FYP Final Report

Monitoring User Preferences - An Application to Desktop Environments

Supervisor: Dr Jesús García Galán

Student: Michael Hallinan - 12134635

Table of Contents

|  |  |
| --- | --- |
| Section Title | Page |
| Table of Contents | *1* |
| [Acknowledgements](#_Project_Summary) | *2* |
| 1. Introduction and Objectives | *3* |
| 1.1 Introduction |  |
| 1.2 Motivation |  |
| 1.3 Objectives of Project |  |
| 2. Research |  |
| 2.1 Introduction |  |
| 2.2 Papers |  |
| 2.3 Existing Projects |  |
| 2.4 Other Sources |  |
| 3. DESIGN |  |
| 3.1 Introduction |  |
|  |  |
|  |  |
| 4. IMPLEMENTATION |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 5. SYSTEM TESTING AND EVALUATION |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 6. CONCLUSIONS AND RECOMMENDATIONS |  |
|  |  |
|  |  |
|  |  |

1: Introduction and Objectives

# 1.1 Introduction

Monitoring User Preferences - An Application to Desktop Environments

*A critical requirement of software systems is to satisfy the needs of the user.*

The focus of this project will be on desktop environments and will be aimed to identify strategies to capture user preferences and detect changes. This is intended to be done explicitly by requesting user feedback, as well as implicitly through mining of user behaviour or system logs.

Proper user preference specification is believed to be very important for maximum user enjoyment when using software. This project will focus upon acquiring these specifications through monitoring the user and to apply this information to create adaptive software tailored to the individual user.

Many users are not fully aware of the configuration capabilities of a system. Advanced configuration options can also intimidate novice users of a system. This project will seek to present a simple configuration but allow for more complicated configuration through its use of user-profiling. The chosen solution will be a “Web Assistant” Extension to the Google Chrome web browser that will monitor the users browsing habits and provide recommendations to the user based upon this.

The enormous growth in web content and software applications as well as the increased popularity of these products has led to increased demands for user accessibility. In recent years, there has been much effort put into developing user-adaptive web-sites and applications (Perkowitz & Etzoni, 2000).

Though many techniques exist, a commonly used method is based upon monitoring User Preferences. User preferences are frequently assessed in order to improve the user experience.

### 1.1.1 User Preferences

A user is described as being “any person who uses or operates something”. (Oxford Dictionary, 2016) User Preferences can be described as “Specifications or settings for features of a software program as defined by the user.”

Another term frequently used for the type of study is “User Profiling”. A user profile has been defined as the description of the user interests, characteristics, behaviours, and preferences. User profiling is the practice of gathering, organizing, and interpreting the user profile information. (Godoy & Amandi, 2005)

These are frequently options selected by users in order to simplify or enhance their experience while using a software program. Explicitly defined examples include simple alterations such as colour, font, or language as well as more advanced modifications such as key-binding or behavioural changes.

A common example of implicit information gathering would be the use of link personalisation to recommend items based on buying history or ratings by e-commerce applications. This is detailed further in (Rossi, et al., 2001)

### 1.1.2 Proposed Solution

This project is intended to study and monitor these preferences in order to allow for the optimal experience for users of all levels. This will be accomplished through the gathering of information from the user and modification of the desktop environment accordingly. This will be done explicitly through user feedback such as user-defined settings or implicitly through data logging and mining.

Many existing studies and papers have already dealt with similar topics and these will be detailed later in the **Research** section of the report. This project is intended to focus some of these studies into a single project.

Though numerous ideas and possible proposals were considered, the approach selected for this project was a scheduling and recommendation system for a web browser. The main part of the extension is the scheduler, which recommends to the user websites at certain times based upon their past browsing history. The second part of the extension is a recommender system that will show the user similar sites to those they are viewing. The final part of the extension is a profiling system that will create “snapshots” of the user’s patterns in order to properly complete a User Profile of the user and further recognise patterns in their browsing.

The goals for the planned extension are for it to contain various methods of retrieving data from the user through feedback such as buttons and option pages, as well as monitoring the browsing patterns of the user in order to further present recommendations and alter the interface accordingly.

This solution is intended to support the User’s web experience through the recommendation of both new sites and schedule entries. The scheduler will remind the User of certain websites at the appropriate times with a popup notification, and facilitate them by allowing the User to navigate to that page if the notification is clicked. This should allow for a more fluid and enjoyable experience from the user.

# 1.2 Motivating Factors

When I set out to select a subject for my Final Year Project, I wanted to pick something that was relevant but was also something that I could be interested in. Initial ideas for a self-submitted project entailed a voice activated user-access program similar to the existing digital assistants Siri (Apple Inc., 2011) and Cortana (Microsoft, 2015). This idea was grown from my interests in Machine Learning and Artificial Intelligence, as well as my interest in language parsing and syntactic analysis. Another reason for the decision to create a digital assistant was from my experiences on Co-op, where I worked as IT Help Desk in a bank in Luxembourg. While working, I saw the problems that many users had when using their computers. Many of these problems could be quite simple, but novice computer users would be intimidated. I had hoped to create an interface that could solve these problems for the users as they arose. This would have uses of increasing productivity in numerous organisations.

Though I had hoped to base my project upon general user interface assistance, upon beginning research into that initial project I began to realise the sheer magnitude of the task. I decided to focus the scope of the project into a more concise area because of this. One of the Final Year Project Proposals was based upon User Preferences, and after meeting with the project supervisor, and after further research into the subject, I decided that this project was very similar to my initial goals and felt that the experience of the supervisor would be a huge benefit. (REPHRASE?)

I began initially by reading papers by (Iglesias, et al., 2012) and (Holland, et al., 2003) that were provided by my supervisor. These papers mentioned the use of Machine Learning and heuristic algorithms in determining the user preferences, as well as the methods for collecting user information. This helped to give me an idea of where to begin my project. Through the use of Google Scholar, I was able to find more papers that dealt with these topics, and form ideas of how best to propose a solution to the project. Though the initial research and design was quite slow, once I began implementing my idea everything quickly began to come together.

# 1.3 Objectives of the Project

The objective of this project was to create a program capable of adapting to the user’s requirements. This was to be done through the monitoring of the user’s habits and preferences. These habits were to be processed and used to create an interface more tailored to the user. Extensive testing was to be done in order to ensure that the proposed solution would successfully tailor to the user’s needs as well as be readily and easily usable by everybody.

Prior to commencing the solution for this project, I felt it necessary to examine existing research as well as any similar projects or other documentation in this area. This existing research has been presented in **Chapter 2** along with a summary of the initial findings, while the efforts completed for the design of the implementation have been described in **Chapter 3**. Thedetails for the selected implementation have been given in **Chapter 4**, with the description and evaluation of testing conducted presented in **Chapter 5**.Finally the conclusion and recommendations for further projects are shown in **Chapter 6**.

2: Research

# 2.1 Introduction

An important part of the project was the research and reasoning behind any decisions made. Numerous studies and papers exist in similar fields and these were consulted during the initial stages of the project. A number of papers were supplied by the supervisor and through the use of Google Scholar, more papers were found. These papers were viewed using the University of Limerick Library’s services, and the books used for studying were also borrowed from the library.

A large portion of the first semester was spent researching the topic and forming possible ideas for the project. Below you will find the research papers that were read as well as any similar existing projects reviewed. You will also find the books and other sources that were consulted as part of the research.

# 2. 2 Research Papers

The initial papers consulted were *Creating Evolving User Behavior Profiles Automatically* (Iglesias, et al., 2012) and *Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications* (Holland, et al., 2003), among others. They detailed self-learning systems and were the first resources consulted upon initiation of the project. These papers were useful in presenting proper recording technique as well as showing previous studies in order to help form an idea for a possible solution for the project. The reports described the processes completed and their methods of creating a user profile from the user’s data.

References from these papers were consulted for further reading material. Google Scholar was also consulted to find more recent papers referencing these sources. These efforts led me to *Detecting innovative topics based on user-interest ontology* (Nakatsuji, et al., 2009) and *User Preference Mining Techniques for Personalized Applications* (Holland & Kießling, 2004)*.*

These papers spoke further about the processing of the information, such as mining algorithms and applications of the information. The subjects of both papers were based upon web-based applications which helped to cement the decision to work on a web-based application for the project. Nakatsuji, et al. based their report on recognising user-interest. This topic choice was major influence in where to base the proposed solution for this project, and led to the decision to implement the Recommendations system in the project.

The papers spoke at length about the algorithms used and processing methods but were brief with the description of the methods of information gathering that were used. This helped to decide how to process the information but it was necessary to look elsewhere in order to find other possible methods of gathering the information for processing.

Papers were researched dealing with search engine queries such as (Silverstein, et al., 1999) which dealt with the analysis of the search requests made to the AltaVista Search engine over a period of six weeks. This documented the analysis of the queries and detailed the session information and correlation between them. The paper researched most similar to the project solution was about a system incorporated into Internet Explorer that would profile a user’s interest based upon their recent browsing history (Grčar, et al., 2005). This system would perform clustering algorithms upon word-vectors formed from the text content of pages and identify the user’s interests. This was very similar to the eventual project solution and helped to influence the final decision.

# 2.3 Existing Projects

### 2.3.1 Google Now

The most prevalent example found of proper monitoring of User Preferences is Google Now by Google. This application gathers information from users’ search history in order to display possibly useful material related to fields the user may be interested in. The application can then gather explicit information from the user such as feedback to this material or selection of certain fields such as “Home” location. This clearly displays both implicit (material) and explicit (feedback) information gathering. It also processes this information to display information relative to the user, as is intended by our application.

The application may also gather location information using Wi-Fi or GPS. This information may be used to automatically determine a “Home” location and a “Work” location, along with promoting local news sources. These locations will then be used to alert the user to weather and traffic conditions around these locations. This information is processed primarily by recording location versus time, with night time locations being evaluated higher as being home locations. Daytime locations, particularly weekdays, would also be evaluated more highly as possible work locations. Another capability of the application is contextual assistance such as displaying local attractions if the user is in a location far from their “Home” location.

Research has been presented recently investigating the abilities of Google Now as a Personal Assistant (Guha, et al., 2015). The research details the personalisation functionalities of the application as well as the contextual assistance provided. This paper detailed the current algorithms and functionalities of Google Now, such as context identification and task assistance that are currently in use, as well as the shortfalls and possible growth areas for the application.

### 2.3.2 Flix Plus by Lifehacker

Originally entertaining the idea of a Netflix extension for Chrome, the many existing products were reviewed. Though these applications were less focused upon adaptive interfaces and information gathering, they did perform actions similar to those desired by the project application. A primary example of a similar extension was *“Flix Plus”* by Life Hacker. This extension performed many of the originally conceived ideas for the Netflix implementation. These included interface overlays and improved customisation options for the interface such as hiding elements or highlighting certain aspects. However, this extension used only explicit user information gathering and failed to display any signs of an automatically adapting interface or user profiling. Most of the features contained were limited to improved user settings and primarily covered visually improving the user interface. Examples include blurring of episode information and dynamic removal of show categories.

### 2.3.3 Google Analytics

Qwdqw

Qwe

Qe

qweqwe

### 2.3.4 Proactive Assistant

Proactive Assistant is a recently released feature created by Apple Inc. in order to work along with many of their other existing applications such as Siri, the voice-activated assistant, found on many Apple devices. This system is designed to utilise the information gathered from the user’s data stored on the device, such as use of Siri, in order to form a user profile. This application was designed to be a competitor to Google Now, which is described above in section 2.3.1.

Proactive Assistant utilises the user’s profile in order to assist the user similar to those functions performed by Google Now. It will avail of other Apple services such as Wallet, an application to store coupons, tickets, and boarding passes, in order to tailor the service more closely to the user’s needs. The program will also recognise the patterns of use for certain applications and will provide a button to that application on the Proactive Assistant screen. Other examples are notifications for nearby restaurants at times such as lunch and dinner and other relevant locations, similar to those functionalities performed by Google Now. (Lardinois, 2015)

### 2.3.5 Personal Smart Assistant for Digital Media and Advertisement

*Personal Smart Assistant for Digital Media and Advertisement* (Hussain, 2013) was a thesis found online from the University of Western Ontario. The thesis dealt with a subject similar to that of my project and had performed research of its own into related fields. The thesis detailed a generic personal smart assistant that provides relevant assistance to the user based upon a profile developed through modelling the user’s interests and behaviours. This report also proposed a service whereby the assistant would recommend an advertisement to the user. This idea was partially utilised when designing possible methods of commercialisation, as detailed in section 3.2.2 of this report.

The thesis spoke at length about the proposed concept of the Personal Smart Assistant and the methods used in designing the software. This detailed a User Model composed of a Behaviour Model, consisting of deliberative (Goal-driven) and reactive (Events-triggered) behaviour, which would capture the user’s interests into an Interests Model. Although the thesis dealt with a very similar topic to this final year project, it was not discovered until the later stages of the implementation, and had only a minimal impact upon the final product.

### 2.3.6 Summary of Existing Products

Will I do this?

# 2.4 Other Sources

Other than existing Academic Papers, a number of books were consulted about datamining and machine learning, as well as human-computer interaction. Many of the papers had mentioned the use of datamining for analysing the collected data, and further research was done into the possible techniques that could be used by the project.

The first book read was *Datamining: Practical Learning Tools and Techniques* (Witten, et al., 2011). This book detailed many of the popular datamining techniques and algorithms. It also contained numerous examples of the uses of these mining techniques in order to further explain when they are used. This was particularly useful for understanding these techniques before delving deeper into their uses. These explanations and ideals were greatly beneficial when deciding how to process the information that would be gathered.

After initialstudy of Datamining, *Machine Learning: A Probabilistic Perspective* (Murphy, 2012) was consulted in order to further research possible methods of gathering meaningful and useful information from the collected data. Many of the papers mentioned using clustering and other machine learning techniques in order to mine the data and this book was used to further research these techniques. This was important for full understanding of the techniques as well as selection of the processes used. This helped to improve upon the explanations of the procedures detailed in the previous book. Although the processing system used was different to those detailed in the book, many of the explanations were vital for creating the eventual processing algorithm.

As the interface and user-interactability was a vital part of the project design, a number of user-interface books and guides were consulted. *Don’t Make Me Think: A Common Sense Approach to Web and Mobile Usability* (Krug, 2014) was a very short but enlightening book revolving around the importance of simplicity in design. The book itself was made to be short in order to exemplify good design and simplicity. Much of the book detailed around creating interfaces that are self-explanatory and allow for quick navigation and require little “thinking” in order to perform an action. This thought process was carefully preserved when designing the interface of the project.

*The Design of Everyday Things* (Norman, 2013) was also selected in order to help understand the thought process behind many designs. This book delved into the psychology of everyday actions as well as understanding faults of bad design and how “Human Error” is often attributed to cases of faulty design. In this book, Norman stressed the importance of intuitive understanding and being able to use anything with as little instruction as possible. This book was important for understanding common pitfalls and constraints for the design of the program as well as techniques for the testing of the program interface.

Much of the ideology and information discussed within the books was utilised when designing the extension. The usability and design principles

3. Design

# 3.1 Introduction

The design of the extension was of high importance. Much of the extension focused on appealing to the user and of being simple and straightforward. This meant that much time was spent designing the interface as was spent designing the software itself. Usability and Learnability were important characteristics considered when creating the interface. One of the vital necessities of the project was to be simple to use and unintimidating to new users. Further efforts were put into extensibility and maintainability, for adding and modifying the features of the software. Efficiency and reliability were also crucial for user acceptance of the intended solution. Security was another concern in design, and necessary measures needed to be put in place.

In the following sections you will find details of the design process for creating the extension. This will detail much of the thought process behind many of the decisions as well as the alternative ideas that were considered.

# 3.2 System Overview

### 3.2.1 Project Features

Wwgwgwgg

### 3.2.2 Capturing User Preferences

### 3.2.3 Processing User Information

### 3.2.4 Commercial Viability

3.4.1 Anonymous Profile Information

Sell for use in data mining etc

3.4.2 Targeted Advertising

Have an example with extension connecting with website and correctly targeting ads. Google Analytics.

# 3.3 Software

### 3.3.1 Environment

From the initial stages the idea of a Web Browser Extension was decided. With the intention of allowing the extension to be as widely used as possible, the most popular web browser needed to be determined. Research into the usage of web browsers determined that the most commonly used web browser was Google Chrome, followed by Microsoft’s Internet Explorer and Mozilla Firefox. (StatCounter Global Stats, 2016). This measured the traffic of over 3 million websites and recorded data based upon page views. Another source ranked Internet Explorer as the most commonly used, followed by Google Chrome (NetMarketShare, 2016), determined by unique daily users. This method also weighted records by average country traffic, meaning the recorded traffic from one country would be weighted to relate more closely to national internet traffic for that country.

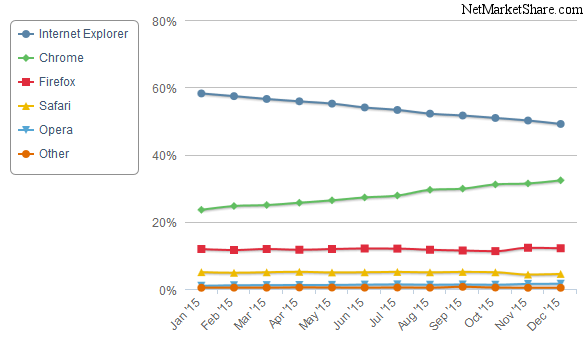


Fig. 2.? - Diagram of Browser usage from NetMarketShare.com for the year 2015

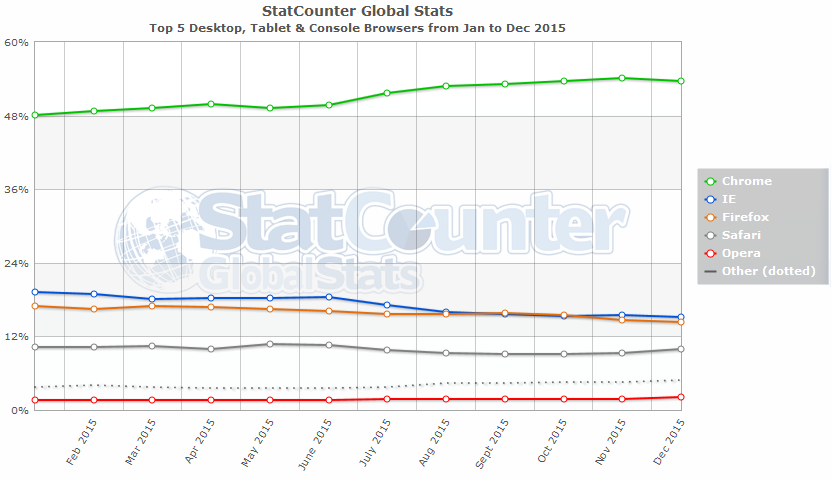


Fig. 2.? - Diagram of Browser usage from gs.StatCounter.com for the year 2015

The information above was assessed and Google Chrome was decided upon for creating the extension. The use of Google Chrome had numerous other benefits, such as greater portability across different operating systems and a more simplified approach to installing extensions. The release of Windows 10 with the new Edge web browser also meant a more limited life-span for Internet Explorer. The browser can also be seen to be already declining in both of the graphs above. The Google Store would also be beneficial in the distribution of the extension for testing.

### 3.3.2 Languages

After the selection of the project type and environment, the means of creating the application were determined. Google Chrome Extensions were primarily written in HTML, CSS, and JavaScript. Although HTML and CSS had already been briefly covered in previous modules throughout the course, JavaScript had not. This required the learning of JavaScript in order to create the extension.

Numerous resources were consulted in the early stages of development such as *JavaScript and JQuery: Interactive Front-End Web Development* (Duckett, et al., 2014) and online resources were consulted in order to properly learn the API of both JavaScript, and Google Chrome. Though the requirement of learning a new programming language slowed the initial stages of the project, progression improved throughout the course of the implementation.

Initial design stages involved a separate Java web server that would process the User’s information. This would have utilised the open-source WEKA plugin (WEKA, 2015) to process the user information. This server would have communicated with the extension and allowed for accessing other online resources. This was eventually rejected due to possible issues with connectivity, and was deemed overly complicated. Instead it was decided to process the information using JavaScript.

### 3.3.3 Storage

Storing the information used by the extension was greatly facilitated through the use of the Google Chrome API. Though other storage methods were possible through HTML5, the Chrome API allowed for further options for storage such as synchronised storage. Synchronised storage allowed for information to be stored on the User’s profile, to allow settings and other information to be accessed across multiple devices. This was a major benefit of using Google Chrome for my application. All storage access was completed using asynchronous functions. This meant that much of the functionality needed to accommodate for this. This will be detailed further in the Problems Faced section in Chapter 5 Testing and Evaluation of this report.

For storing many of the lists and static components in the extension, JSON was used. JSON stands for JavaScript Object Notation, and is a “lightweight data-interchange format” based upon a subset of *JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999* (Standard, ECMA, 2011)*.* The formatis similar to that of JavaScript Objects, and this means that the format is easy to read and write for both humans and computers. Information could be stored as an array or object, and could be easily accessed through the extension.

### 3.3.4 System Architecture

An important part of the design was usability. A large factor with usability is the efficiency of the program. This is because users will often abandon a service that has overly lengthy loading times. Much of the design of the extension incorporated many factors such as binary searches and proper utilisation of asynchronous functions. Asynchronous functions are functions that run on another thread and this allows them to run concurrently, or in parallel to other functions. This meant that functions can be run in the background and allows for faster loading of important items. Along with proper search and sorting techniques this ensured that the program would load all necessary information in as short a time as possible.

Another key ideal when designing the software was that of maintainability and extensibility. The code was designed to be easy to read and understand, as well as reducing the amount of “hard coded” values and code duplication. This meant that extending the project to include additional features, or perform maintenance such as updates or bug fixes would be much easier, even if performed by somebody new to working on the project.

# 3.3 Interface

### 3.3.1 Usability

As mentioned above, the usability of the extension was a major focus of the extension. Much of this usability relied on the User Interface. As discovered in the research, many well-designed interfaces rely on intuitive understanding and minimal effort. With this in mind the interface was designed with a minimalistic approach. Buttons were made large and of a different colour. The background was made white so as to provide contrast to and highlight the coloured features.

### 3.3.2 Visibility/Readability (Look up Quality Terms)

Blah

### 3.3.3 Understandability

How easy the software was to understand

4. Implementation

# 4.1 Introduction

The implementation of the KNOWLEDGE learned through the research the previous semester was…

# 4.2 Scheduler

Blsa

### 4.2.1 Processing Algorithm

rgrg

# 4.3 Recommender

Wgwgwgwg

# 4.4 Profiler?? Options?

Wgwgwgwgwg

5. Testing and Evaluation

# 5.1 Problems Faced

Oh so many...

### 5.1.1 JavaScript Language

WORK ON THIS NOW THAT YOU’VE FINISHED

In order to create the extension for Google Chrome, it was required to use JavaScript for a large section of the functionality. This required learning JavaScript without any prior experience with the language. This was initially performed primarily online, intending to avail of the many free services available online.

JavaScript is an interpreted, dynamic, and untyped programming language based off of ECMAScript (Standard, ECMA, 2011)

Learning JavaScript proved to be quite challenging as it was different to many of the programming languages taught during the course of the degree, such as Java, C, and C++. JavaScript is dynamically typed and interpreted; this means that … which made certain familiar processes more complicated. Though the syntax for the language is derived from C, the semantics and design are influenced by other languages. The resources available for JavaScript

This meant that large a large amount of time was spent analysing tutorials and guides that dealt with already familiar subjects. The book *JavaScript and JQuery: Interactive Front-End Web Development* (Duckett, et al., 2014) was also consulted later in the semester due to the number of problems faced learning the language online. These problems learning the language meant significant delays for the implementation of the application.

### 5.1.? Netflix API

Initial plans for the extension were for it to work along with the Netflix streaming website. After completion of research and the commencement of the implementation of the extension, it was realised that the API for Netflix was no longer publicly available. This caused obvious issues with interfacing between the extension and the web application. Of the existing API that can be found online, a vast majority of it is outdated due to frequent updates by Netflix.

Due to the aforementioned issues with the availability of the API, the project was changed. The initial plan of an extension for Google Chrome was maintained, but was altered in order to work under a different medium while maintaining the ideals and intentions of the project. Thankfully much of the work completed thus far had been research or general extension functionality and allowed for a much more painless transition.

### Time Processing and Daylight Savings

Oh what a bother that was…

### Miscellaneous

Sgsgsgsgs

# 5.2 Project Testing

# 5.3 Evaluation?

6. Conclusions and Recommendations

# 6.1 Introduction

The following sections will outline the conclusions gathered from the project as well as recommendations for further or similar work.

# Citations and Bibliography

Barakova, E., Mestrom, R. & Willemsen, W., 2011. The Application of Learning Algorithms in the Development of Natural Interaction. *Procedings of the Second Conference on Creativity and Innovation in Design,* pp. 81-84.

Duckett, J., Ruppert, G. & Moore, J., 2014. *JavaScript and JQuery: Interactive Front-End Web Development.* First ed. s.l.:Wiley.

Godoy, D. & Amandi, A., 2005. User Profiling in Personal Information Agents: A Survey. *The Knowledge Engineering Review,* 20(4), pp. 329-361.

Google, 2016. *JavaScript APIs - Google Chrome.* [Online]   
Available at: https://developer.chrome.com/extensions/api\_index  
[Accessed 7 January 2016].

Grčar, M., Mladenič, D. & Grobelnik, M., 2005. User Profiling For Interest-Focused Browsing History. *Proceedings of the Workshop on End User Aspects of the Semantic Web,* pp. 99-109.

Guha, R., Gupta, V., Raghunathan, V. & Srikant, R., 2015. User modeling for a personal assistant. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining,* pp. 275-284. ACM.

Holland, S., Ester, M. & Kießling, W., 2003. Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications. *Springer Berlin Heidelberg,* pp. 204-216.

Holland, S. & Kießling, W., 2004. User Preference Mining Techniques for Personalized Applications. *Wirtschaftsinformatik,* 46(6), pp. 439-445.

Hussain, A., 2013. *Personal Smart Assistant for Digital Media and Advertisement (Doctoral dissertation, The University of Western Ontario).* [Online]   
Available at: http://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=2371&context=etd  
[Accessed 10 April 2016].

Iglesias, J. A., Angelov, P., Ledezma, A. & Sanchis, A., 2012. Creating Evolving User Behavior Profiles Automatically. *Knowledge and Data Engineering, IEEE Transactions on,* 24(5), pp. 854-867.

Krug, S., 2014. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability.* Third ed. s.l.:New Riders.

Lardinois, F., 2015. *Apple Announces Proactive Assistant, Its Google Now Competitor| Tech Crunch.* [Online]   
Available at: http://techcrunch.com/2015/06/08/apple-announces-proactive-assistant-its-google-now-competitor/  
[Accessed 12 January 2016].

Mozilla Foundation, 2016. *JavaScript | MDN.* [Online]   
Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript  
[Accessed 16 March 2016].

Murphy, K. P., 2012. *Machine Learning: A Probabalistic Perspective.* First ed. s.l.:The MIT Press.

Nakatsuji, M., Yoshida, M. & Ishida, T., 2009. Detecting innovative topics based on user-interest ontology. *Web Semantics: Science, Services and Agents on the World Wide Web,* 7(2), pp. 107-120.

NetMarketShare, 2016. *Browser Market Share.* [Online]   
Available at: https://www.netmarketshare.com/browser-market-share.aspx?qprid=2&qpcustomd=0  
[Accessed 20 March 2016].

Norman, D., 2013. *The Design of Everyday Things: Revised and Expanded Edition.* Revised ed. s.l.:Basic Books.

Oxford Dictionary, 2016. *User - Definition of 'user' from the Oxford Dictionary.* [Online]   
Available at: http://www.oxforddictionaries.com/definition/english/user  
[Accessed 10 April 2016].

Perkowitz, M. & Etzoni, O., 2000. Towards adaptive Web sites: Conceptual framework and case study. *Artificial Intelligence,* 118(1), pp. 245-275.

Rossi, G., Schwabe, D. & Guimaraes, R., 2001. Designing Personalized Web Applications. *Proceedings of the 10th international conference on World Wide Web,* Volume pp. 275-284. ACM.

Roush, W., 2010. *The Story of Siri, from Birth at SRI to Acquisition by Apple—Virtual Personal Assistants Go Mobile.* [Online]   
Available at: http://www.xconomy.com/san-francisco/2010/06/14/the-story-of-siri-from-birth-at-sri-to-acquisition-by-apple-virtual-personal-assistants-go-mobile/#  
[Accessed 9 April 2016].

Silverstein, C., Marais, H., Henzinger, M. & Moricz, M., 1999. Analysis of a very large web search engine query log. *ACM SIGIR Forum,* 33(1), pp. 6-12.

Sipper, M. et al., 1997. A Phylogenetic, Ontogenetic, and Epigenetic View of Bio-Inspired Hardware Systems. *Evolutionary Computation, IEEE Transactions on,* 1(1), pp. 83-97.

Standard, E. C. M. A., 2011. 262: ECMAScript Language Specification, December 1999. *Web link: http://www. ecma-international. org/publications/standards/Ecma-262. htm (2011).*

StatCounter Global Stats, 2016. *StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share.* [Online]   
Available at: http://gs.statcounter.com/  
[Accessed 20 March 2016].

Vogeley, K. & Bente, G., 2010. "Artificial humans": Psychology and neuroscience perspectives on embodiment and nonverbal communication. *Neural Networks,* 23(8), pp. 1077-1090.

WEKA, 2015. *weka - Use WEKA in your Java code.* [Online]   
Available at: https://weka.wikispaces.com/Use+WEKA+in+your+Java+code  
[Accessed 26 October 2015].

Witten, I. H., Frank, E. & Hall, M. A., 2011. *Data Mining: Practical Machine Learning Tools and Techniques.* Third ed. s.l.:Morgan Kaufmann.

Ask about referencing the APIs and other sites frequently visited.

# To Do

* Sort out formatting for this DONE
* Ensure introductions follow description of following sections format
* Introduction
  + References for
    - Cortana DONE
    - Siri DONE
  + Sources for
    - User DONE
    - User preferences NOT NEEDED
* Design
  + Finish
* Implementation
  + Write about all sections
  + Insert code samples
  + Problems faced in implementation as separate section?
* Testing Evaluation
  + Problems faced
  + Test cases
  + Conclusion
* Research
  + Existing Projects
  + Talk more about papers content DONE