University of Huddersfield

MEng Group Project

Cryptic Crossword Solver

Research Document

Authors:
Mohammad Rahman
Leanne Butcher
Stuart Leader
Luke Hackett

Supervisor:
Dr. Gary Allen
Examiner:
Dr. Sotirios Batsakis
Moderator:
Dr. Colin Venters

Friday, 9^{th} May 2014

Contents

List of Figures

List of Tables

Chapter 1

Research

In the previous chapter a detailed approach to what the project will cover was discussed. In order to correctly implement a working piece of a software a number of areas will need to be researched thoroughly.

Based upon the problem analysis, the the main topics that have been identified are:

- Cryptic Crosswords
- Natural Language Processing
- Application Platform
- Web Services

1.1 Crosswords

Arthur Wynne produced the first crossword puzzle which was printed on December 21st 1913. A crossword is a puzzle which involves the solver resolving the answer to a clue and placing it in the correct space within the grid.

A grid is made up of black and white squares, the black squares are blanks and the white squares are where the solver must place the answers. A crossword grid comes with a set of clues. The clues are usually arranged based upon their positing within the grid. Clues that that appear downwards in the grid are kept separate to the clues that appear across the grid.

The white squares which are used for the first letter of an answer to a clue usually have a number in the top left hand corner to indicate the clue which links to this area of the grid.

There are different types of crosswords such as quick, cryptic and double-clue. A quick crossword has clues which simply define the answer. A cryptic crossword is more complex as it has word play as well as simple definitions and many different types of clues. A double-clue crossword combines the two and allows for a simpler option for the solver when the cryptic clues become too difficult.

1.1.1 Cryptic Crosswords

Most cryptic clues consist of two different parts, the word play and the definition itself. The definition is like a clue found within a quick crossword and the word play is an indication to the answer.

Clue types such as double definition and purely cryptic break the usual format of cryptic clues. Double definitions miss out the word play whereas the purely cryptic clues miss out the usual simpler definition and become a fully cryptic definition. Other types of clue add to the usual format when smaller clues are embedded within the larger clue to assist with the word play.

Punctuation within clues should always be disregarded unless it is a question mark or an exclamation mark. Punctuation such as commas and hyphens are used to distract the solver from the answer usually by attempting to dictate how the clue is read.

A question mark tells the solver that the clue requires creative thinking to work out the answer which could be witty in nature. An exclamation mark can mean that the word play and the definition may intersect which is otherwise known as a clue of the type "& lit". Articles within clues can also be very important and should not be disregarded when reading the clue.

Cryptic clues which have a particular tense will always be for a clue with the same tense. Similarly a plural clue determines that the answer will also be plural.

1.1.2 Crossword Clue Types

An important skill needed to solve a cryptic crossword is to be able to spot the type of clue given. Below is a list of the most common types of cryptic clue and expected rules the clue should follow so they are identifiable.

Purely Cryptic

Although clues within a cryptic crossword usually include both a definition and word play, this type of clue is an exception because the whole clue is a definition written in an unusual way.

Word play within other clue types assist the user in being able to determine their answer is correct as well as solve them, because there is no word play more than one answer could be found which are incorrect. Possible indicators that denote a purely cryptic clue are:

- Question mark
- Exclamation mark

Example: *Frames for summer's activities? (5)* - (Upadhyay, 2008c)
Answer: ABACI

- "Summer" as in a person who does mathematical sums
- An abacus is a frame which holds moving beads
- As the clue is plural so must the answer be, hence abaci

Hidden

The answer for a hidden clue is concealed within the clue itself and can be spread over more than one word as well as possibly being hidden in reverse. The clue will have a definition, an indicator that the answer is hidden within the clue and a word or set of words which have the answer in them. Possible indicators that denote a hidden clue are:

- Word/s e.g. contains, in, within, held by, from
- Large words with a hidden word indicator before it may have the answer inside them
- A clue which seems inelegantly written or a clue which contains proper nouns

Example 1: *Metal concealed by environmentalist (4)* - (Upadhyay, 2008e)
Answer 1: IRON

- "Concealed by" is a phrase indicator for hidden clues
- "Environmentalist" is a large word with an indicator in front
- "Metal" is the definition so the answer is a type of metal which can be found within the word "environmentalist", hence iron

Example 2: *Mountain range in central Taiwan (5)* - (Upadhyay, 2008e)
Answer 2: ALTAI

- "in" is a word indicator for hidden clues
- "central Taiwan" contains a proper noun which indicates the answer is hidden here
- "Mountain range" is the definition
- Without knowledge of mountain ranges the answer could be narrowed down down to the following words (assuming "central" would not be within the clue without a purpose):
  - TRALT
  - RALTA
  - ALTAI
  - LTAIW
- If some of the crossword is completed within the area this clue is placed, the correct answer could be found through trial and error.

Charades

A charade clue forms its answer with the use of smaller answers to smaller clues within the main clue. Abbreviations and first/last letters of words are common within charade clues to make up the complete answer.

Two or three parts are usually within the charade clue to solve the correct answer, they may not be in the right order however, and word indicators will be used to warn the solver.

Other types of clues can also be used within charade clues for the different parts such as reversals and homophones, if this is the case there will be indicators for the specific type. Possible indicators that denote a charade clue are:

- Words e.g. with, follows, behind, after to indicate joining of answers to parts of the clue

Example: *Prior belted one that is ultimately right (7)* - (Upadhyay, 2008b)
Answer: EARLIER

- "Prior" is the definition
- "belted one" gives earl
- "that is" gives the abbreviation for i.e. or ie
- "right" gives the abbreviation for r
- All the segments put together give the word "earlier" which can also mean "prior"

Anagrams

Anagram clue types have a definition, a word or phrase to indicate the clue is of this type and an element called "fodder".

An anagram is a word whose letters can be rearranged to form another word; within an anagram clue the letters to rearrange are known as "fodder" and are placed next to the indicator. Possible indicators that denote an anagram clue are:

- Words which could mean change or shifting
- A clue which seems inelegantly written or a clue which contains proper nouns

Example: *Toy breeds trained to find out a place for pearls (6,3)* - (Upadhyay, 2008f)
Answer: OYSTER BED

- "trained" indicates the clue is an anagram
- "Toy breeds" is an abnormal phrase and is the "fodder" of the clue
- to find out a place for "pearls" is left to become the definition
- "Toy breeds" is then moved around to give oyster bed

Homophones

A homophone is a word which sounds like another word but has a separate meaning. This type of clue has a definition, a word or phrase which means the same as the homophone to find and an indicator. Possible indicators that denote a homophone clue are:

- Words which indicate hearing or sound e.g. said, heard
- Normally the indicator for a homophone is next to the word or phrase which is to be used to find a homophone

Example: *Refer to a location, reportedly (4)* - (Upadhyay, 2008j)
Answer: CITE

- "reportedly" is the homophone indicator
- "a location" is the phrase which needs to be used to find a homophone
- "Refer to" is the definition
- A location can be otherwise known as a "site", hence cite

Acrostics

An acrostic clue commonly involves picking the first letter from a group of words and putting them together to form the answer. It is possible that the clue will require the last or middle letters from words to solve them or that the letters should be put together in reverse order.

This clue type has a definition and an indicator as well as "fodder" which in this case means the group of words the necessary letters will come from. Possible indicators that denote an acrostic cryptic clue are:

- Words which could mean start or beginning
- If the clue is unusually long and so is the number indicator to determine how long the answer should be

Example: *Some URLs recommended for beginners to explore online (4)* - (Upadhyay, 2008a)
Answer: SURF

- "beginners" is the indicator
- As the clue states the answer should be of length four it is assumed "to explore online" is the definition as there are only three possible letters for an acrostic in the phrase
- "Some URLs recommended for" is the "fodder" for the clue. The indicator implies that the first letters should be taken from the first letters of the words within the "fodder", hence surf

Palindromes

A palindrome is a word which reads and looks the same when it is reversed. This type of clue has an indicator and a definition. Possible indicators that denote a palindrome clue are:

- Phrases which may mean either way or going around in circles

Example: *Unacceptable, going up or down (3,2)* - (Connor, 2012a)
Answer: NOT ON

- "going up or down" is an indicator as it could mean "in either direction" like the format of a palindrome
- "Unacceptable" is then left as the definition which gives the answer, not on

Reversals

A reversal requires the solver to reverse a number of letters to give a new word. The clue consists of a definition, an indicator that the clue is a reversal clue and some "fodder" which is a phrase or word which could contain the letters to be reversed or a smaller clue which leads to the letters which need to be reversed. Possible indicators that denote a reversal clue are:

- Words/phrases used for directional purposes e.g. left, up
- The word indicators may also be relative to the direction the clue should be placed within the crossword (down or across)

Example: *Stop the flow in crazy get-up (3)* - (Upadhyay, 2008h)
Answer: DAM

- "Stop the flow" is the definition
- "get-up" is the indicator that the clue is a reversal
- Another word for "crazy" is mad which reversed gives the answer dam

"& lit"

"& lit" clues, which means "and literally so", is a type of clue where the definition and the word play are the same and are not split out into separate phrases or words as with other clues. The definition is the whole clue and the word play can be one or more of any of the normal clue types such as anagrams and charades. Possible indicators that denote a "& lit" clue are:

- Exclamation mark

Example: *Cop in male form (9)* - (Upadhyay, 2008g)
Answer: POLICEMAN

- The whole clue is the definition
- "form" is an indicator for an anagram clue
- "Cop in male" can be rearranged to policeman which is also the answer to a "Cop in male form"

Double Definition

A double definition clue has no word play and is purely a clue with two (or possibly more) definitions which lead to the same answer. Possible indicators that denote a double definition clue are:

- Possibly shorter than most clues (2 or 3 words)
- Although it is advisable to ignore punctuation when solving cryptic clues, a double definition may have a piece of punctuation separating the definitions.

Example: *Robust author (5)* - (Upadhyay, 2008d)
Answer: HARDY

- Both words are separate definitions which could both mean hardy

Containers

A container clue includes three definitions and an indicator. One of the definitions is for the final solution whereas the other definitions describe two separate answers where one is contained within the other.

They are similar to charade clues in the way that other types of clues can be used within container clues such as anagrams and charades themselves. Possible indicators that denote a container clue are:

- Word indicators which could be used to indicate the inner word (e.g. inside, held) or the outer word (e.g. outside, external)

Example: *Building for the workers in principle (8)* - (Upadhyay, 2009a)
Answer: TENEMENT

- "principle" is the definition for the outer word which gives tenet

- "in" is the indicator that the answer for the inner word will be placed within the outer word (tenet)
- "the workers" is the definition for the inner word which is men
- 'Building' is the overall definition which could mean tenement which is also given when men is put within tenet

Deletions

Deletion clues require the solver to retrieve the answer by looking at the definition and the word play and removing the correct letters from the correct word. The clue will not usually have the word which needs letters removing from it directly within the clue. Possible indicators that denote a deletion clue are:

- Words to indicate letters should be removed from a certain place within the word such as its first or last letter
- Words to indicate certain letters should be removed from the word found. These could be words that can be abbreviated .

Example: *Little shark edges away from diver's equipment (3)* - (Upadhyay, 2009b)
Answer: CUB

- "edges away from" indicates that the first and the last letter should be removed from the answer which comes from "divers equipment"
- "divers equipment" is otherwise known as scuba
- Removing the "s" and the "a" from scuba leaves the word cub which is also the answer given from the definition "Little shark"

Spoonerisms

A spoonerism clue has a definition and word play which is usually a phrase which describes another. This phrase, which is usually two words long, is taken and the first letter of each is swapped around to gain a new phrase which is in turn the answer to the clue. Possible indicators that denote a spoonerism clue are:

- The word "Spooner" or "Spooners" is placed within the clue

Example: *Spooner's cheerful enthusiast? He'll get you across (8)* - (Connor, 2012b)
Answer: FERRYMAN

- "Spooners" indicates that this clue is a spoonerism
- "Hell get you across" is used as the definition
- "cheerful enthusiast" could also be known as a merry fan
- Swapping the first letters from the words merry and fan give the answer ferryman

Pattern

This type of clue has a definition, an indicator that the clue is of the type pattern and a phrase or word which has the answer within them arranged as a pattern. These patterns can be odd or even letters joined together to make a word or possibly letters picked from regular intervals.

Pattern word play can also be joined together with other types of clues such as charades to form a more complex answer. Possible indicators that denote a pattern clue are:

- Words which could mean even, odd or routine

Example: *Beasts, free, ginned, we hear regular losses there! (8)* - (Upadhyay, 2009d)
Answer: REINDEER

- "regular" indicates it is a pattern clue as well as "losses" to indicate the dropping of letters.
- "Beasts" is the definition.
- "free, ginned, we hear" holds the answer reindeer by picking out the first "r" within free and each letter alternately from then on.

Substitutions

A substitution clue involves removing letters from a word and replacing it with another to retrieve the answer. There are two definitions within the clue, one definition to retrieve the word to substitute letters from and another to define the final answer.

The letter or letters to substitute are usually an abbreviation which can be found within the clue itself. Possible indicators that denote a substitution clue are:

- Words which mean substitution e.g. replace, switch, exchange

Example: *Unexciting story gets mark for length (4)* - (Upadhyay, 2008i)
Answer: TAME

- "mark for length" is an indicator for a substitution clue
- "mark" can be abbreviated to "m" and "length" can be abbreviated to "l", therefore replace "l" with "m"
- "story" is the definition for the word which needs the substitution and could be defined as a tale
- Replacing the "l" in tale with "m" gives the word tame which can also mean unexciting

Shifting

A shifting clue has an indicator, a definition of the final answer and another definition for the word which needs to be used to shift a letter to a different position within the word to find the final answer.

The shifting of a letter could be moving the first letter to the last position in the word or in a more complex clue letters could be shifted within the middle of the word. Possible indicators that denote a shifting clue are:

- Words e.g. shift, change, move
- Phrases e.g. head to foot

Example: *Character needs help, head to foot (4)* - (Upadhyay, 2009e)
Answer: BETA

- "head to foot" indicates moving a letter from the front of a word to the end
- "help" is the definition for the word which requires letter shifting and can also be defined as abet
- Moving the first letter of "abet" to the end gives the word beta which is a "Character"

Exchange

An exchange clue is similar to a shifting clue however instead of only one letter shifting positions within a word, two letters within a word exchange places to form a new word.

Typically the letters to exchange will be the first and last letters of a word or two letters next to each other, however it is possible more than one letter on each side will need to be swapped.

For example, the word "rage" can be split into two sections "ra" and "ge" which can then be exchanged to make the word "gear". Possible indicators that denote an exchange clue are:

- Words e.g. swap, exchange, change

Example: *Doomed king switching sides? True (4)* - (Upadhyay, 2009c)
Answer: REAL

- "switching sides" indicates that this clue is an exchange clue
- A "Doomed king" can also be known as a "Lear"
- "True" is the definition which can also be defined as "real" which can be gained by exchanging the first letter of "Lear" ("L") with the last letter ("r")

1.2 Natural Language Processing

An inherent disparity between the languages used and understood by human beings and those which are interpreted by machines introduces the requirement for a system which is able to provide a form of compatibility between the two. Natural language can be described as a system of communication that has not consciously been invented (Collins, 2004) and typical examples of this are the languages used by human beings to communicate with each other. English and Korean are two such examples of natural languages and these evolve over time as words are added and removed and the rules that define the language are refined or adapted, meaning there is no immutable grammar for the language.

Conversely, formal (or artificial) languages are those which have been fabricated for a specific purpose, and can be described using precise and unambiguous mathematical rules (Jiang et al., 2010). An example of such a mathematical rule is the BackusNaur Form (BNF) grammar. Programming languages such as Java or Python are examples of artificial languages, as these have been created with a specific purpose in mind. The vocabularies of these languages are well defined and free from ambiguities, making interpretation by the languages corresponding compiler a black or white task where the given language to compile is either valid or invalid.

Natural language processing (NLP) allows for the manipulation of natural languages by a computational device (Bird, 2009) and there are a number of identified steps of the NLP process which have been developed, from scanning and parsing, to meaning extraction and integration. In the context of solving cryptic crosswords, a given clue would need to be broken down into meaningful components which would allow, as an example, the word play and the definition components to be identified and passed forward to the appropriate algorithms to identify the correct answer.

```
Reversible fasteners pinning baker, European
(TOP (NP (NP (NNP Reversible) (NNS fasteners)) (VP (VBG pinning) (NP (NN baker,)
)) (. European)))
```

Figure 1.1: Creating a parse tree for a given sentence

(Halpern, 2013)

1.2.1 The Steps of Natural Language Processing

Scanning

A number of steps are typically involved in the processing of a natural language. Assuming the input format of the natural language is in a textual form as sentences (rather than as speech or handwriting); these chains of text may be broken down into their separate components - the words which make up the sentences. This process is known as scanning, and may itself comprise of several sub-steps. The first step could likely involve breaking down the given textual input into separate sentences (Apache, 2013).

```
C:\Users\u0955187\Downloads\apache-opennlp-1.5.3-bin\apache-opennlp-1.5.3\bin>op
ennlp SentenceDetector en-sent.bin < input.txt
Loading Sentence Detector model ... done (0.068s)
Reversible fasteners pinning baker, European
```

Figure 1.2: A single sentence is already in the desired state

Once the separation of any sentences present in the input has occurred, these can be further divided into the separate words to achieve an array-like data structure, such as [*Reversible*, *fasteners*, *pinning*, *baker*, *European*], a process which is referred to as tokenising. It is important to consider the inclusion and placement of any punctuation provided in the input text, as this may provide some indication as to the separation of the cryptic crossword components and increase the chances of successfully calculating the answer. In the example provided, European corresponds to the definition component of the clue, where Reversible fasteners pinning baker relates to the word play component. Both components have been clearly separated by a comma, though not all cryptic clues are presented in this fashion.

Parsing

Parsing is a process used to ensure that a given sentence conforms to a grammar and thus showing that it is syntactically correct (McCluskey, 1999), and this process can be demonstrated by creating a parse tree to represent the structure of a given sentence. It may become apparent when attempting to solve the clues of a cryptic crossword that certain elements of the provided clue sentences bear some relation to the correct answer. To give an example, it may prove more likely that the adjectives or verbs present in the input are synonyms of the correct answer, where there may be little probability that the answer stems from a synonym of any present determiners (the, this, a, some) or prepositions (on, beneath, over, under). Likewise, adjectives or adverbs that hint at a particular action such as *reversible* or *backwards* could reveal themselves as more likely to be indicators that describe in what way the clue should be manipulated.

Referencing

The ambiguous nature of cryptic clues may not allow for a straightforward parse of an already ambiguous (natural) language, possibly contributing to the complexity of the process of solving. It may be the case that multiple parses of the same clue have to be taken forward to be processed before the solution can be found. Once a parse tree has been selected for the given input, a number of steps can be performed, which attempt to reduce the level of ambiguity. Lee McCluskey outlines four methods which can be applied to achieve this:

- **Contextual Disambiguation:** Such as referring to segments of a previous piece of text. Fred Bloggs may have been declared in a previous sentence, and from that point referred to as he.
- **Physical Constraints:** This refers to the literal meaning of a sentence of phrase, and uses known facts to assist in the reduction of ambiguity. The phrase I answered the door in my pyjamas could be taken to mean that the subject was either wearing their pyjamas when they answered the door, or that the door was wearing the subjects pyjamas when they answered it. As it is unlikely that the door was physically wearing pyjamas, the parse suggesting this could be disregarded.
- **Default Roles in Known Verb Structures:** Some verbs may only be applied to a subset of subjects in order to make reasonable sense. In the example sentence fruit flies like a banana, one parse may take fruit as the noun, and flies as the verb. In this scenario, fruit is physically incapable of flying, and so an alternative parse which taken fruit flies as the noun-phrase is more likely to be the correct parse of the sentence.
- **General Defaults:** To say that one is going to visit an old friend may imply that the friend has been known for a long time, or that the friend is of old age. While the correct interpretation may be context specific, the likelihood is that one particular interpretation will hold precedence over another. In this case, the friend is likely to have been known for a long time.

(McCluskey, 1999)

A parsing algorithm is used to obtain a complete parse tree for a given sentence, providing insight into the complete structure of the sentences. It is possible that this may not be required for the task at hand; a full parse may provide more information than what is required to solve a cryptic clue. An alternative mechanism exists, chunking, which may also be referred to as *light parsing*. This technique divides a sentence into a series of non-overlapping chunks of text, which provide an overview of the input sentence (Litman, 2003). One of the major benefits of this technique is the removal of the need to resolve ambiguity, as chunks of the sentence are non-recursive, unlike parsing which produces components of a sentence which may be nested in each other.

```
Reversible_NNP fasteners_NNS pinning_VBG baker,_NN European._.
```

Figure 1.3: Assigning *Parts of Speech* tokens to an input sentence

Meaning Extraction

Meaning extraction refers to the process of taking the parse and storing it in a way which accurately represents the information it models (McCluskey, 1999). For example, the following cryptic crossword clues all map to the same answer, and whilst they are written differently, they each imply the same meaning of *DRILL*:

- *Doctor gives patients exercise*
- *Doctor needing a doctor for practice*
- *GP not fit for practice*

(Gordius, 2003)

Effective meaning extraction will ensure that each distinct clue may help to ensure that variations of the same will be mapped to a single, internal representation and this will allow for the answer to be retrieved from a database without having to recalculate what has already been processed once.

1.2.2 Natural Language Processing Libraries

There exists a range of NLP libraries that offer many natural language processing functions for a variety of programming languages. Development on a number of these libraries has become stale, and others are available only for use after purchasing a commercial licence. Below are a selection of available libraries which have recently been updated and are available under some form of free software licence.

Library	**Licence**	**Last Updated**	**Supported Languages**
OpenNLP	Apache 2.0	April 2013	Java
NLTK	Apache 2.0	November 2012	Python
Stanford CoreNLP	GNU General Public Licence	June 2013	Java

Table 1.1: A comparison of available NLP libraries

OpenNLP

OpenNLP provides the ability to carry out a range of natural language processing functions, including parsing, chunking and name finding, where the latter aims to recognise proper nouns in specified text such as person names. The features of the library utilise a maximum entropy model, and this allows the performance of the software to be enhanced as it uses training data to learn (Apache, 2013). This allows for each component of the toolkit to refine as the subset of training data increases over time, with the aim of increasing the accuracy of the corresponding NLP component. The library is available for use in a Java environment or can be accessed directly through a command line interface, but the general usage of the library requires providing an applicable model and language data as input, for which the model will be utilised for the language processing.

Models A number of models are provided for use with the NLP library. These include, but are not limited to, models to assist in the process of parsing, chunking and detecting sentences. A number of libraries also exist which allow for the identification of person names, company names, times, dates or location in input text. These existing models may be further trained in a bid to increase their effectiveness.

DocumentCategoriser Another prominent feature of the library is the ability to classify input text into a range of predetermined categories using the aforementioned maximum entropy model. Once a model has been created, which contains a series of example inputs along with their corresponding categories, further inputs to the system will be paired with a *best outcome* classification.

```
<terminated> NLP [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (Nov 1, 2013 3:18:08 PM)
Category for: Posy says no, according to Spooner is: SPOONERISM (0.18832631809541703)
Category for: Preparation for documents once Spooner's transporting luggage is: SPOONERISM (0.005477603261500726)
Category for: Preserve holy one appointed by Spooner is: SPOONERISM (0.003470437596580894)
Category for: Theme''s proper order reversed is: REVERSAL (0.016981012268904402)
Category for: This compiler going to the stake? Quite the reverse! is: REVERSAL (3.198100077871112E-4)
Category for: This vehicle reverses into its club is: REVERSAL (0.01006333130189762)
```

Figure 1.4: A demonstration of the OpenNLP DocumentCategorizer

NLTK

The Natural Language Toolkit is an alternative library which exists with an open-source licence variant. NLTK exists for use with the Python programming language and possesses a similar infrastructure to OpenNLP, where trained models are used to allow each component of the software package to function correctly (NLTK Project, 2012). The package is

supplemented by a comprehensive e-book, detailing and providing examples of the usage of each component available.

1.3 Mobile Platforms and the Market

The demand for applications to be portable on mobile phones and tablets has become a fashion as well as accessibility on the go. In the current market Cryptic crosswords are available in a variety of newspapers, magazines and online websites. These are everyday media in which a person can access their favourite Cryptic Crosswords and participate in. With the expansion of mobile phone and tablet applications research has shown that a unique Cryptic Crossword solver has not been incorporated in any of the mobile operating systems. There are applications, which can be downloaded, that are pre-solved crosswords but there is no real solver, which solves Cryptic Crosswords in real time. In fact research has shown that there is only a very small handful of Cryptic Crossword applications across all mobile operating systems.

1.3.1 Mobile Operating Systems

The Oxford English dictionary states an "Operating System" as 'the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals' (Dictionaries, 2011).

A mobile operating system has the same definition but supports the basic functions for handheld devices such as mobile phones and tablets. In the current market the term mobile operating system is associated with smartphones rather than mobile phones due to the powerful processors, which are embedded in the mobile phone devices.

The official first mobile operating system for smartphones was by IBM introduced in 2000 as the 'Simon' but it was Ericson who created the first all in one smartphone with the Symbian OS, which incorporated a keyboard hence, the term smartphone was introduced. This ran on various mobile phones by companies such as Ericcson, Samsung but predominantly on smartphones by Nokia. This created a path for a market, which was to be dominated by others in the coming future (Akman, 2013).

By 2007 from the back of a successful campaign selling music devices known as the iPod, Apple introduced the Apple iPhone which came with their fist mobile operating system the iOS which was a full operating system used from the Mac OS X 10 (Apple, 2007).

By July 2008 Apple released iOS 2.0 and with this came the application store. This was a revolution to the mobile market allowing a platform for third party developers to sell and market their own applications for the mobile operating system. On the 7th January 2013 Apple announced that they have had more than 40 Billion downloads of applications through their application store (Apple, 2013).

In September 2008 Google released their own version of a mobile operating system Android' which had its similarities of marketing with its very own application store known as the

android market which since has been rebranded to the Google play store.

In April 2009 BlackBerry also launched it's own application store called the BlackBerry World, which works with their mobile operating system the BlackBerry 10 (Cha, 2009).

Finally windows released their version of an application store for distribution of applications on October 26th 2012 (Businessline, 2012).

Although there are now several platforms that run on a mobile device, Android and iOS combine for 91.1% of the Worldwide Smartphone OS Market (IDC, 2013).

This shows that although BlackBerry has been in the market since 2009 there isn't much of a rise to interests in their applications and while Windows is fairly new it has a lot of ground to cover to catch up with their main competitors.

For the purpose of this project it is pretty clear to what is demanded from consumers in the real world with Apple and Google being the two main competitors for mobile applications. In order to decide on what platform the project will be suitable for to design, maintain and deploy a good working product below is the following page contains a table which covers some of the reasons which could be possible to allow the team members to come to a decision to what pathway the project will be going in.

Platform	**Programming Language**	**Open Source**	**Open API/SDK**	**License**
Android	Java	Yes	Yes	Apache
iOS	Objective C	N/A	Yes	Proprietary
BlackBerry	Java, C++	N/A	Yes	Proprietary
Windows	C, C++, C#	N/A	Yes	Microsoft

Table 1.2: A comparison of mobile platforms

Platform	**Latest Version**	**Debugging**	**Hardware / Software Requirements**	**Emulator**
Android	4.3 Jelly Bean	Yes	Windows / Mac	Yes
iOS	iOS 7	Yes	Intel Based Mac	Yes
BlackBerry	BlackBerry 10	Yes	Windows / Mac	Yes
Windows	Windows Phone 8	Yes	Windows	Yes

Table 1.3: A further comparison of mobile platforms

Platform	Underlying OS	Development Environment	Submission To application Store	Development Cost
Android	Linux	Eclipse	Unlimited	$25 One Off cost
iOS	Darwin	XCode	Unlimited	£60 Yearly / $99
BlackBerry	BlackBerry OS	Eclipse	100 Per Year	$100 One off cost
Windows	Windows	Visual Studio	Unlimited	$19 Yearly / Free for Dreamspark Students

Table 1.4: More comparisons of mobile platforms

Although these are the four main types of platforms available in the market for mobile development there is a growing amount of organisations, which are developing tools to allow developers to create cross platform application with ease. Two of these are Appcelerator and Adobe AIR.

Platform	Programming Language	Open Source	Open API/SDK	Underlying OS and License	Development Environment
Appcelerator	JavaScript	Yes	Yes	Linux Apache 2.0	Eclipse Based IDE / Titanium Studio
Adobe AIR	ActionScript, HTML, CSS, JavaScript	No	Yes	Darwin Proprietary	Adobe AIR

Table 1.5: A comparison of cross platform development tools

1.3.2 Appcelerator

Appcelerator is a platform created by Appcelerator Inc to allow developers to create cross platform native applications for Android and iOS. They later introduced compatibility for BlackBerry 10 and are in the process of developing for Windows Phone. The main development environment used to create applications is an Eclipse based IDE known as Titanium Studio. Developers can create great looking native applications using JavaScript. The use and license of Appcelerator falls under Apache 2.0.

1.3.3 Adobe AIR

Adobe Integrated Runtime (Adobe AIR) uses adobe tools such as flash to allow developers to create platform independent web applications. Unlike Appcelerator this means that applications created can only be web based and not native. For this reason a lot of developers avoid using Adobe AIR. It supports all the major vendors for mobile applications but applications created in Adobe AIR are a lot slower.

1.3.4 Native Applications

A native application is a platform independent application designed to work with a particular mobile OS. The application is installed on the device whether it's a smartphone or a tablet.

Advantages

- Faster at accessing device features such as camera and accelerometer
- Easy to find in application stores such as Google Play and Apple application store
- Secure as they go through an approval phase from vendors

Disadvantages

- Expensive to develop
- Expensive to maintain
- Approval process can take from days to months
- Support of the application can be hard to maintain due to different people using different versions of operating system installed.

1.3.5 Web Applications

A web application is really a website designed to look and feel like an application. This is wrapped by the web browser, which means an Internet connection is required to use the applications. Google Chrome and Safari are examples of Web applications.

Advantages

- Easy to maintain
- Can be compatible with any device
- Approval is not required
- Easy to maintain and update without affecting the user to update the software

Disadvantages

- Limited to what can be accessed on the devices
- Hard to provide support over various Web browsers
- Not easy to find and promote, users will have to browse websites
- Can be insecure due to the application being web based

There are various platforms available for mobile devices. The most popular platforms are the as previously mentioned. Although they are in contest the use and popularity of platforms vary from region to region. An article published by Sabri (2013) states that windows phone is the most popular platform in Latin America. Although the platform was produced later than the other platforms this has become popular due to the lower prices of devices. Devices such as the iPhone and the IPad are a lot dearer and can cost a user a couple of hundred pounds. In September Forbes reported that the most popular platform on mobile devices is Android.

Germany	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	78.7	78.7	0.0
BlackBerry	0.6	0.5	-0.1
iOS	11.1	9.5	-1.6
Windows	3.8	8.8	5.0
Other	5.7	2.6	-3.1

GB	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	62.7	56.3	-6.4
BlackBerry	10.1	3.7	-6.4
iOS	21.4	27.5	6.1
Windows	4.5	12.0	7.5
Other	1.3	0.5	-0.8

France	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	62.7	63.3	0.6
BlackBerry	8.2	4.2	-4.0
iOS	13.5	17.5	4.0
Windows	5.6	10.8	5.2
Other	10.2	4.1	-6.1

Italy	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	58.6	71.6	13.0
BlackBerry	3.9	2.1	-1.8
iOS	15.0	14.4	-0.6
Windows	10.3	9.5	-0.8
Other	12.2	2.4	-9.8

Spain	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	84.9	90.8	5.9
BlackBerry	5.6	0.7	-4.9
iOS	2.8	5.3	2.5
Windows	2.1	2.2	0.1
Other	4.7	1.0	-3.7

USA	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	60.7	55.1	-5.6
BlackBerry	2.1	1.8	-0.3
iOS	33.9	39.3	5.4
Windows	2.6	3.0	0.4
Other	0.8	0.8	0.0

China	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	63.2	72.4	9.2
BlackBerry	0.2	0.0	-0.2
iOS	23.5	20.8	-2.7
Windows	4.7	2.1	-2.6
Other	8.5	4.7	-3.8

Japan	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android		48.6	
BlackBerry		0.3	
iOS		47.4	
Windows		0.8	
Other		2.9	

Australia	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	65.9	62.1	-3.8
BlackBerry	1.6	0.5	-1.1
iOS	25.8	28.7	2.9
Windows	3.7	6.5	2.8
Other	3.1	2.2	-0.9

EU5	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	68.8	70.1	1.3
BlackBerry	5.8	2.4	-3.4
iOS	14.1	16.1	2.0
Windows	5.1	9.2	4.2
Other	6.1	2.1	-4.0

Figure 1.5: Mobile phone market share September 2013

(Jones, 2013)

1.3.6 Currently available Cryptic Crossword Applications

After performing various research on the apple application store, the Google play store and BlackBerry world, it was discovered that there is not many Cryptic Crosswords available to download. There were two, which could be clearly defined as Cryptic Crossword applications, and these have been analysed in the following sections.

1.3.7 Puzzler Super Cryptic Crosswords

Platform: Apple iOS
Price: £3.99
Compatibility: iOS 4.3 or Later
Website: https://itunes.apple.com/gb/app/puzzler-super-cryptic-crosswords/id616060420?mt=8

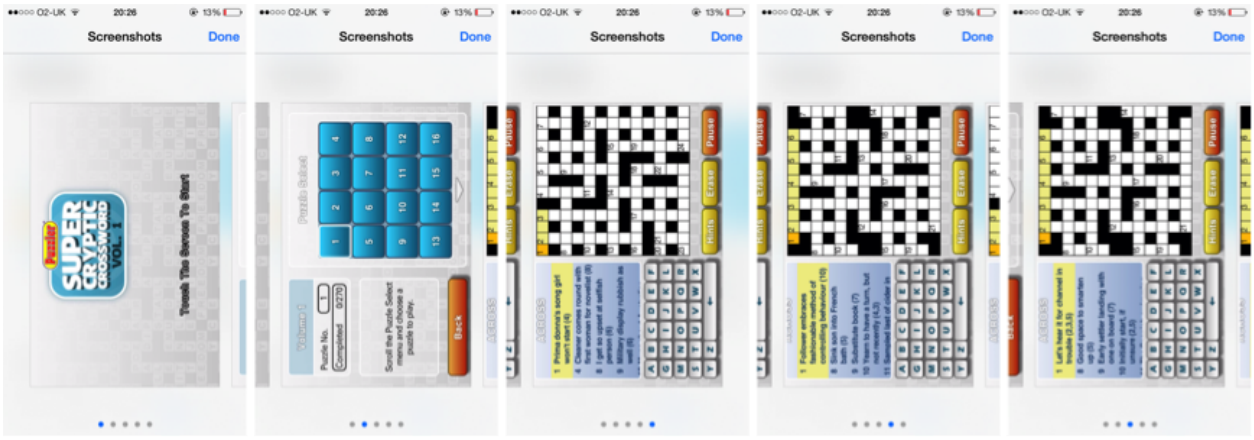

Figure 1.6: Screenshots of Puzzler Super Cryptic Crosswords on iPhone

This application contains 270 pre built crosswords. The interface is really easy to use and it clearly shows what has been completed and what needs to be completed. What's nice about this application is that the crosswords are in various levels and as you complete one crossword you can move on to the next. What was noticed is that the application already stores the answers and also the application allows hints, which makes it a little easier to use. The other noticeable thing about this application is that it didn't fully use the native features of the mobile phone. Like the keyboard is a custom keyboard.

1.3.8 Cryptic Crossword

Platform: Apple iOS, Android
Price: Free with 2 puzzles on Apple devices – In application purchase available – £1.99 on Android Devices
Compatibility: iOS 6.0 or Later
Website: https://itunes.apple.com/gb/app/cryptic-crossword/id661608021?mt=8

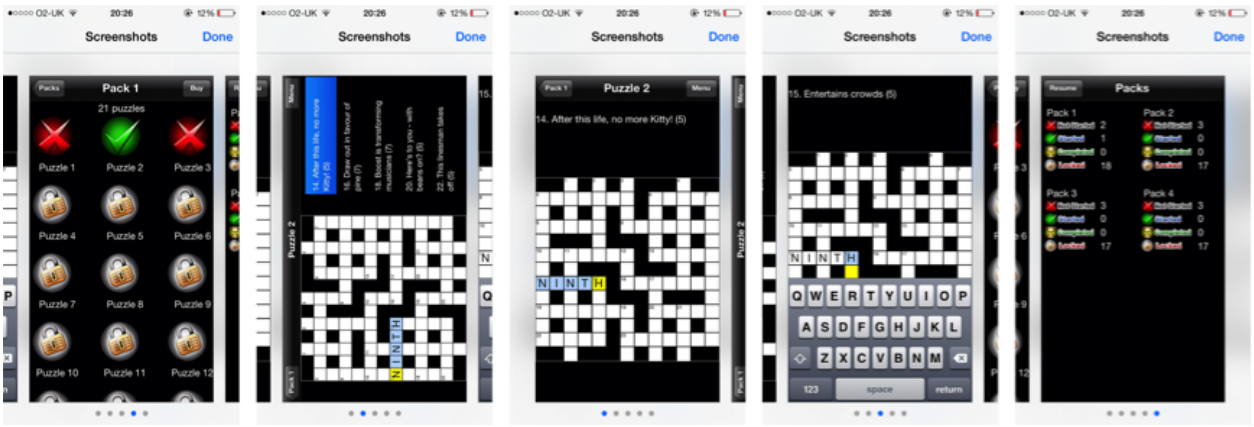

Figure 1.7: Screenshots of Cryptic Crossword on iPhone

The Cryptic Crosswords application comes with 4 different packages and a total of 81 puzzles. Each pack can be bought for £0.69 or all 4 for a discounted price of £1.99. Each package contains various cryptic crossword puzzles. After having a play around with this application at first point of contact it is noticed that look and feel of the application is not that great. The use of colours and the layout styles of the application can be better. The application consists of 81 puzzles, which can be played only after each crossword has been completed. Some of the features of the application are:

- Checking answers
- Revealing a letter, a word or the whole puzzle
- Clearing the puzzle
- Moving the character bar to next box
- Greying out completed clues

These are some of the features the application uses but there are plenty more.

1.4 Web Services

Over the past decade web services have exploded into the computing space. However the concepts that underlie web services are not new. Web services originally evolved from the Remote Procedure Call mechanism that was found in a software development framework used in the 1990s (Kalin, 2013).

During the late 1990s, XML-RPC was developed, which was a stripped down, light weight version of the Remote Procedure Call mechanism. The XML-RPC system only supported a small number of data types along with a number of simple commands. XML-RPC contained two key features, which are use of XML serialise/deserialise for data types and the reliance on HTTP for transport XML-RPC (Kalin, 2013).

XML-RPC is designed to be as lightweight as possible, and thus can be supported on a wide range of devices. XML-RPC was ultimately implemented fully and became known as SOAP. As well as SOAP, another implementation of XML-RPC occurred, which was entitled REST. Both of these technologies fall under the term 'web services'.

Since 2001 a vast range of companies have adopted the web services movement including (but not limited to) IBM, Oracle, Hewlett-Packard, Amazon, Google, Facebook and Twitter (Sullivan, 2001; Kalin, 2013).

Web services generally tend to reside upon public networks such as the Internet. However it is possible for a web service to run within a private network, such as a company's internal Intranet.

1.4.1 What are Web Services?

Although there are many companies adopting web services, the term web service has a diverse and loose definition (Kalin, 2013). During the initial explosion, many providers created heavily detailed plans upon the direction of their web service, but failed to exactly define what a web service is.

It was only until the explosion subsided that authors were able to define what a web service is (Kalin, 2013). Kalin (2013) Kalin highlighted three common characteristics between web service providers:

1. Can be thought of as a 'webified application'
2. Typically delivered over Hyper Text Transport Protocol (HTTP)
3. Typically has some form of distributed nature allowing for components to be deployed and executed across multiple devices.

For the purposes of this project a web service will be defined as:

> A service that contains one or more software components that are designed to allow machine-to-machine interaction over a network using standard protocols.

Web services follow the client-server model, which is the standard architecture for accessing a website. However unlike the traditional approaches to client/server models (such as a web server/web page setup), web services do not provide the end user with a Graphical User Interface (GUI).

The web service will provide the end user with machine readable data — i.e. the data must be put into a pre-defined GUI. This architectural design concept is not new and has been around for a number of years. Web services often can be thought of as imitating mainframes — i.e. a 'dumb terminal' sends a request to a service hosted upon a central computer system.

Web services can be broadly categorised into the distributed software systems category (Kalin, 2013). Broadly speaking a distributed software system is a system that is often split up into various components. Each component can run upon a separate physical machine, and is able to communicate with other parts of the system by passing 'messages' around. Although a web service does fit into that broad definition, there are several features that are unique to a web service.

Firstly web services heavily depend upon open, industry-standard, vendor-independent protocols such as HTTP, JSON and XML. By adding networking, data formatting and security features, web services can effectively lower start-up costs and promote interoperability between new and existing services (Kalin, 2013).

It is the interoperability that allows web services to promote language transparency. This means that web services and client programs do not need to be developed in the same

language. Many popular languages (e.g. C/C++, Java, and Python) provide inbuilt libraries or frameworks in support of web services (Kalin, 2013).

Finally web services are designed to be modular in design. This allows new services to be brought online in staggered stages, as well as allowing for laying of existing services. Again as previously mentioned each new service, can be written in the same language as the previous service, or use a completely new language (Kalin, 2013).

1.4.2 Web Service Categories

Web services can be divided into two distinct groups — SOAP based and REST-style (Kalin, 2013). Interestingly the distinction could be described as being little at most, but they are not necessarily directly compatible with each other.

SOAP

SOAP originally stood for Simple Object Access Protocol, but is often referred to as Service Oriented Architecture (SOA) Protocol (Kalin, 2013). At a glance the name change doesn't appear to be too trivial, but it is acutely an example of the technology becoming better defined (Kalin, 2013).

SOAP utilises concepts that can be seen throughout the industry, but none more so than the use of XML. One of the major advantages of XML, is that it is able to provide flexible, self-describing data structures that can easily be produced and read.

SOAP tries to imitate the postal system — i.e. allowing two machines to send and receive letters. In this analogy, the letter is the raw XML data, and the envelope is an additional data layer that wraps around the letter. The envelope adds additional information to the request, such as which operation is being requested, and may also include authentication and session information in envelope headers (Gershon, 2004).

In order to ensure one client or service can 'talk' to another service, SOAP responses must use a Web Services Definition Language (WSDL). The WSDL defines the inputs (e.g. parameters), the outputs, the operations, the protocols and the network addresses that are required and used by the service (Gershon, 2004).

The underlying implementation is loosely coupled with WSDL, which means the provider is able to change the implementation, without negatively impacting the end service users. It is the configurable services aspect that is the central concept behind all service oriented architectures (Gershon, 2004).

REST

REST stands for Representational State Transfer, and is a relatively new architecture for creating web services. Despite its relatively new architecture it is actively used by some of the larger vendors such as Google and Amazon (DOSPINESCU and PERCA, 2013).

REST relies upon the emerging architecture known as resource-oriented architecture. Essentially, these resources are a number software components that can be combined together to create reusable functionality.

As well as using a resource-oriented architecture, REST makes clever and effective use of open standard web technologies, such as the Hypertext Transfer Protocol (HTTP), the Uniform Resource Identifier (URL) and the Extensible Mark-up Language (XML) (DOSPINESCU and PERCA, 2013).

Although not all of the features have been implemented, (mostly because they are layout properties rather than data properties) the major concepts found in web technologies have been implemented and the most notable of these features are:

1. Data from the client is transmitted to the server via the URI
2. The server will perform the operation described by the HTTP method (such as GET, DELETE)
3. The URI for each resource will contain the server name and address

As previously mentioned, HTTP methods are widely used within REST. A HTTP method will describe the necessary action (Create, Read, Update and Delete — CRUD) that is required to be performed by the server (DOSPINESCU and PERCA, 2013).

The HTTP methods follow another standard in terms of the basic functions of a database management system. It must be said that REST and the HTTP protocol are mutually exclusive — REST doesn't require HTTP (DOSPINESCU and PERCA, 2013).

Table 1.6 describes common HTTP verbs and the associated CRUD operation.

HTTP verb	**CRUD operation**
POST	Create a new resource
GET	Read a resource
PUT	Update an existing resource
DELETE	Delete a given resource

Table 1.6: HTTP verbs mapped to the associated CRUD operation.

There are additional optional verbs, such as HEAD, TRACE, CONNECT, OPTIONS and INFO, but these may not be implemented by the server and/or service for security reasons. Every HTTP request will include a verb to indicate which CRUD operation should be performed upon the resource (DOSPINESCU and PERCA, 2013).

SOAP vs REST

Both REST and SOAP utilise standard protocols when communicating, and also originate from a similar specification. The real difference between the two technologies is that SOAP utilises it's own application protocol by extending current protocols.

This causes a number of issues, such as protocol standardisation. Although SOAP is based upon the HTTP protocol, each client will have to correctly understand the new extended protocol — via an additional layer of software or libraries. This adds weight to the overall technology.

SOAP describes functions, and the types of data, which requires large amount of documentation in order to use the service. As well as this there are several protocols and technologies that directly relate to it, such as Web Services Description Language, Web Servicing Addressing, XML Schema Definitions.

All binary data that is to be transmitted must be first encoded in a supported format (e.g. base64), which increases processing power at both the client and server ends. All requests are transmitted via XML, which is much slower to parse and interpret than other text-based human readable data, such as JavaScript Object Notation (JSON).

REST on the other hand is based upon uniform interfaces. This means the various clients will have a small understanding of the web service, but not necessarily how it operates or what it will return.

REST doesn't need to operate over HTTP, and doesn't contain the complexity that SOAP provides. Rather than utilising XML, REST uses the standard HTTP methods to describe what a service should do. For example obtaining a resource would use GET, and for creating a resource POST would be used.

Clients do not require additional REST supporting libraries. As long as the language supports HTTP, the client will be able to consume a REST HTTP service easily.

Unlike SOAP, REST can deliver binary data without having to encode, and responses can be formatted to either XML or the more popular JSON (due to speed increases).

1.4.3 Clients

As previously mentioned, the broader web server architecture follows the client-server application model. When designing a client-server application, a decision has to be made as to which operations (or parts of operations) should be performed upon the client and the server.

This decision is vitally important as it can affect the speed to which a system can be brought to market. It might also affect any additional extensions or updates that the system might receive in the future, as well as affecting the design flexibility.

In order to simplify the design the client will need to fall into one of the two categories — 'thin' client or a 'thick' client.

Thin

A thin client is a computer system that depends largely upon a main server, or a number of servers in order to complete any computation tasks. The client has no knowledge of how to process data, it simply knows how to pass data to another entity, and receive data from another entity.

A recent example of a thin client is Google's Chromebook. Unlike typical computers where by the applications are installed locally upon the computer, the Chromebook allows for applications to be installed within the cloud — upon an external server.

The thin client design presents a number of advantages and disadvantages. Firstly an application that is hosted upon a central server can be easily updated — as there is only one code base. Once the application has been updated, this will be pushed immediately to all thin clients.

This obviously provides an advantage in some use cases such as trying to sell goods over the Internet. For example if a product's price changes, the update will only need to be applied once to the central server, rather than having to update all clients wishing to purchase the product.

Thin clients will utilise powerful servers to do the majority of the processing. This allows for the thin clients to be less powerful, and hence the overall costing to reduce.

However this will mean that thin clients will have poorer response times. The main reason for this is the fact that the majority of the operations are being complete upon another machine (potentially many miles away). Simple operations such as populating a menu, might require a request to the main server, thus increasing the overall time to achieve something.

Resources within a thin client network will need to be managed more effectively. Thin clients will use more bandwidth upon the network, and will make more connections to the server. This would require the server to be able to handle lots of potentially fast and slow connections, with each connection using a wide range of internal server resources (CPU, Memory etc).

Thick

A thick client is a computer system that has little dependency upon a main server, or a number of servers in order to complete computational tasks. The client will still require a limited connection to a server, but will not use the connection as often in comparison to a

thin client. A thick click will often be able to perform many operations without a connection to a network.

An example of a thick client would be a standard desktop installation. The desktop installation might provide various pieces of software that are installed locally upon the computer. For example the computer would be able to produce various documents regardless of the state of the network connection.

The thick client design presents a number of advantages and disadvantages. Firstly due to the fact that clients are able to do more of the computational work, server specifications do not need to be as high. This allows for cheaper servers to be purchased, and few overheads in terms of running and maintenance costs.

This will also lead to an increase in server capacity, again due to the fact that the client is carrying out more work. This ultimately means that the server is required to do less work, and can hence support a larger number of users.

Thick clients have an increased advantage over thin clients in terms of network connectivity. Thick clients do not require a constant connection to a server. This in turn frees up bandwidth that is being used upon the network, as well as reducing server loads.

Finally the end user is able to store files and applications locally upon the machine. This in turn allows for a faster application start up time, and a reduced file access time. Hence increasing the speed of operations, as well as reducing bandwidth upon the network.

However thick clients are more expensive to purchase, deploy and maintain. The reason being is that there will be more computers with higher specifications. This can lead to more expensive repair bills, should systems fail.

Fixing and troubleshooting become more difficult, simply because there are more machines to troubleshoot and fix should problems occur. This is obviously not a problem if there was a central server, such as found within the thin client model.

Glossary of Terms

The following section contains a glossary with the meanings of all names, acronyms, and abbreviations used by the stakeholders.

Term/Acronym	**Definition**
The Guardian	A national UK newspaper that prints daily cryptic crosswords
Android	A mobile phone software platform by Google Inc.
BlackBerry	A mobile phone hardware and software platform developed by BlackBerry Limited
iOS	A mobile phone software platform developed by Apple Inc.
iPhone	A smart phone developed by Apple Inc.
iPod	A portable digital music player developed by Apple Inc.
iPad	A tablet developed by Apple Inc.
NLP	Natural Language Processing
SRS	Software Requirements Specification
App	Shorthand for application

Bibliography

Akman, O. (2013). The smartphone revolution. Retrieved from http://mail2web.com/blog/2011/05/smartphone-revolution-growth-smartphones-exchange-activesync/.

Apache (2013). Apache opennlp developer documentation. Retrieved from http://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html#tools.sentdetect.

Apple (2007). Macworld san francisco 2007 keynote address. Retrieved from https://itunes.apple.com/gb/podcast/apple-keynotes/id275834665.

Apple (2013). App store tops 40 billion downloads with almost half in 2012. Retrieved from http://www.apple.com/pr/library/2013/01/07App-Store-Tops-40-Billion-Downloads-with-Almost-Half-in-2012.html.

Bird, S. (2009). *Natural Language Processing with Python*. O'Reilley Media, Inc.

Businessline (2012). Microsoft to release windows 8 on oct 26. Retrieved from http://search.proquest.com/docview/1321630361?accountid=11526.

Cha, B. (2009). Rim store crowned blackberry app world. Retrieved from http://news.cnet.com/8301-17938_105-10188400-1.html?tag=mncol;txt.

Collins, J. (2004). Languages, natural and formal. Retrieved from http://www.uea.ac.uk/~j108/languages.htm.

Connor, A. (2012a). Cryptic crosswords for beginners: palindromes. Retrieved from http://www.theguardian.com/crosswords/crossword-blog/2012/nov/01/cryptic-crosswords-beginners-palindromes.

Connor, A. (2012b). Cryptic crosswords for beginners: spoonerisms. Retrieved from http://www.theguardian.com/crosswords/crossword-blog/2012/mar/01/cryptic-crosswords-beginners-spoonerisms.

Dictionaries, O. (2011). *Concise Oxford English Dictionary*. Oxford University Press, twelth edition.

DOSPINESCU, O. and PERCA, M. (2013). Web services in mobile applications. *Informatica Economica*, 17(2):17 – 26.

Gershon, G. (2004). Web services. *e-Doc Magazine*.

Gordius (2003). Cryptic crossword no 22,717. Retrieved from `http://www.theguardian.com/crosswords/cryptic/22717`.

Halpern, J. (2013). Cryptic crossword no 26,091. Retrieved from `http://www.theguardian.com/crosswords/cryptic/26091`.

IDC (2013). Android and ios combine for 91.1% of the worldwide smartphone os market in 4q12 and 87.6% for the year, according to idc. Retrieved from `http://www.idc.com/getdoc.jsp?containerId=prUS23946013`.

Jiang, T., Li, M., Ravikumar, B., and Regan, K. W. (2010). *Algorithms and theory of computation handbook*. Chapman & Hall.

Jones, C. (2013). Idc: Windows phone is the second most popular mobile platform in latin america. Retrieved from `http://www.forbes.com/sites/chuckjones/2013/09/30/iphones-market-share-down-prior-to-5s-5c-launch-with-windows-almost-double-digits-in-europe/`.

Kalin, M. (2013). *Java Web Services: Up and Running*. O'Reilly Media, second edition.

Litman, D. (2003). Cs 1573: Artificial intelligence application development (spring 2003). Retrieved from `http://people.cs.pitt.edu/~litman/courses/cs1573s03/lec/chunking.ppt`.

McCluskey, T. L. (1999). Cha 2555 - artificial intelligence. Retrieved from `http://helios.hud.ac.uk/scomtlm/cha2555/lecture13_18.ppt`.

NLTK Project (2012). Nltk 2.0 documentation. Retrieved from `http://nltk.org/data.html`.

Sabri, S. (2013). Idc: Windows phone is the second most popular mobile platform in latin america. Retrieved from `http://www.wpcentral.com/windows-phone-second-most-popular-mobile-platform-latin-america`.

Sullivan, T. (2001). Web services. *InfoWorld*, 23(11):38.

Upadhyay, S. (2008a). Acrostics. Retrieved from `http://www.crosswordunclued.com/2008/09/first-and-last-letters.html`.

Upadhyay, S. (2008b). Charades. Retrieved from `http://www.crosswordunclued.com/2008/11/charades.html`.

Upadhyay, S. (2008c). Cryptic definitions. Retrieved from http://www.crosswordunclued.com/2008/12/cryptic-definitions.html.

Upadhyay, S. (2008d). Decoding double definitions. Retrieved from http://www.crosswordunclued.com/2008/10/decoding-double-definitions.html.

Upadhyay, S. (2008e). Digging out hidden words. Retrieved from http://www.crosswordunclued.com/2008/08/digging-out-hidden-words.html.

Upadhyay, S. (2008f). How to spot anagrams. Retrieved from http://www.crosswordunclued.com/2008/08/how-to-spot-anagram.html.

Upadhyay, S. (2008g). &lit, literally so. Retrieved from http://www.crosswordunclued.com/2008/08/and-literally-so.html.

Upadhyay, S. (2008h). Reversals. Retrieved from http://www.crosswordunclued.com/2008/11/reversals.html.

Upadhyay, S. (2008i). Substitutions simplified. Retrieved from http://www.crosswordunclued.com/2008/12/substitutions.html.

Upadhyay, S. (2008j). Tune in to homophones. Retrieved from http://www.crosswordunclued.com/2008/10/homophones.html.

Upadhyay, S. (2009a). Containers. Retrieved from http://www.crosswordunclued.com/2009/02/containers.html.

Upadhyay, S. (2009b). Deletions. Retrieved from http://www.crosswordunclued.com/2009/03/deletions.html.

Upadhyay, S. (2009c). Letter exchange. Retrieved from http://www.crosswordunclued.com/2009/12/letter-exchange.html.

Upadhyay, S. (2009d). Letter picking. Retrieved from http://www.crosswordunclued.com/2009/04/letter-sequences.html.

Upadhyay, S. (2009e). Letter shifting. Retrieved from http://www.crosswordunclued.com/2009/12/letter-shifting.html.