSM data to a pgRouting readable format. This could be done with the osm2pgRouting conversion tool. Once the database had been set up the option to run a spatial query displaying the shortest path from point A to B could be calculated using the Dijkstra algorithm.

The process of setting up a pgRouting database and optimizing the routing calculations proved too time exhaustive, and due to the limited time and skill in our group, we elected to try the alternative method, OSRM.

OSRM is a routing engine that calculates shortest path between selected points using contraction hierarchies. This requires a lot of pre-processing of the data prior to integrating it into the front end of an application. In order to have access to the walking profiles generated by the OSRM project, we were required to clone the project's backend onto our own virtual machine and then pre-process the data we acquired from OpenStreetMap for Brisbane, Australia. The pre-processing phase for OSRM proved much simpler than the setting up pgRouting, due to the documentation provided by Project OSRM. Once the routing capability was included in the front end of our application, OSRM provided the desired capability by calculating and displaying routes in just milliseconds.

OSRM was selected for the final version of the application (version IV), because it provided the desired functionality and, given the relative programming inexperience of the group, was more user friendly to implement compared with the alternative - pgRouting. Testing the OSRM and pgRouting options, OSRM proved faster and more accurate than pgRouting.

5.4 Remote Group Work

Research question: How can we create the application whilst the team is working remotely from different parts of the world?

This project was the collaboration of four individuals working in different time zones around the world for much of the project. At the commencement, physical meetings with the group and project supervisor occurred on a regular basis, and then online meetings were conducted approximately every week and whenever deemed necessary once group members separated. This was achieved by using Skype for group meetings, Microsoft OneNote for sharing ideas, and Google Docs to draft the report.
Group meetings on Skype proved most effective when video conferencing was available, to allow viewing of flow diagrams and avoid speaking over the top of each other. Video conferences were conducted rarely however, as often one or more group members had poor internet access, so the meetings were conducted with voice alone. Skype proved a very effective method to conduct group meetings overall, due to the generally high voice quality and free access.

Microsoft OneNote was a highly suitable platform for sharing ideas, creating task lists, and recording meeting minutes. It updated information in real time and provided extremely versatile formatting. It was free to use and operated on all operating systems, however it was noted that different editing options are available on different platforms (eg. Microsoft version provides different shapes to insert than Mac OS). A mobile app is also available, which proved useful to keep updated whilst away from the computer.

Google Docs proved a useful platform to draft the report, however several limitations were noted. It was useful because it is free to use, compatible with multiple web browsers, and updates in real time. It indicated when other group members were online, and this function was often used to discuss topics. Limitations included no capability to add captions to Figures, slow response time, and formatting issues when exporting to a Word document. Overall the positives outweighed the negatives, and the group is content with the use of Google Docs throughout the project.

The prototype developed for this project compared with the original idea differ significantly. This is due to a range of factors including:

- Limited time
- Limited experience developing such applications
- Lack of programming knowledge
- Lack of experience sharing work (tasks)

5.5 Future Directions

The concept of TSPF is novel and is not currently utilized in the available property sales portals in Australia. The current search options are limited to searching by location (eg suburb or city) rather than location relative to other points of interest.

However, the application has a number of limitations and weaknesses in its current state; no live sales data, hyper localized, and is only focused on train stations. In order to rectify these weaknesses, a number of improvements are suggested. These include, integration of live sales data through a property sales portal API, inclusion of a larger area of interest e.g. the entire city of Brisbane, and the option for users to search relative to other points of interest, such as schools, hospitals or supermarkets.

The application itself is in a primitive, early stage due to our limited time and technical knowledge. However, it is useful as a proof of concept for the possibility of creating such an application the usefulness it provides to its end users. Future version would prove to be faster, more visually appealing and more user friendly.

People are more interested in using public transport, and current property search options fail to provide an adequate solution to finding property that is within and easily reachable distance of a public transport hub. Our application aims to fill in this gap in the market by providing alternative search capabilities to the traditional property search options.

6 Conclusion

In conclusion, whilst developing our application, in a trial and error approach we have answered all research questions in a timely and effective manner. Whilst the application has not been completed to the initially proposed standards, with additional time and technical knowledge a more effective, faster, and user friendly application can be created. The research questions and their answers are as follows:

1. What data and data quality could be used in the design and implementation of a web based application to determine the closest railway station to a property for sale?

Data for this project was selected from several sources - OpenStreetMap, Queensland Government Spatial Catalogue, Australian Bureau of Statistics, and Australian real estate portal domain.com.au. These sources provide free data access and are considered reliable.

2. What technologies could be used to design and implement a spatially enabled property search engine?

The majority of the application was developed using open source software. The database consists of a PostgreSQL database, with a PostGIS extension. The raw data has been pre-processed, filtered and manipulated to suit our needs using ESRI ArcMap, or QGIS. The travel isochrones were pre-processed using Grass GIS and served to the map via GeoServer.

On the front end, we have created a web application using the python based web framework Django with the GeoDjango module utilized to take advantage of the spatially enabled data. The routing engine used was OSRM, as this provided better, more timely functionality than pgRouting, and the final results were displayed to the client using the JavaScript mapping library LeafletJS.

3. Which is the best route finding method to be implemented to assist in displaying routes to properties for sale from the nearest train station?

OSRM was selected for the final version to provide the routing functionality to the application. The OSRM Project backend was cloned to a local virtual machine in order to better utilize the walking profiles of the routing engine.

OSRM proved to be much simpler to implement out of the box than the alternative – pgRouting. When comparing the two options for speed and accuracy, OSRM also proved to provide faster and more accurate responses to routing queries.

4. How can we create the application whilst the team is working remotely from different parts of the world?

A combination of group sharing software was utilised allowing us to organise, work on the same document, and communicate as a group. The software used includes Skype, Microsoft One Note and Google docs.

7 References

Aberle L., Entity Relationship Diagram. Retrieved December 7, 2016, from http://searchcrm.techtarget.com/definition/entity-relationship-diagram

Akash A., (2015, February 5) What is HTML? [Web log post] Retrieved from https://alleasytricks.wordpress.com/2015/02/05/what-is-html/

Armstrong, Roger J. (1994). "Impacts of Commuter Rail Service as Reflected in Single-Family Residential Property Values." The National Academies of Sciences, Engineering, Medicine, 88–98**.**

Aruodas. Home Page. Retrieved December, 10, 2016, from www.aruodas.lt

Ball, J. & Srinivasan, V.C. (1994). Using the Analytic Hierarchy Process in house selection, *Journal of Real Estate Finance and Economics*, Springer accessed 01 November 2016

Banfi, S., Farsi, M., Filippini, M., & Jakob, M. (2008). Willingness to pay for energy-saving measures in residential buildings. *Energy economics*, *30*(2), 503-516.

Barber, C.G., Haque, N., & Gardner, B. (2009). 'OnePoint' – combining OneNote and SharePoint to facilitate knowledge transfer. *Drug Discov. Today,* 14, 845 - 850.

Bauer, U., Holz-Rau, C., Scheiner, J., & Walther, M. (2003) Entscheidungsprozesse regionaler Wohnstandortmobilität. *Zwischenbericht AP, 131*

Benjamin, John D., and G. Stacy Sirmans. (1996) "Mass Transportation, Apartment Rent and Property Values." The Journal of Real Estate Research 12 (1): 1–8.

Blanchette, J. (2008) The little manual of API design. *Trolltech*, Nokia.

Carl, P. (2000). Urban density and block metabolism. In K. Steemers, & S. Yannas (Eds.), Architecture, city, environment. Proceedings of PLEA 2000 (pp. 343–347). London: James & James

Cervero, R. (2006). Public Transport and Sustainable Urbanism: Global Lesson. *University of California Transportation Center*.

Chubirka, M., Open-Source Vs. Commercial Software: A False Dilemma. Retrieved December 5, 2016, from http://www.informationweek.com/strategic-cio/it-strategy/open-source-vs-commercial-software-a-false-dileekemma/d/d-id/1252665

Comer, D. (1979), The Ubiquitous B-Tree, Computing Surveys, 11 (2): 123–137

Cowburn, P. (2016) PHP Manual. Retrieved January 3, 2017, from http://dk.php.net/manual/en/index.php

Crockford, D., (2008) JavaScript: The Good Parts: The Good Parts. Publisher: O'Reilly Media

Dekeyser, S., & Watson, R. (2007) Extending Google Docs to Collaborate on Research Papers. *The University of Southern Queensland, Australia*.

Django [Computer Software]. (2016) Documentation. Retrieved from https://docs.djangoproject.com/en/1.10/

Docforge. Web Application Framework. Retrieved December 16, 2016, from https://web.archive.org/web/20150723163302/http://docforge.com/wiki/Web_application_framework

Downey, A., Elkner, J., & Meyers, C. (2002). How to Think Like a Computer Scientist Learning with Python.(1era ed.).

Ducket, J., (2010) Beginning HTML, XHTML, CSS, and JavaScript. Wiley Publishing, Inc

Enemark, S. & A. Rajabifard (2011), Spatially enabled society, Geoforum Perspektiv no. 20: Spatially enabled society – do geographical data play a societal role? Geoforum.

Entity Relationship Diagram (ERD) Tutorial. Retrieved December 9, 2016, from https://www.lucidchart.com/pages/er-diagrams

Esri. ArcGIS for Server Features. Retrieved December 15 December, 2016, from http://www.esri.com/software/arcgis/arcgisserver/features

Evans, Sarah C. (2016) "Property value changes from public transportation: How the Greenbush Commuter line affected property values," Journal of Environmental and Resource Economics at Colby: Vol. 3: Iss. 1, Article 4.

Fajfar I., (2015). Start Programming Using HTML, CSS, and JavaScript. CRC Press

Farmer J., Interview Questions: Database Indexes. Retrieved December 7, 2016, from http://20bits.com/article/interview-questions-database-indexes

Flanagan D., (2006) JavaScript: the definitive guide. O'Reilly Media

GeoNet (2016, March 2). Simplified Capacity Planning Tool (CPT) -- please! [Web log post]. Retrieved from https://geonet.esri.com/ideas/11927

GeoSever. GeoServer 2.11.x User Manual. Retrieved December 1, 2016, from http://docs.geoserver.org/latest/en/user/index.html

GIS Geography. (n.d.) 27 Differences Between ArcGIS and QGIS – The Most Epic GIS Software Battle in GIS History. [Web log post] Retrieved from http://gisgeography.com/qgis-arcgis-differences/

Goldratt M. (2002), Critical Chain Eliyahu, The North River Press.

Goodman D., (2007) JavaScript bible. Publisher: John Wiley & Sons

Güting, Ralf Hartmut. "An introduction to spatial database systems." *The VLDB Journal—The International Journal on Very Large Data Bases* 3.4 (1994): 357-399.

Han, J. H., Kim, J. Y., & Kim, J. (2016). Dynamics of Housing Mobility in Australian Metropolitan Areas, 2001–2010: A Longitudinal Study. *Urban Policy and Research*, 1-14.

Hansen, H.S., Hvingel, L. & Schrøder, L. (2013). Open Government Data – A key element in the Digital Society. *Lecture Notes in Computer Science*, vol. 8061

Hess, Daniel Baldwin, and Tangerine Maria Almeida. 2007. "Impact of Proximity to Light Rail Rapid Transit on Station-Area Property Values in Buffalo, New York." Urban Studies 44 (5/6): 1041–68.

IBM. Relational database. Retrieved December 9, 2016, from http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/reldb/

Intergovernmental Panel on Climate Change. (2015). *Climate change 2014: mitigation of climate change* (Vol. 3). Cambridge University Press.

Jabareen, Y. (2013). Planning the resilient city: Concepts and strategies for coping with climate change and environmental risk. *Cities*, *31*, 220-229.

Leach, L.P. (2014). Critical Chain Project Management. Artech House

Kalter, F. (1994). Pendeln statt Migration?. *Zeitschrift für Soziologie*, *23*(6), 460-476.

Kavas, S., and Topçu Y, I. (2014) AHP based decision model for appraising residential real estates in an abstracted zone. International Symposium of the Analytic Hierarchy Process, Washington D.C., U.S.A. accessed 01 November 2016

Kavourgias, C. (2015) What's the Difference Between the Front-End and Back-End? [Web log post] Retrieved from http://blog.digitaltutors.com/whats-difference-front-end-back-end/

Kessler, C. (2016) Lecture No. 5, Aalborg University, Copenhagen. Accessed 20 December 2016

Krämer-Badoni, T., & Kuhm, K. (2000). Mobilität. In *Großstadt* VS Verlag für Sozialwissenschaften.

National Association of Realtors. Digital House Hunt: Consumer and Market Trends in Real Estate. Retrieved December 10, 2016, from https://www.nar.realtor/sites/default/files/Study-Digital-House-Hunt-2013-01_1.pdf

Michaelis, C.D., & Ames, D.P. (2008) Web Feature Service (WFS) and Web Map Service (WMS). *Encyclopaedia of GIS*, Springer, US

Momjian, Bruce. PostgreSQL: introduction and concepts. Upper Saddle River, NJ: Pearson Education Corporate Sales Division, 2001.

Neteler, M., & Mitasova, H. (2013). *Open source GIS: a GRASS GIS approach* (Vol. 689). Springer Science & Business Media.

Neteler, M., Bowman, M. H., Landa, M., & Metz, M. (2012). GRASS GIS: A multi-purpose open source GIS. *Environmental Modelling & Software*, *31*, 124-130.

Olsson M., (2014). CSS Quick Syntax Reference. Apress

OpenSource. What is open source? Retrieved December 12, 2016, from https://opensource.com/resources/what-open-source

OSGeo OpenLayers Info Sheet. Retrieved December 16, 2016, from http://www.osgeo.org/openlayers

Peng, Zhong-Ren, og Ming-Hsiang Tsou (2003) Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks. *John Wiley & Sons*, Hoboken, New Jersey

Peters, D. (2014) Building a GIS: Implementation Strategy and Best Practices. *ESRI Press*, accessed 01 December 2016

Pickett-Baker, J., & Ozaki, R. (2008). Pro-environmental products: marketing influence on consumer purchase decision. *Journal of consumer marketing*, *25*(5), 281-293.

Pihkala, Kari, Mikko Honkala, and Petri Vuorimaa (2013) "Multimedia web forms." Proc. Synchronised Multimedia Integration Language European Conference, SMIL Europe.

Python Software Foundation. General Python FAQ. Retrieved November 22, 2016, from http://docs.python.org/faq/general

Realestate.com.au. Home Page. Retrieved January 7, 2017, from www.realestate.com.au

Rigaux, P., Scholl, M., & Voisard, A. (2003). Spatial Databases with application to GIS. *SIGMOD Record*, *32*(4), 111.

Robbins, R. J. (1994). Database Fundamentals. Johns Hopkins University

Rising, J (2009, May 6). Sashimi Waterfall Software Development Process. [Web log post] Retrieved from http://documents.mx/documents/sashimi-waterfall-software-development-process-eunice.html

Sack, C (2015, April 4). Connecting PostGIS to Leaflet Using PHP. [Web log post] Retrieved from https://northlandia.wordpress.com/2015/04/20/connecting-postgis-to-leaflet-using-php/

Samet, H. (2002). Spatial databases and geographic information systems (GIS), University of Maryland, College park.

Scheiner, J., & Kasper, B. (2003). Lifestyles, choice of housing location and daily mobility: the lifestyle approach in the context of spatial mobility and planning. *International Social Science Journal*, *55*(176), 319-332.

Scheiner, J. (2006). Housing mobility and travel behaviour: A process-oriented approach to spatial mobility: Evidence from a new research field in Germany. *Journal of Transport Geography*, *14*(4), 287-298.

Schlossberg, M., & Brown, N. (2004). Comparing transit-oriented development sites by walkability indicators. *Transportation Research Record: Journal of the transportation research board*, (1887), 34-42.

Shannon R., (2012, August 21). What is HTML? [Web log post] Retrieved from http://www.yourhtmlsource.com/starthere/whatishtml. html

Shukla, D., Shivnani, C., & Shah, D. (2016) Comparing Oracle Spatial and Postgres PostGIS. *International Journal of Computer Science and Communication*, 7(2), pp 95 - 100
Sian, D. (2012 August 14). Critical chain project management methodology. [Web log post] Retrieved from http://creativeagencyprocess.com/limiting-wip-with-ccp/

Smets, A.J.H., 2000. Wervende Woonmilieus in de Stad? Stedelijke Herstructurering Gñëvalueerd. Nederlandse GeograWsche Studies 276. University, Utrecht.

Timothy A., Agile Project Management In eLearning Development. Retrieved December 2, 2016, from https://elearningindustry.com/agile-project-management-elearning-development

Trulia Maps. Commute times in Las Vegas. Retrieved December 10, 2016, from https://www.trulia.com/local/las-vegas-nv/driving:1%7Ctransit:0%7Cposition:36.166498;-115.255793%7Ctime:60_commute

Tsatsaronis, K., & Zhu, H. (2004). What drives housing price dynamics: cross-country evidence. *BIS Quarterly Review, March*.

Woodruff, A. and Mullins, R. (n.d) Leaflet: Make a Webmap!. [Web log post] Retrieved from https://maptimeboston.github.io/leaflet-intro/