User Modelling in Search for People with Autism

Author: **Esha Massand**, Supervisor: **Keith Mannock**

MSc Computer Science
Department of Computer Science and Information Systems
Birkbeck College, University of London

Project report submitted in partial fulfilment of the requirement for the MSc in Computer Science

September 2015

Abstract

This project report presents my research and the development of a prototype web application to assist users with Autism when they search the web. The system has modelled user interactions with the search process into a user profile for this category of users, integrating insights from the core features of autism into the model. The system is integrated with an infra-red, motion controlled, user interface component to assist users with Autism during search. The project provides insights into how search can be improved for users with Autism.

word count (project text only) : 8443 words

Contents

Abbreviations

API	Application Programming Interface
AQ	Autism Quotient
ASD	Autism Spectrum Disorder
DSM	Diagnostic and Statistical Manual
GCS	Google Custom Search
HCI	Human Computer Interaction
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
KWIC	Key Word In Context
LEAP	LEAP Motion Controller
REST	Representational state transfer
RIFT	Oculus Rift Virtual Reality Head Mounted Display
TDD	Test Driven Development
UI	User Interface
UX	User Experience
VR	Virtual Reality

Definitions

Autism	Autism is amongst the most common neuro-developmental condition and it is currently estimated that 1/68 children meet criteria for Autism Spectrum [6]. Autism is five times more common amongst boys than girls (1/42 boys, and 1/189 girls). According to the Diagnostic and Statistical Manual (2013), Autism is characterised by persistent and early deficits in reciprocal social interaction and repetitive behaviours. Individuals vary from high functioning to low functioning (along a spectrum), with behaviours emerging around 2 to 3 years of age.
Stereotyped User Model	Stereotyped user models infer characteristics about a user from data gathered from other users within that distinct subset. They can be built quickly using clusters of characteristics of groups of individuals.

Search Engine Measures

Precision

The field of information retrieval defines precision as the fraction of retrieved documents that are relevant to the query.

$$Precision = \frac{|(relevant\ documents) \wedge (retrieved\ documents)|}{|(retrieved\ documents)|}$$

Recall

The field of information retrieval defines recall as the fraction of relevant documents that are retrieved by the search query.

$$Recall = \frac{|(relevant\ documents) \wedge (retrieved\ documents)|}{|(relevant\ documents)|}$$

1 Introduction

This project report presents my aim to research and develop a prototype web application for people with Autism when they search the web. This project will combine interactive, motion recognition hardware with search to improve the UI (user interface) of search for individuals with Autism.

1.1 Background Research

According to the Diagnostic and Statistical Manual [6], Autism is characterised by persistent and early deficits in reciprocal social interaction. Autism is the most common neurodevelopmental condition (1/68 children meet criteria for Autism Spectrum [6]). It is well known that interaction with computers is prominent in this group, and that individuals with Autism are more engaged when using technology that is receptive and interactive (e.g., games, responsive consoles, motion controlled devices) compared to technology that is not [17].

Research has shown that people with Autism are less context-sensitive [23]. Generally speaking, individuals with Autism prefer, and are more likely to engage in an item-specific, or detailed processing style. There is good reason to postulate then, that web-search queries are formed very differently to the typical population, more likely consisting of single-order associations. In addition to a detailed, item-specific processing style, individuals with Autism are also less likely to engage in a contextual processing style. For example in a search for ‘Guitar’, a contextual style of processing would imply an awareness that the word is related to ‘Piano’, but also that both words are hierarchically related to ‘Instrument’.

Working memory is the memory system that assists with the transient holding and processing of information for updating, learning and comprehension. In day to day life, a major factor determining the effectiveness of working-memory and the quality of information that an individual can recall (e.g., enter into the web-search), is the number of cues that are available at the time. For example, if presented with a list of 50 items to remember, you are more likely to remember that I presented ‘Zebra’ if I tell you that I presented an animal than if no cue of ‘animal’ was given. This type of ‘cued’ recall can be used to bolster recall of information, and is a technique that will be applied in the current project to enable individuals with Autism to formulate better search queries in their browser.

1.2 The Problems with Current Web-Search

The Internet is one of the largest resources of information. Search engines allow users to collate hundreds of links on a single topic, using only a few words or phrases. The typical user sorts the returned results into ‘relevant’ or ‘irrelevant’ categories, flexibly shifting (mentally) between one result and the next, to determine the relevance of each page returned by the search engine. Search allows the user to assimilate the information on the page into their knowledge and is an important learning tool. For people with Autism, the requirement is no different, however, I argue that for individuals with Autism, current search is not adequate. A large body of research has shown that ‘mental shifts’ are a known area of weakness for people with Autism [11]. The information is therefore harder to assimilate or learn, and judging the relevance of each document ‘on the fly’, and speedily, becomes near impossible. One successful Psychological technique to increase the assimilation of information for individuals with Autism is to present it in clearer, smaller amounts [31]. This technique avoids overwhelming individuals, and although less information is presented as a whole, whatever is presented to the user becomes ‘digestible’ and ultimately, the information as a whole becomes more understandable.

Search queries usually fall into one of three broad categories [12]. ‘Do’ queries which characterise transactions between the user and the search engine, for example when the user wants to do something such as *buy a plane ticket*, *listen to a song* or, *download a screensaver*. ‘Know’ type queries, which are informational in nature, usually covering a broad topic, for example, the *name of a band* or *restaurant in London*, *trucks*, or *Colorado*. The third broad category is ‘Go’ type queries, which are navigational in nature, for example, searching for a particular home page on the web, *YouTube* or *American Airlines*. There are also many stages to the search process [12]. After identifying the information need, the user must formulate a search query. The user must browse through results once the query has been entered into a search engine. The whole process can be repeated if the user is not satisfied.

1. Experience the need for an answer, solution, or piece of information.
2. Formulate that need in a string of words and phrases, also known as ‘the query’.
3. Enter the query into a search engine.
4. Browse through the results for a match.
5. Click on a result
6. Scan for a solution, or a link to that solution.
7. If unsatisfied, return to the search results and browse for another link or ...
8. Perform a new search with refinements to the query

1.3 Current Project

This project report presents a prototype browser-based application to assist users with Autism when they search the web. The application is hereafter referred to as Jellibeans[1]. Jellibeans utilises gesture and hand movement data recorded using the Leap Motion Controller (LEAP) to enable an interactive search in the web browser.

One aim of the current project was to build a tool that synthesised three leading search engines. The tool can be used to present the top results of search queries to ensure the best possible results were retrieved and presented to the user. The first stage was to therefore build the combination search engine and test the results on a user group with Autism. The goal of the research was to understand if the synthesis of the search engines enhanced search experience (in other words were users happy with the combination search), or, whether the results introduced redundancies or oddities in the results.

In programmatic terms, Jellibeans implements a research-guided user model in search. The developer for the current project has extensive research experience from working with individuals with Autism, and will apply these insights into the development of the Jellibeans. The development of this user model consisted of iterations of the 'research, development, test, evaluate' lifecycle. I conducted research surveys to identify differences in user queries and analysed search behaviour patterns from people with and without Autism. Using these data I built a set of features into search, to guide the user through forming a more complete search query. Development and evaluation included the implementation and testing of the features, and further implementation and revision of the software to enhance the search process and the precision of the search engine for the users' intended search query.

Jellibeans integrates a motion controlled user interface (UI) using the LEAP motion controller. The interface is very dynamic as opposed to static, which can hold users' attention for long periods of time. In the past, users with Autism have struggled to maintain their attention to sift through the large number of search results. In line with the attention difficulties individuals with Autism face, Jellibeans has been designed to reduce the amount of text on the search results page, presenting only 3 results per page, instead of the default 10.

[1]Jellibeans are a rainbow of colours, different sizes and shades, and the name represents the difference in style of processing of individuals with ASD.

1.4 Parametric Search Design for Web Applications

Search engines like Google apply strong reduction techniques to navigation of the web. For example, one common way this reduction pattern is implemented is by assuming the behaviour of the current user is similar to the behaviour of other users in similar situations. The principle applied by Google is to 'make it easy' for the user [18], by assuming that users form search queries similarly, and returning similar results to those users. This is often seen in other recommendation engines also, e.g., Amazon.

As we have seen in the introduction, users with Autism behave in different ways to typical users when navigating search. Users with Autism do not use the same key phrases when looking for documents with several attributes, i.e., queries that would be best formulated using several iterations of search, or multiple search parameters. This leads to an ineffective search; one that requires users to sift through results which are in large-part irrelevant, and a bad user experience.

Parametric search queries allow users to specify parameters for their search in an increasingly logical and structured way. As an example, consider the experience of searching for flights to a particular holiday destination. This requires high cognitive 'load' (remembering and manipulating arrival, departure, destination, timings, airlines, seat preferences etc.) so searches are often structured using fixed options (see Figure 1).

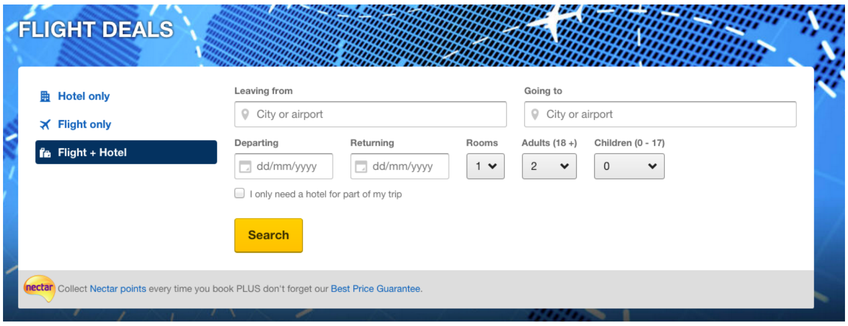


Figure 1: Expedia Parametric Search Example

For typical users, parametric search is more structured, and in some circumstances seems more natural than a free keyword search. It makes search queries easier to formulate in situations where there is a high cognitive load. We can apply this idea to web search for people with Autism, by asking them to enter criteria that can be applied to subgroups of search queries. Parametric search can assist the user with capturing the search parameters that are useful for a query, but it does not ultimately reduce the number of search results returned; the possibility of a large result set is most definitely true. So, although this tackles one aim of Jellibeans, further refinement of the search results would also be necessary.

1.5 Motion controllers

Individuals with Autism have poor attention [4]. The static 2-dimensional interface of many current search engines is unlikely to maintain adequate (sustained) interest levels. Individuals (and especially teenagers) with Autism spend a substantial amount of their time using computers, web, portable or console devices [30], as they find these more stimulating and attention-grabbing. For these individuals, computer-based technologies provide a stable, consistent learning environment that can be customized [25]. Furthermore, motion recognition devices can be programmed to make consistent responses to environmental triggers. These controlled and interactive environments have shown promise for improving social communication skills and reducing repetitive behaviours [13]. For the current project a motion-controlled learning environment will be bolstered to improve attention within search for people with autism.

Figure 2: UI component for the project [14].

1.6 Aims for Jellibeans

Jellibeans is a prototype search tool that assists users with Autism search and navigate the web. As search can be compartmentalised into 'Do', 'Know' and 'Go' type queries, it provides a natural way to tackle the issue of simplifying the search process for people with Autism. Users first identify the type of search they would like to carry out. Then Jellibeans better phrases a question to guide the user through their informational demand. Each stage of the search flow process is better tailored towards providing a concrete and unambiguous experience for the user with Autism, and reducing the amount of text on the page.

When using Jellibeans, users are assisted through the process of formulating a search query, much like a parametric search engine form (for example when you buy a car, you fill out many criteria which must be simultaneously searched for). Concretely phrased questions are aided by suggestions that appear laterally on the page. The user can add these suggestions into their search query without the need to type them in, just by using natural hand movements, tracked by the LEAP motion controller. The UI component makes the experience of searching the web more interactive and receptive for user input. The suggestions also serve as cues for the user, and to simplify the search process for them.

The implementation of a parametric-like search means the process of forming a query is more tightly defined and predictable for the user compared to the current systems for web search. The size of the result-set is also reduced, and adapted towards users with Autism.

Each of these features were tested in one-on-one interviews and a group of users with Autism[2]. The prototype was integrated into the web browser and within a motion-controlled environment. The prototype was modified accordingly in line with the outcome of several stages of user research. An agile methodology was used to build the final prototype using static HTML. The static website prototype can be found at esha.mseth.co.

Using Jellibeans, users can complete a screening tool for Autism and save their score (which is stored in a database), so that they can log in and out of Jellibeans without having to complete the Autism quotient each time they log in. Jellibeans will remember their Autism score the next time they sign in.

The final product is presented at the end of this report, and is written using php, JavaScript, HTML and CSS (using BootStrap). The dynamic website has a greater degree of functionality and promise for future updating. The dynamic website was created once the 'quick and dirty' prototype was completed. The final prototype can be found at jellibeans.mseth.co.

The core features of Jellibeans are:

1. A web-based search tool, that uses a *stereotyped*[3] user model of Autism to filter search results. The features of the model were determined following analyses of survey responses gathered from 7 individuals with Autism

[2]Autism diagnostic status was determined using the Autism Quotient, which strictly speaking is not a diagnostic assessment tool. However the measure has a 79% sensitivity to Autism if the user scores above 32 on the measure.

[3]Stereotyped user models infer characteristics about a user from data gathered from other users within that distinct subset.

or a high occurrence of autism-like symptoms (above a score of 30 on the Autism Quotient). The survey was administered online using a web-based survey engine [33], and focused on identifying the differences in user search query generation for people with autism. The individual participants also took part in a one on one interview session and focus group at the end of project.

2. Jellibeans implements a parametric-like search to gather a number of search keywords from the user. Jellibeans returns small snippets on the results page to a maximum of 3 search results per page (the result page is not verbally-overloaded).

3. Jellibeans prioritises results which have first-order semantic relations to the query words. Jellibeans employs a boolean 'AND' between each search word searched for, so that results are more likely first order semantic relations to the searched query.

4. Jellibeans uses a data store to saves the user's Autism score for the next time they sign in. The user's score and the search queries that were formed using Jellibeans, were used to evaluate the search tool. The datastore was implemented using MySQL.

5. Jellibeans implements within a motion controlled UI which uses the LEAP controller. This makes search more interactive and receptive for users. The LEAP also works to maintain the attention of the user.

1.7 Challenges

There are several expected challenges for the current project. One such challenge concerns the very nature of trying to develop a tool for individuals with Autism, knowing that there is a large variability in the phenotype (or presentation) of the condition. Trying to address the clinical features of Autism across the entire spectrum is no mean feat. The sample selected for the current study were *verbal* adults with high functioning Autism, or Asperger Syndrome.

Another challenge relates to gathering feedback on Jellibeans. I conducted focus groups and interviews which required lot of verbal communication. This is always a challenge when working with individuals who are known to have difficulties with communication.

More of a technical challenge concered the reliance on open source APIs, open source libraries and packages which were not regularly maintained. For resources that were incomplete, I implemented the tools myself, or, I searched for alternative ways to achieve that goal.

To complete the aims for the current project within the time frame was challenging given that most of the project used novel technologies for the developer. Jellibeans was implemented in an agile, iterative fashion, so that a static prototype (esha.mseth.co) could be quickly developed before I developed a final dynamic website for Jellibeans (at jellibeans.mseth.co). The project also included building the user model, using a dynamic website framework, learning to use several new or unfamiliar programming languages including php, HTML, CSS, SQL, JavaScript and JQuery, and, integrating the motion controlled user interface within that framework. Learning to use these novel technologies simultaneously, and with a time constraint was a challenge.

I did not expect to have to integrate Jellibeans with a SQL database, as it was expected that I could store user scores in their Google+ profile. However, as user scores would not persist after a session in user's Google+ profile, I decided to implement Jellibeans with its own database. The time needed to deploy the final website with a database was not accounted for in the planning for the project, so this was an additional challenge.

Conducting surveys, interviews and a focus group to gather feedback on the tool, and to implement this feedback into a revised prototype for Jellibeans was a time consuming task. Often this required that I worked around participant schedules. Two participants out of the 7 were unavailable during the user feedack stage of the study, and so these particpants were excluded from any subsequent analyses.

2 Project Specification and Design

The requirements for Jellibeans web application are described below.

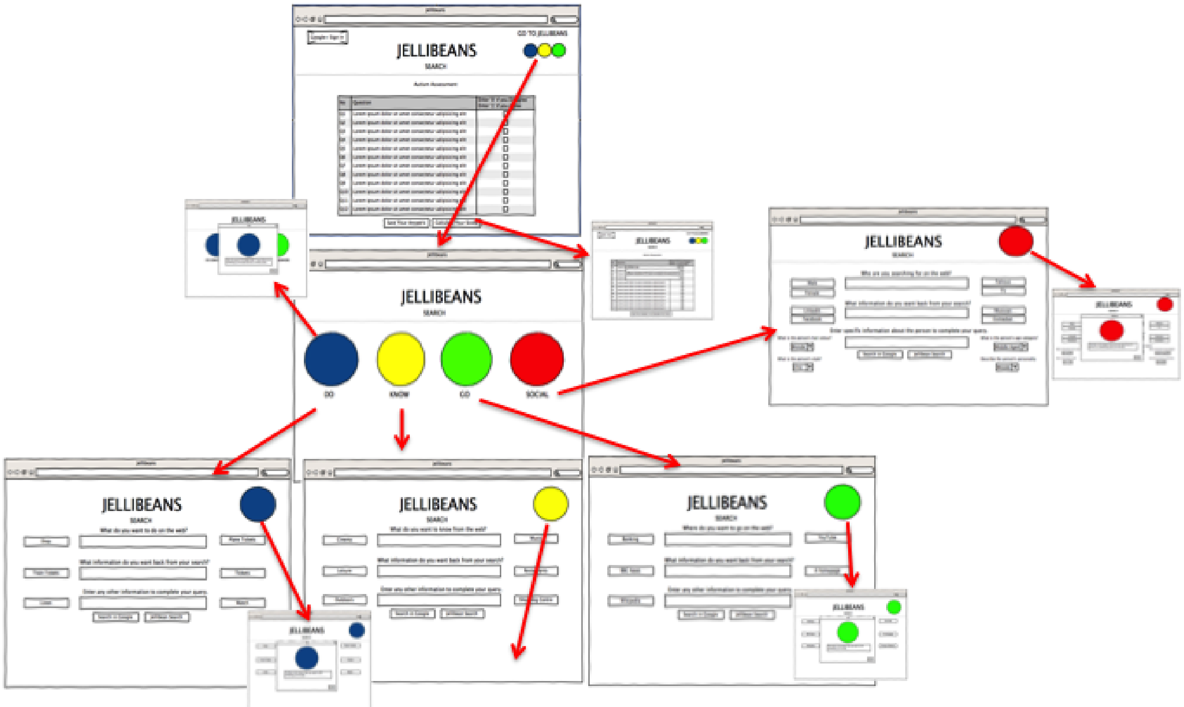


Figure 3: Wireframe Sketches for Jellibeans User Flow. A prototype version of Jellibeans using static html pages can be found at http://esha.mseth.co. A 'dynamic' version of Jellibeans using routes rather than static HTML, which is fully LEAP motion controller-enabled can be found at jellibeans.mseth.co.

Jellibeans is a dynamic website and implements a combination search. This means it synthesises the results from three of the largest and most popular engines; Google, Bing and Yahoo [24].

Jellibeans integrates with Google+, to allow the user to sign in and have their profile picture and name displayed, enhancing the feel of a personalised search engine (see Figure 4).

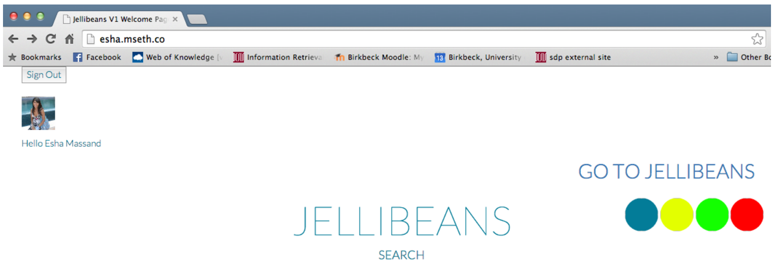


Figure 4: Using the Google+ API Allowed the User to Sign In and Retrieve User Name and Profile Picture.

Jellibeans implements a user-model for people with Autism, within search. The user-model employs a parametric-like search, that accurately quantifies the user's search query by 'breaking it down' into smaller sub-queries, until the user has exhausted the information they can provide to complete their search. Jellibeans also offers *suggestions* to guide the user towards forming succinct, accurate and complete search queries (see Figure 3).

Once signed in, the user can fill out the Autism Quotient[4] and get feedback regarding their autism symptomology. Users can save their responses for each question to a local comma-separated-values file for their reference. Their score is also saved to a data store (database).

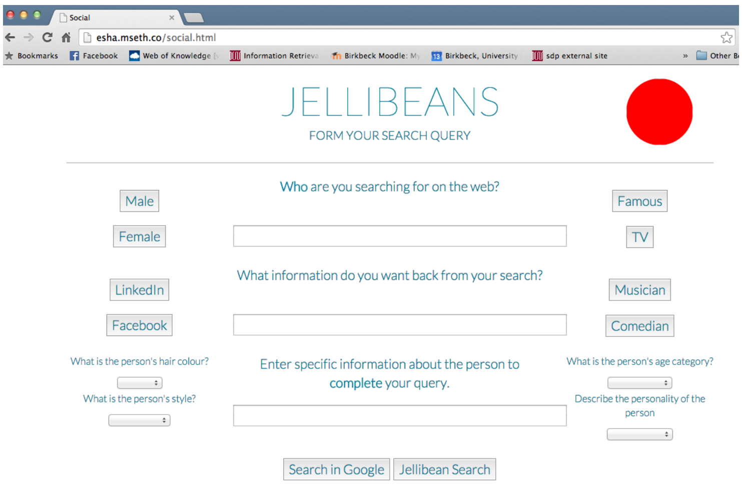


Figure 5: An example page from Jellibeans: the Social Query page.

[4]The Autism Quotient [1] is a Psychological assessment and validated screening tool for autism spectrum disorders.

Jellibeans web application is integrated with a motion controlled UI, the LEAP controller (see Figure 6) giving the application a receptive and interactive user interface.

Figure 6: User Forming A Jellibean Search Query with LEAP Using Both Hands.

Jellibeans prioritises and displays the three highest precision search results for users. Recall of the search engine is sacrificed in line with the demands of the user model, to have less textual information on the page.

Jellibeans reduces the amount of text on the webpage by having results display on a modal (a hovering display panel on the page) which can be easily closed, and reopened to display results (see Figure 7). Due to the nature of parametric search, boolean 'AND' operators are used in Jellibeans. This restricts the search results displayed. The results from Jellibeans are more likely a direct semantic relation to the search query.

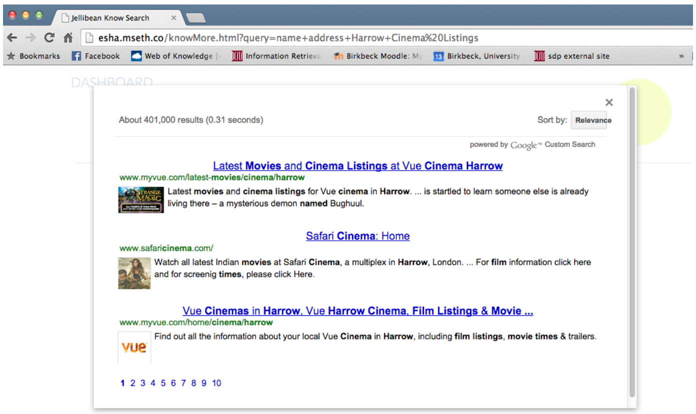


Figure 7: Top 3 results are presented back to users in a modal to reduce information on the page. The modal can easily be closed and reopened.

3 System and Software Architecture

3.1 Architecture of the System: 3-tier Architecture

A client-server software architecture was used for Jellibeans. The presentation layer consisted of the user interface and motion controller. The presentation layer, in turn, has access to the application layer, which controls the functionality of the system by performing detailed processing. This included management of Jellibeans within the **Laravel** framework.

Laravel is a PAAS (platform as a service), which manages the deployment of Jellibeans to **Digital Ocean** Cloud Server. Laravel implements a *rewrite engine*, which is software that modifies the URLs appearance so that the URL is shorter and more informative/relevant for the user. The technique loosens the coupling between the HTML files and the URL that is presented to the user.

- The application layer therefore included (amongst other things):

  The php routing code. The routing code allowed the URI (Uniform Resource Identifier) to be broken up into parameters and processed to determine which view (presentation/html file/web page) to present to the user.

  The JavaScript code used to implement the custom search.

  The SQL execution code for the storage of data in the data store.

The data store's primary responsibility was to manage the storage and retrieval of user data via the application layer.

The presentation, application and data store layers were maintained as three distinct modules. The advantages of this type of design is that it allows each of the tiers to be modified, upgraded or replaced independently of one another. For example, if Jellibeans were to be implemented with another motion controller (e.g. the Oculus VR Rift), it would only impact the Presentation tier (see Figure 8).

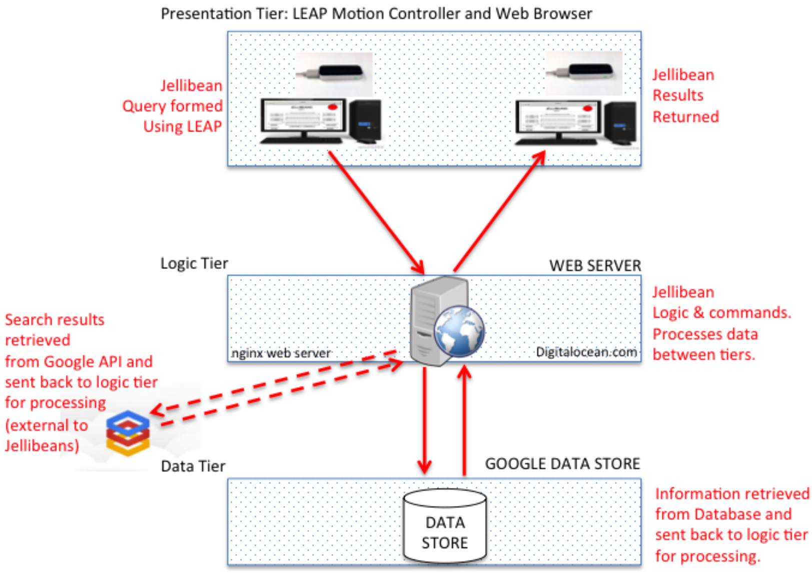


Figure 8: 3-tier architecture implementation

To see a list of the API's, development languages used and their justifications, see Appendix 10.3.

Using the Laravel framework the following modules were implemented. The APIs and languages are listed in the context they were used below.

- Presentation Module
  The presentation module consists of:

  Jellibeans Web Application: Rendering the web application to the browser (HTML, CSS PHP and Bootstrap).

  Rendering a Parametric-Like Search in the browser: The user is asked specific questions to break down their search query into smaller manageable terms, i.e., into 'Do', 'Know', 'Go', or 'Social' queries, and then into direct search terms. This reduces the ambiguity individuals with Autism face when forming a search query. The web application provides suggestions or cues for users. Once the search terms are entered by the user, they are sent to the logic module to form the search query string, and then onto the data store to retrieve the results which are subsequently rendered onto the screen for the user (uses HTML, CSS, php and Bootstrap).

  Google+ Integration: Jellibeans integrates with Google+ API to retrive and render user profile data. This enhances the feel of a personalised search engine. Once signed in, users complete a diagnostic assessment of Autism (the Autism Quotient [1]). The user's entries are sent to the logic module to be calculated. A user can also save a copy of their diagnostic information to a local file on their computer (uses JQuery, JavaScript and Google+ API).

  The user's AQ score is saved to the database once the score is calculated. The score is fetched from the database whenever the user logs in. A unique identification number assures that the user data is never overwritten or duplicated.

  LEAP UI Component: Jellibeans is a LEAP enabled website, and has a Motion Controlled UI in the web browser. The motion controller provides an interactive search experience, and maintains the user's full attention during search (uses LEAP SDK and LeapStrap).

  Importantly - the Presentation module includes the routing logic to determine the view (or web page) to present to the user.

- Logic Module
  At the logic layer, Jellibeans works to build user search queries from the elements on the web page. The user's search-parameters are entered when they form their search query at the presentation layer.

The Logic module enables:

Logic for the user to carry out a single search that retrieves results from a combination of search engines. The results returned are those synthesised from Google, Bing and Yahoo. The aim of this feature is to increase recall of the search engine by increasing the number of potential relevant results (programmed into a Java applet).

Logic to integrate with the Google+ API (uses JavaScript and Google+ API).

Calculation of the users' AQ Score using the questionnaire entries that are entered at the presentation layer. The request is sent to the database (MySQL). The score is returned to the presentation layer after calculation (uses JavaScript).

Logic to formulate the user's search query, given their entries at the presentation layer (uses JavaScript).

Logic to send the query to the Google Data Store, and, to retrieve the search results to render them to the presentation layer (uses JavaScript and Google Custom Search API).

Logic to prioritise the three highest ranking search results and to reduce the text and verbal information on the page (in line with the aims of Jellibean Search), (uses JavaScript and Google Custom Search API).

The logic layer communicates with the Google Data Store layer to retrieve the search results for the user.

- Data Store Module

  Jellibeans stores and retrieves the user's Autism score each time the user signs in from a database (uses MySQL).

  Jellibeans works with the Google Custom Search engine to retrieve the (indexed) documents from the Google data store. The documents that are retrieved match the participants' search query string after processing at the logic module level.

4 Software Design Pattern

I will now describe the design pattern that guided the software development process. Jellibeans uses the model-view-controller (MVC) software design pattern. There are three components to the MVC pattern (see Figure 9).

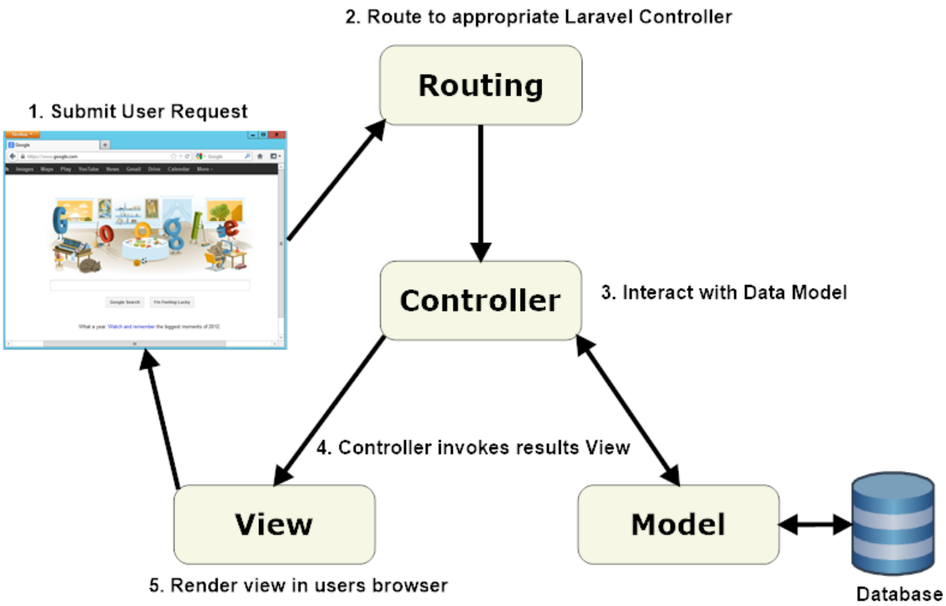


Figure 9: Application Model-View-Controller pattern as described by the Laravel framework[22].

For Jellibeans this results in the following components and classes:

- The **Model**- is the domain or real world entity that the software is built around. The model in this case is the User.

- The **View**- is the visual representation of the user, given a context. So, in this case it is the resulting markup that the php framework (Laravel) renders to the users browser (i.e., the html code). The view generates the user interface, and is accessible via the user (the Model). The view doesn't handle the data that comes in (although users input the data into the view that is presented to them on the application's webpage). There are 6 views of the system.

  index.php renders the homepage
  search.php renders the Search screen
  do.php renders the Do search page
  know.php renders the Know search page
  go.php renders the Go search page
  social.php renders the Social search page
  result.php renders the Results page

- The **Controller**- Coordinators provide links between the view and the model. Coordinators are responsible for input-processing and determining the behaviour of Jellibeans, such as which new screen to render. Controller.php is an abstract class that all other controllers extend (see Figure 10). Jellibeans has the following Controllers:

  PagesController.php controls the page behaviour when a user clicks on the navigation buttons, for example, return the user back to the homepage.

  RegistrationController.php controls the model's storing and registering of the user and user credentials.

  SearchController.php controls the behaviour of which search to carry out given the view.

  SessionsController.php controls the behaviour of the view depending on whether the user is logged in or logged out. For example, if the user is logged out it will display the register and login buttons on the html page. If the user is logged in, it will return the logout button.

  UserController.php verifies the CRUD (Create, Read, Update and Delete) actions form the resource.

SearchController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;
use App\Http\Controllers\Controller;

class SearchController extends Controller
{

    public function index()
    {
        return view('search.index');
    }

    public function searchDo()
    {
        return view('search.do');
    }

    public function searchKnow()
    {
        return view('search.know');
    }

    public function searchGo()
    {
        return view('search.go');
    }

    public function searchSocial()
    {
        return view('search.social');
    }

}
```

Figure 10: Example of a Controller Class (the SearchController class) which return views of the system dependent on user behaviour.

- The **Routes** - decompose the configuration of which action of that controller should receive the request.

  Application Routes were set up which listed the URIs that Jellibeans should respond to, and the Controller to call when that URI is requested (see Figure 11).

```
Route::get('/', ['as' => 'home', 'uses' => 'PagesController@index']);
Route::get('home', 'PagesController@index');

Route::get('search', 'SearchController@index');
Route::get('do', 'SearchController@searchDo');
Route::get('go', 'SearchController@searchGo');
Route::get('know', 'SearchController@searchKnow');
Route::get('social', 'SearchController@searchSocial');
Route::get('results', 'SearchController@results');


Route::post('home', 'UsersController@saveScore');


// Authentication
Route::get('auth/login', 'Auth\AuthController@getLogin');
Route::post('auth/login', 'Auth\AuthController@postLogin');
Route::get('auth/logout', 'Auth\AuthController@getLogout');

// Registration
Route::get('auth/register', 'Auth\AuthController@getRegister');
Route::post('auth/register', 'Auth\AuthController@postRegister');
```

Figure 11: Example class of how Routes were defined.

Then present a class diagram which has

A high level class diagram of the system can be seen in Appendix 10.1.

A use case diagram of the system can be seen in Appendix 10.2.

5 Implementation

The prototype version of Jellibeans is hosted at esha.mseth.co. This prototype was implemented quickly for use with user testing. The prototype version of Jellibeans used static web pages which allowed the prototype to be quickly tested and finalised.

Jellibeans was finalised into a dynamic website and is hosted at jellibeans.mseth.co. The advantages of a dynamic website over a static website are that it offers increased functionality for users, no need to edit individual html pages if there are subsequent modifications, and no code repetition.

The next section of the report describes how each module of work was completed.

5.1 Combination Search Engine: Google, Yahoo and Bing.

To implement the combination search engine three API's were used. Namely, the Google Custom Search API, Yahoo BOSS Java API and Bing Search API.

To get started with the Google Custom Search API, I created a project called *Jellibeans* in the Google Developers Console, and an OAuth 2.0 Client ID. I obtained a Consumer Key and Secret to use the API, and used these in the application code to access the Google Custom Search Engine (see Figure 12).

```
public class GoogleCustomSearchApiApplet {

    String qry = null;
    public ArrayList<String> jSoupGoogleLinks = new ArrayList<>();

    public void cse() throws IOException {

        String key = "AIzaSyCnAIDiZchNkR0OTBH3NMMNt4GmRiwpdnA";// server key

        String cx = "008818185974073145685:ga_fmgk9gf0";
        URL url = new URL("https://www.googleapis.com/customsearch/v1?key="
                + key + "&cx=" + cx + "&q=" + qry + "&alt=json");
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Accept", "application/json");
        BufferedReader br = new BufferedReader(new InputStreamReader(
                (conn.getInputStream())));

        String output;
        while ((output = br.readLine()) != null) {

            if (output.contains("\"link\": \"")) {
                String link = output.substring(output.indexOf("\"link\": \"")
                        + ("\"link\": \"").length(), output.indexOf("\","));
                jSoupGoogleLinks.add(link.toString());
            }
        }
        conn.disconnect();
    }
```

Figure 12: Jellibean Applet Combination Search Engine

Following that I also registered my JavaScript origins within the console to access the Google+ API, and redirected URIs so that once users Sign-In using their Google+ login credentials they are redirected to the Jellibeans homepage. This was done so that when users sign in, they can access their google profile from Jellibeans and are greeted by the application (see Figure 14), giving the user the feel of a more personalised search experience.

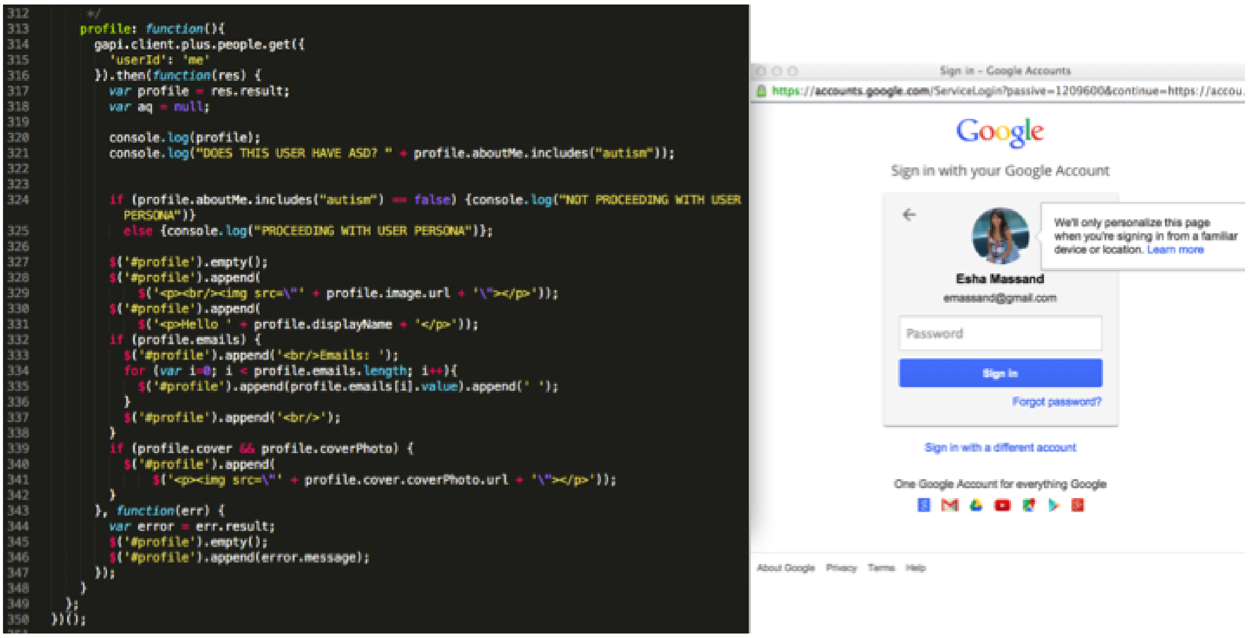


Figure 13: Google+ Integration Code and Sign In Pop-Up (JQuery).

Following a similar protocol for Yahoo and Bing Search API's, I created projects in the Yahoo Developers Network, and Microsoft Azure Marketplace, and purchased an API Consumer Keys and Secrets (for use with the APIs).

5.1.1 Web Scraping

As Bing and Yahoo Search API's were not opensource, and charged per search, I explored the efficacy of web scraping using the JSoup API. With the JSoup API I was able to retrieve the search results, using a different `href` filter for each search engine. Web-Scraping is against Google policies, so this could technique could not be used to retrieve Google Search links (see section 5.3 of Appendix 10.4 for Google Terms of Service).

I used the Jsoup API [21], which is a Java-written API for HTML. The library provides methods to conveniently extract data using the DOM (Data Object Model) and CSS (Cascading Style Sheet) methods.

The combined Google, Yahoo and Bing results were integrated these into a Java Applet that runs in Eclipse Luna IDE (see Figure 15). The program ran so that the user could enter a search query and have the results presented back to them for inspection. Only the top 10 links from each search engine were presented to users. Results were ranked prior to presentation, such that result 1 from Google was followed by result 1 from Yahoo, and that was followed by result 1 from Bing. Then results 2 from Google, Yahoo and Bing were presented and so on, until the 30 links were produced (top 10 links for a search from each search engine), in ranked order from the three search engines.

```
public void jsouphtmlYahoo() {

    Document doc;
    try {

        String searchPage = "https://uk.search.yahoo.com/search?p=";
        String tempSearch = searchPage + query;

        // need http protocol
        doc = Jsoup.connect(tempSearch).get();

        // get page title
        String title = doc.title();
        String selection = "h3.title > a";

        Elements links = doc.select(selection);
        for (Element l : links) {
            jSoupYahooLinks.add(l.toString());
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

}

public void jsouphtmlBing() {

    Document doc;
    try {

        String searchPage = "https://www.bing.com/search?q=";
        String tempSearch = searchPage + query;

        // need http protocol
        doc = Jsoup.connect(tempSearch).get();

        // get page title
        String title = doc.title();
        String selection = "h2 a";

        Elements links = doc.select(selection);

        for (Element l : links) {
            jSoupBingLinks.add(l.toString());
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

}
```

Figure 14: JSoup HTML Parser for Yahoo and Bing Search Engines.

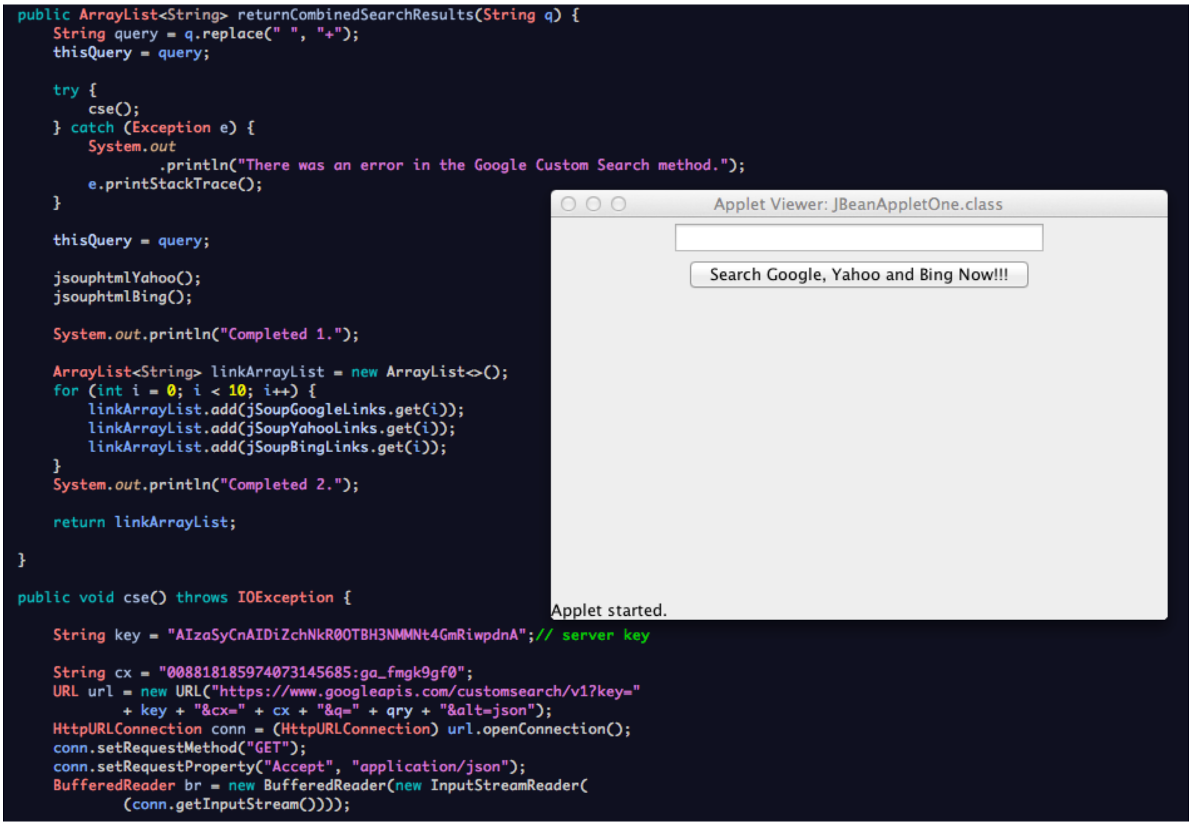


Figure 15: Yahoo and Bing Results Integrated with Google Custom Search to Produce the Complete Query Results.

5.2 Building a User Model of Autism.

To identify the features of the user model to build, I ran a study to collect example search queries on a set of informational needs. Data was gathered from 37 participants. The participants were asked to give examples of search queries they would use, for example, to identify the name of a song they had heard (given the lyrics), or the name of a breed of a dog they had seen (given a picture of the dog). There were in total 8 search queries; the study was distributed widely via Surveymonkey.com [33] and can be seen in the Appendix 10.5. All participants were also asked to indicate which search engine they most often use and to complete the Autism Spectrum Quotient – a 50-item screening questionnaire for Autism (see Appendix 10.7). The participant responses were then analysed.

Participants were divided into two groups; low AQ scorers (scores below 30, indicating a low number of Autism traits), and high AQ scorers (scores equal to and above 30, indicating a high number of Autism traits). There were 30 low AQ scorers and 7 high AQ scorers.

5.2.1 Differences in Search Queries Between Users With and Without Autistic-like traits.

A qualitative analysis on the search query strings from both low and high AQ scorers are reported below.

In both groups, Google was the prefered search engine by far, with all participants reporting that they used Google as a first choice. No one in the current sample used Yahoo or Bing.

The low AQ scorer responses were analysed together to establish a baseline answer. This was generated using a frequency criterion of 40% i.e., if 12 out of 30 respondents or more generated the same portion of a query string given an informational need, it was included in the model below. If two responses were equally as common, both are reported in the model. Data was discarded when a response indicated that the participant would do an image search, as this was not the aim of the survey. The results from the frequency analysis are presented below.

1. You hear a song on the radio with the lyrics, 'Look at your children', and you want to download it. What would you type into search on your favourite search engine to find out what song it was?
   Look at your children song.
   Look at your children lyrics.
2. You've lost touch with an old school friend (you went to St. Mary's School). What key words/queries would you use to find them?
   St. Mary's School Year of X.
3. How would you identify what this is using a search engine (pretend you don't know what it is called). What key words/queries would you use?
   Star shaped brown plant.
4. How would you find out the name of this famous person using a search engine? What key words/queries would you use?
   Brown hair famous young women.
5. How would you identify what breed this animal is using a search engine? What key words/queries would you use?
   Small dog fluffy breed.

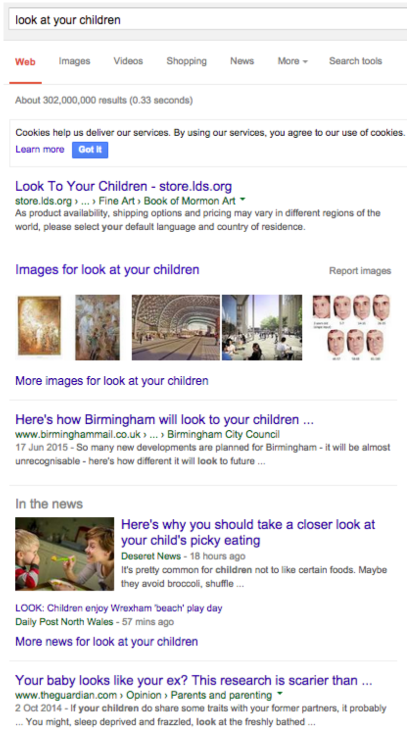
6. Your friend and you can't agree on how Thandie Newton pronounces her first name. How would you resolve this using a search engine?
   Thandie Newton pronunciation.

7. What would you search for to identify this pattern's name, and which country it originates from?
   Repeating square maze pattern border.

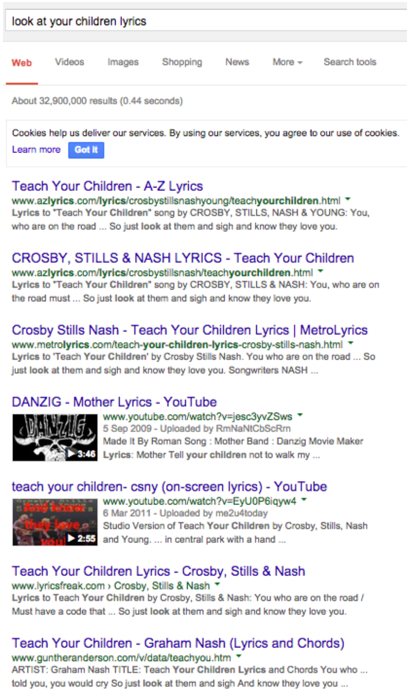
8. How would you search for delay's relating to your (imminent) flight to Paris?
   Flight number, Paris, airport, flight time.

There was a lot more variation in the responses gathered from the group with Autistic-like traits. The responses can be seen in Appendix 8.2. The following observations were made for the users in the high AQ group:

1. There were an increased number of incompletely-formed queries. In the high AQ group, participants were more likely to miss off words in the query string. For example, when searching for the name of a song contating the lyrics "Look at your children", *all* low AQ scorers searched for "Look at your children lyrics", whereas two out of seven respondents in the high AQ group did not put 'lyrics' or 'song' in the search query. When these search strings are entered into Google, the results are very different (see Figure 16). A larger number of results are returned to the incomplete query (302,000,000 compared to 32,900,000). In this instance, the high AQ user group were presented with results that have a lower precision, i.e., more irrelevant information that they must sift through to find the answer to their search query.



(a) Incomplete query results.



(b) Complete query results.

Figure 16: High and low AQ scorers both formed search queries accurately, however there was an increased tendency to omit the word "lyrics" in the high AQ group resulting in very different search results.

2. Although many high AQ scorers' formed query strings well, there was increased use of idiosyncratic words in the query strings that were formulated. This is in line with previous research that suggests that people with autism organise information in more subjective and individual ways [5]. For example, referring to the picture of the dog as 'yorkie pooh' (not listed in frequency index), 'aeroplane' (frequency of 8254 words per million) instead of 'plane' (frequency of 33900 words per million) or 'flight' (29535 words per million) , 'miniature' (less than 4973 words per million) instead of 'small' (185463 words per million). These words are considered more idiosyncratic because of their lower frequency of use in the English language. Search engines use term frequency to determine if a document is relevant to the users search query. If the frequency of words used to form search queries differs between low and high AQ scorers, so will the rankings of the returned search results.

3. One individual of the 7 individuals in the high AQ group demonstrated ambiguous use of third-person pronouns, which is characteristic of some indi-

viduals with Autism [26]. This includes using first names. This is particularly detrimental to search engine query strings because the use of names disrupts the term frequency - inverse document frequency weighting [34] of the search query and subsequently the results returned to the user.

4. For questions that were 'social' in nature (e.g., featuring a face of a famous woman), 2 out of 7 individuals in the high AQ group indicated that these were types of queries that they would not normally be interested in, and so "wouldn't bother asking it". For these queries, it was common for individuals in the high AQ group to include information in their search query string that was extraneous to the search question itself. For example, in query 4 above (which asked respondents to indicate how they would identify a famous person), 2 high AQ scorers included information about the woman's earring. Inevitably this 'dilutes' the search query, and in turn results in reduced precision for the search engine.

5.3 Transforming the User Query.

Given the set of observations in the data reported above, the aims were to 'transform' queries made by individuals in the high AQ group to queries more similar to the low AQ group. The search engines already handle some of the observations from high AQ scorer queries. For example, the use of pronouns ('I' and 'You') is already taken care of with the use of stop words. The aim of the project is to therefore address issues that result in the search query string being misleading, and returning differnt results to low AQ scorer search queries. The rule engine is a concrete and operationalised framework, for a theoretically-grounded stereotyped user model of autism within search.

Jellibeans works to address multiple observations made during the data collection stage, so the order in which the observations are tackled in the user model is not linear or consecutive by any means. As a general rule, changes will take the form of 'add on' questions that aim to structure the individuals search query in a logical way, so that key search terms are not dropped. A structured query formation will assist the user with less idiosyncratic search queries.

Building upon the reserach discussed in the introduction [12] (that is, that search can be compartmentatlised into three broad categories: Do's, Know's and Go's) and, to address the observation of idiosyncrasy (point 2 above in the current research), Jellibeans will prompt the user to categorise their search query into one of 3 possible types of queries, a 'Do', 'Know' or, 'Go'. 'Do' queries are when they want to do something on the web like, buy a cinema ticket. 'Know' queries for when they want to know something from the web like, what time the cinema closes. And, 'Go' queries when they want to go somewhere specific on the web like the cinema homepage. Jellibeans will integrate a fourth subtype of query – the ‘Social’ query, especially for the user group in question. ‘Social’ queries are for when their search is about something social; a person, or group of people. Social queries have additional functionality beyond the Do, Know and Go query pages, to reflect the additional difficulties individuals with Autism have when formulating Social search queries on the web.

Each type of query is associated with a different colour. ‘Do’ is blue, ‘Know’ is yellow, ‘Go’ is green and ‘Social’ is red. Once the user has selected that type of query, this colour will be prominent on the page throughout their search to serve as a visual reminder of the task. For individuals with Autism, this technique works to reduce the **working memory load** for the user in the long-term, as it serves as a **goal-directed cue**. Two things we know are difficult for people with Autism are maintenance of information in working memory, and goal-directed tasks involving a high demand on Executive Functioning [5].

To address observation 1, that user queries were often submitted when incomplete, suggestions as well as drop down menus have been implemented on each search page. Users can click on suggestions, and select from the menus if they want the keyword to be added to their search. As individuals with Autism characteristically have an uneven cognitive profile, where verbal ability is usually lower than performance skill [7], the cues will assist users who have particular word-finding difficulties.

Once the user has pressed ‘Jellibean Search’, and requested their search results be returned, Jellibeans collates all information the user has entered into the search boxes, as well as any suggestions, and options from drop down menu’s that were selected. To achieve a good precision and specificity, Jellibeans implemented ‘AND’ boolean operators between each input of the parametric search. This also resulted in a drop in recall, but as only the top 3 results were to be presented, this was in fact advantageous for Jellibeans. The results are then presented to the user, with the exact phrase that was searched for. This means that the user can clearly see their final search query, and rather than a ‘black-box’ system, Jellibeans also works to train the user for the long-term about specific ways to improve their personal search query strings.

Because the current research identified that user’s with Autism formed search queries were particularly incomplete for socially-grounded informational needs,

[5]Executive functions - also known as cognitive control and supervisory attentional system) is an umbrella term for the management, regulation and control of cognitive processes, including working memory, reasoning, task flexibility, and problem solving as well as planning and execution [10].

extra functionality has been added to the 'Social' query page. For example, if the user has indicated that they are searching for a famous person, they will be prompted for descriptions about that famous person's personality, character age bracket, and physical descriptions amongst other attributes. These features work to ensure that for this type of query, where users with Autism have most difficulty, that user search queries are more complete than they otherwise would have been (see Figure 17).

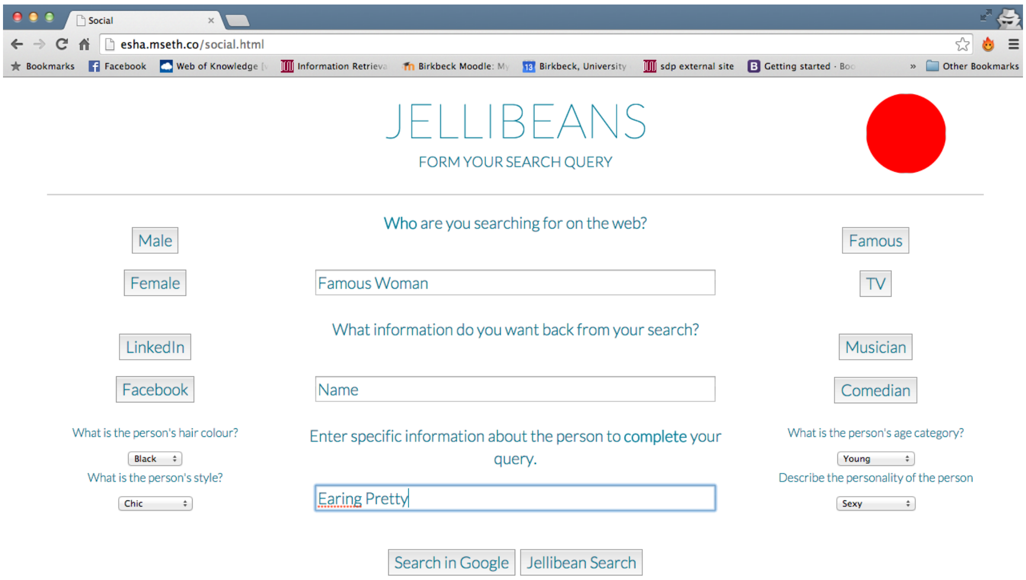


Figure 17: Example of User's Social Search in Jellibeans.

To further address Core Feature 3 of Jellibeans, that is, that results should be returned to the user in smaller snippets rather than large sets of text, I modified the Jellibeans Custom Search Engine to select the 3 highest ranked (precision) results, and to present these back to the user on the first page of their search. The 3 top results are presented on a floating modal[6], so that the user has the option to close and reopen the results very easily, without having to redo the search itself (see Figure 7).

5.4 Storing the User Autism Score Data.

The Autism Quotient score and all 50 of the responses from the user will be stored on a MySQL database system. The database configuration for the system was specified in a configuration php file (see Figure 18).

[6]A modal is a feature in the Bootstrap Framework, for more information see Appendix 10.3

```
],

'mysql' => [
    'driver'    => 'mysql',
    'host'      => env('DB_HOST', 'localhost'),
    'database'  => env('DB_DATABASE', 'jellibeans'),
    'username'  => env('DB_USERNAME', 'forge'),
    'password'  => env('DB_PASSWORD', ''),
    'charset'   => 'utf8',
    'collation' => 'utf8_unicode_ci',
    'prefix'    => '',
    'strict'    => false,
],
```

Figure 18: MySQL Database Configuration for Jellibeans.

Fillable attributes for the user model were specified so that the 'user' model itself could be created. These fillable attributes are fed into the model constructor via mass-assignment (see Figure 19).

```
class User extends Model implements AuthenticatableContract,
                                    AuthorizableContract,
                                    CanResetPasswordContract
{
    use Authenticatable, Authorizable, CanResetPassword;

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'users';

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = ['name', 'email', 'password', 'password_confirmation', 'score'];

    /**
     * The attributes excluded from the model's JSON form.
     *
     * @var array
     */
    protected $hidden = ['password', 'remember_token'];
}
```

Figure 19: `fillable` and `guarded` properties were used for attributes in the database to prevent blind SQL injection / modification of password information, and to limit security concerns.

To prevent user's password information being compromised (in terms of security), this information was given the 'guarded' property (rather than fillable by the user at registration/login). Passwords were also bcrypted[7] before they were

[7] bcrypt is a key derivation function for passwords [3]

stored in the database (see Figure 20).

```php
/**
 * Create a new user instance after a valid registration.
 *
 * @param  array  $data
 * @return User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => bcrypt($data['password']),
    ]);
}
```

Figure 20: User passwords were bcrypted before being stored to the database for security.

The Jellibeans model / table had to also be created, as follows (see Figure 21).

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.04 sec)

mysql> create database jellibeans
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| jellibeans         |
| mysql              |
| performance_schema |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql> use jellibeans;
Database changed

mysql> show tables;
Empty set (0.00 sec)

mysql> create user 'esha'@'localhost' identified by 'x';
Query OK, 0 rows affected (0.06 sec)

mysql> grant all privileges on * . * to 'esha'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Figure 21: MySQL Used to Create and Query the Jellibeans Database.

I created users in Jellibeans to ‘test’ that the user AQ scores were being saved to the Jellibeans database correctly. The User model /table was then queried using the SQL command: `SELECT * FROM USERS;`.

Prior to the live deployment of Jellibeans (from the localhost), the database needed to be migrated using `php artisan migrate` which made use of the Laravel migration files to build the columns in the database tables (see Figure 22, and Figure 23). These migrations were paired with Laravel’s schema builder to build the Jellibeans database schema. Migrations are useful as they allow version control if ever an operation needed to be “rolled back”, or, if in the future it was decided that the table needed additional columns.

```php
<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password', 60);
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('users');
    }
}
```

Figure 22: User Model Migration (Database Table).

```php
<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('password_resets', function (Blueprint $table) {
            $table->string('email')->index();
            $table->string('token')->index();
            $table->timestamp('created_at');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('password_resets');
    }
}
```

Figure 23: User Queries Model Migration (Database Table).

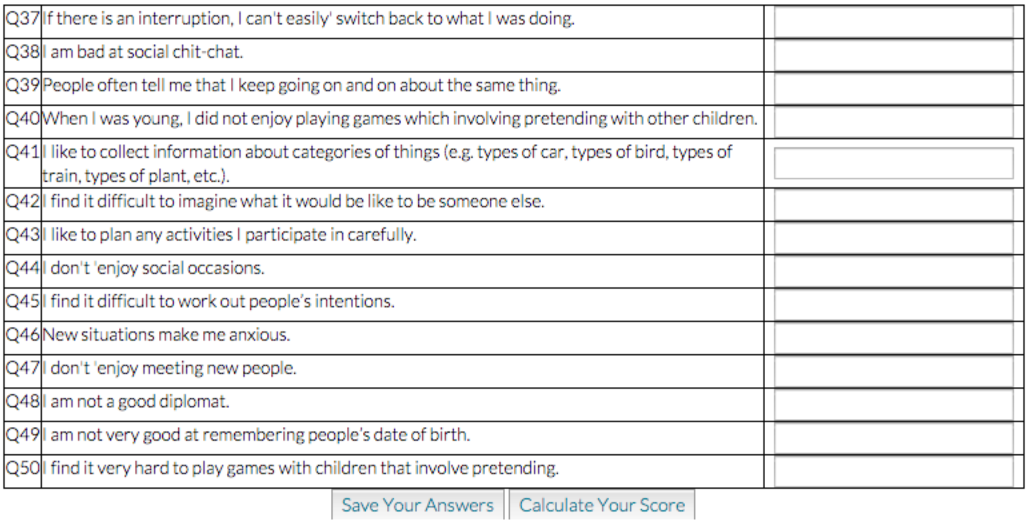
5.4.1 Normal Forms

The database tables were created in 2NF (second Normal Form). The `SCORE` attribute was saved in the User Model with `ID`, `NAME`, `EMAIL`, `PASSWORD`, `SCORE`, `REMEMBER_TOKEN`, `CREATED_AT` and `UPDATED_AT`. This was chosen as the most optimal normal form to limit the number of SQL `JOIN` operators needed to retrieve the user Autism Score (see Figure 24).

```
mysql> select * from users;
+----+----------+--------------------+--------------------------------------------------------------+-------+----------------+---------------------+---------------------+
| id | name     | email              | password                                                     | score | remember_token | created_at          | updated_at          |
+----+----------+--------------------+--------------------------------------------------------------+-------+----------------+---------------------+---------------------+
|  1 | Esha     | e.massand@bbk.ac.uk | $2y$10$U97sq4pgU75q4McP.ZoJ2edb662o3AW4TBOXAwLxROfmz7OcaLLdq |    30 | NULL           | 2015-08-11 00:31:37 | 2015-08-11 00:33:17 |
|  2 | Test     | test@example.com   | $2y$10$2OiD1a4Tl5Itp9WuRNiZ1ecAiWdYxg0oPKjVdt2xYf.U3eq9IP2Q. |    31 | NULL           | 2015-08-12 08:12:47 | 2015-08-12 08:13:44 |
+----+----------+--------------------+--------------------------------------------------------------+-------+----------------+---------------------+---------------------+
2 rows in set (0.00 sec)
```

Figure 24: Using SQL tables to store AQ for each user of Jellibeans (User Table).

The user can also save their data locally to an Excel file by selecting the 'Save Your Answers' button on the Jellibean Homepage if they do not wish to log in/ register with Jellibeans (see Figure 27). Alternatively the user can choose 'Calculate Your Score' which returns the score as an alert to the page.



Q37	If there is an interruption, I can't easily' switch back to what I was doing.	
Q38	I am bad at social chit-chat.	
Q39	People often tell me that I keep going on and on about the same thing.	
Q40	When I was young, I did not enjoy playing games which involving pretending with other children.	
Q41	I like to collect information about categories of things (e.g. types of car, types of bird, types of train, types of plant, etc.).	
Q42	I find it difficult to imagine what it would be like to be someone else.	
Q43	I like to plan any activities I participate in carefully.	
Q44	I don't 'enjoy social occasions.	
Q45	I find it difficult to work out people's intentions.	
Q46	New situations make me anxious.	
Q47	I don't 'enjoy meeting new people.	
Q48	I am not a good diplomat.	
Q49	I am not very good at remembering people's date of birth.	
Q50	I find it very hard to play games with children that involve pretending.	

Save Your Answers | Calculate Your Score

Figure 25: The User Can Choose to Save Their Scores to an Excel File.

5.5 Integration with a Motion Controlled Interface.

The selection process for the hardware to integrate with Jellibeans considered 4 factors; the timing of the device, the 'volume', the applicability to real-world environments and the affordability for users.

Jellibeans is integrated with the LEAP Motion Controller Hardware (see Figure 26) which has good timing as it polls frames at a constant rate to keep to timining of accurate movement, and this is important for user experience. It is very easy to use and only requires a short calibration period. It is non-invasive and has a low sensory experience (sensory stimulation is not tolerated well by individuals with Autism). The cognitive demand needed to operate the LEAP is low, and given that web search is a highly cognitive task, this is of real benefit for the user.

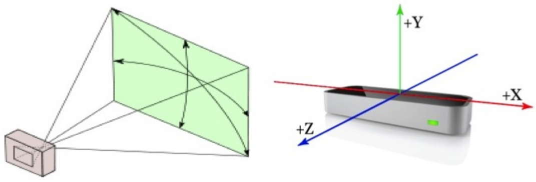


Figure 26: The LEAP controller, with 150 degree view [14].

The LEAP recognises and tracks hands, fingers, finger-like tools, positions, motions and gestures using infrared light and optical sensors along the cartesian coordinate system. The controller has a 150-degree field of view, and operates in a range of 1 inch to 2 feet.

Each of our senses operates with a different lag time. Hearing has the fastest sense-to-cognition/understanding, and surprisingly, sight – the slowest. If the device interferes with the processing of the sense, it will confuse the combinatorial configuration of the senses, leading to misunderstandings in the meaning and a worse user experience. The LEAP does not have the best cognitive lag time compared to other devices such as the Oculus VR Rift. However, for Jellibeans, the browser refresh rate will likely be the biggest bottleneck for performance, rather than the motion-controller chosen. The LEAP is a completely non-invasive in comparison to the Rift (see Figure **??**), so it was considered a better option for the current user group.

Figure 27: Comparison of the invasiveness of the Oculus VR Rift and LEAP motion controller.

I also took into consideration the way in which the device manifests actions into behaviours. That is, how does the user engage behaviourally within the environment using the device? The LEAP was considered very realistic in terms of the transition of behaviour into real-world environments.

5.5.1 Using LEAP in the IDE.

Frames (cartesian coordinate data) from the LEAP were traked using the `Frames` class, and detected user motions. The following code was used to implement a forward-poking pointing motion to control mouse clicks. I ran the code to check how well it worked in Eclipse IDE (see https://youtu.be/ikfiul_JPBk for a video of the LEAP working in my IDE).

```
Controller JellibeanController = new Controller();
Frame jellibeanFrame = JellibeanController.frame();

Robot jellibeanRobot = new Robot();

if(jellibeanFrame.fingers.get(arg0).tipVelocity.get(2)>75)//if finger moves forward past a threshold velocity {

        jellibeanRobot(InputEvent.BUTTON1_DOWN_MASK);
        Thread.sleep(200);
        jellibeanRobot(InputEvent.BUTTON1_DOWN_MASK);

        }
```

Figure 28: Pointing and Poke Motion Used to Control Mouse Clicks.

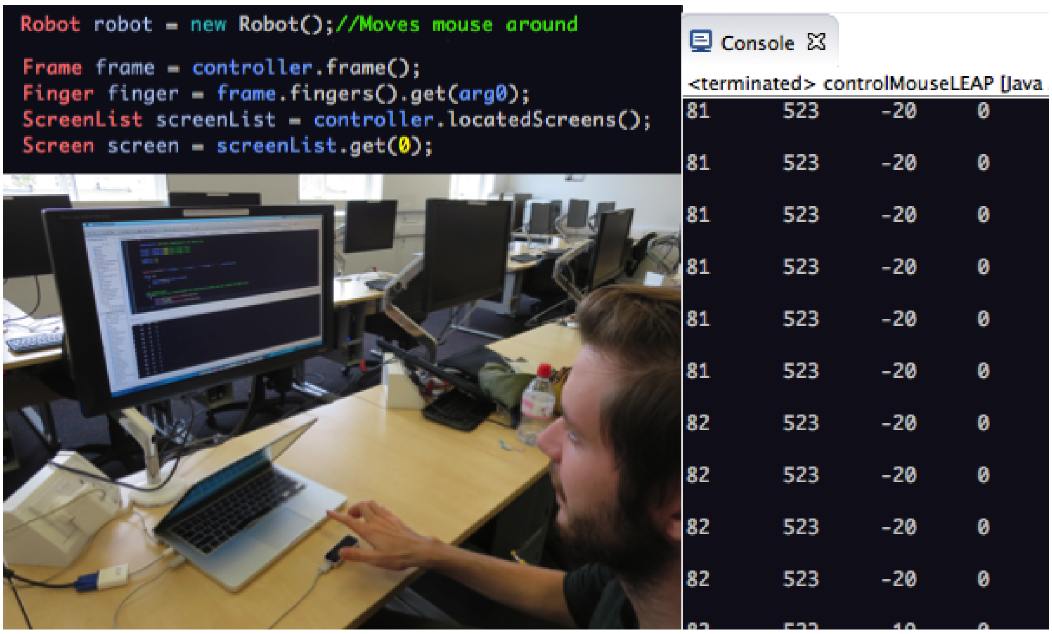


Figure 29: Using the LEAP to Control the Cursor in the IDE. Console Output is Hand Coordinates along the x, y, and z Axes.

Although using the device this way was interesting and novel, the user has unstable control over the cursor, and often hand movements aren't precise enough to have a good experience using the hardware. I decided to look for a library that I could use with the LEAP to overcome this. One such library is LeapStrap which I discuss in the next section.

5.5.2 Using LeapStrap in the Web-browser.

To use the LEAP in a web-browser I used LeapStrap [16], which is built on top of Bootrap and was easy to integrate into the project. To integrate LeapStrap into the HTML I added link in the LeapStrap CSS , jQuery, Leap.js and LeapStrap.js. I also initialised the Leap in the body of the HTML code, and finally made all elements of the class "leap-interactive" (see Figure 30).

```
<!DOCTYPE html>
<html lang = "en">
<head>
<meta charset = "utf-8">
<meta http-equiv = "X-UA-Compatible" content = "IE=edge">
<meta name = "viewport" content = "width = device-width, initial-scale = 1" >
<meta name="google-signin-client_id" content="677279234093-scr4q3jm5ekbgmtqr8eanv61vgurrcut.apps.googleusercontent.com"><
/meta>
<title>Jellibeans V1 Welcome Page</title>


<link href="css/leapstrap.css" rel="stylesheet" media="screen">
<link rel= "stylesheet" href = "css/style.css">
<link href='http://fonts.googleapis.com/css?family=Lato:100,300' rel='stylesheet' type='text/css'>


<script src="https://code.jquery.com/jquery.js"></script>
<script src="js/leapstrap.js"></script>
<script src="//js.leapmotion.com/0.3.0-beta3/leap.js"></script>
<script src="https://apis.google.com/js/client:platform.js?onload=startApp" async defer></script>
<script src="js/jellibeans.js"></script>
<!-- JavaScript specific to this application that is not related to API calls -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js" ></script>
```

Figure 30: LeapStrap Library to Implement the LEAP for Jellibeans.

Once the LEAP was enabled for the web browser I was able to test Jellibeans with users to gather feedback.

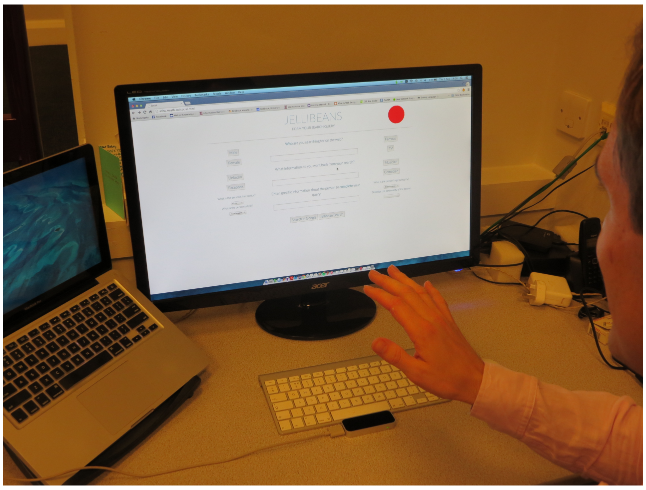


Figure 31: User Forming A Jellibean Search Query with LEAP Using One Hand.

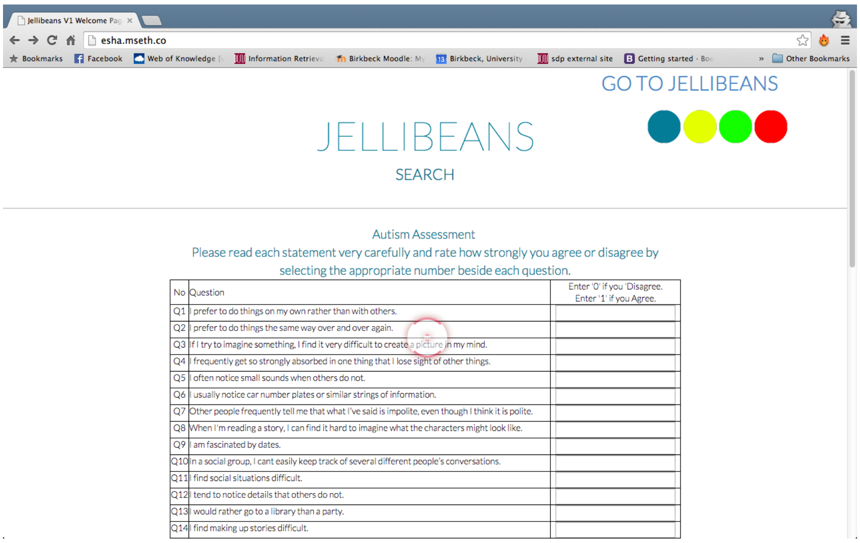


Figure 32: Screenshot of how the LEAP projects a pointing finger onto the Jellibeans. Holding a finger pointer on a button for longer than a second, makes the pointer flash on the screen (indicating a mouse click has occured).

6 Testing and Critical Evaluation of Jellibeans

6.1 Testing and Evaluation of the Combined Search Engine.

To test the integrated combined search engine, 10 participants took part in a study to evaluate the quality of it's output. The participants were selected because they scored highly (above 30, where 32 is threshold for Autism Spectrum Disorder with a 79% sensitivity) on a 50-item screening measure of Autism-like tratis (the Autism Quotient [1]), and these individuals were thought to be most suited to test the current system.

The range of scores for the AQ is 0 to 50 with high scores indicating increased likelihood of autism-like traits. A score under 21 is a low to average result (many women average around 15 and men around 17). A score of 22-25 indicates autistic tendencies slightly above the population average. A score above 26 gives a borderline indication of high functioning autism, or Aspergers. A score above 30 suggests a likelihood of Aspergers syndrome or autism (sensitivity of test measure = 79% [1]). For the purposes of this study, individuals with scores equal to, and above 30 were interpreted as having 'autistic-like traits'.

Participants were asked to comment on their search results from 10 (pre-determined) informational needs. Participants were also asked to choose three out of the links returned that they would choose to follow up with. They were also asked if anything was odd about the results returned. The responses from the 10 users were then analysed.

Somewhat non-surprisingly (given the statistics of most preferred search engines (Google's unique monthly visitors: 1,100,000,000 and query volume: 64.5% ; Bing: 350,000,000, 12.8%; and Yahoo!: 300,000,000, 64.5% ([9] [24])), the results revealed Bing Search was favoured the least, and Google results the most, with Yahoo falling somewhere in between. Out of the 30 responses participants indicated to follow up with (3 per participant), 21 were Google results, 3 were Yahoo, and 1 was Bing and 5 results overlapped between Google and Yahoo. Four participants commented specifically, that the Bing results were distracting rather than helpful.

Given these findings from the user group, future iterations of Jellibeans will continue using the Google results, but drop the results from Yahoo and Bing, in line with the aim of the project as a whole, that is, to improve returned search results for users with Autism.

6.2 Evaluation of the LEAP Motion Controller in Jellibeans.

The LEAP motion controller is an impressive device that uses infra-red light to embed the users (phantom) hands onto the screen. It is a novel, 'at home' technology, and because of the portable nature of the device, it is one that has been introduced to laptops. The LEAP offers a recalibration process if the controller is persistently jumpy, which is good because jumpy movements and mannerisms could be an issue for some of the users in the current user group. The device will recalibrate if there are discontinuities or aberrations in the tracking data in certain field of view, or in poor tracking ranges.

The LEAP does however miss small hand movements, and also very large ones (outside of the 150 degree angle along the y axis (see Figure 26)). The LEAP also neglects to track the bottom left and right-most corners of the screen. I have tried to take this into consideration when designing Jellibeans, so that the user does not need to reach to the corners of the screen.

7 Software Testing

I used Behaviour Driven Development (BDD) to test the integration of my HTML, CSS and JavaScript code. The testing process consisted of development, test, make modifications, test (and so on) until the behaviour of the website was what I intended.

The JavaScript Console was used for testing JavaScript code. [19], I also used JSfiddle [20] to ensure the code contained no bugs, and was particularly good for seeing how the JavaScript code integrated with the HTML code and CSS.

JUnit was used to test the Combination Search Engine and the LEAP motion controller methods, which were written using Java. Test Driven Development unit testing was completed, at the level of each method implemented.

8 User Verification and User Testing

Five high AQ scorers were called back from the previous research study for one-to-one interviews and a combined focus group. During the interview, individuals were given the same user search queries as previously, and for each search that was implemented, users were asked to comment on whether it enhanced, or took away from their search experience, and from their satisfaction with the search results.

8.1 Interviews

8.2 Quantifiable results

The 5 participants were asked to re-construct the same search queries they had been asked to construct in the first research study, using the search engine they identified as their usual choice (all participants said they would usually Google), and also using Jellibeans.

Participants were asked to comment on:

- Whether they found a result they would be satisfied with using both search tools, Jellibeans only, Google only, or neither.
- Which search tool gave them the most relevant results overall, and how many were relevant (focusing only on the top 3 links for comparison purposes, as Jellibeans only returns 3 results.)

The search strings that these users formulated are presented below.
Search strings formed by users with Autism, without using Jellibeans.

1. You hear a song on the radio with the lyrics, 'Look at your children', and you want to download it. What would you type into search on your favourite search engine to find out what song it was?
   Song lyrics look at your children
   look at your children
   Look at your children song
   "Look at your children" lyric
   Look at your children
   "Look at your children" lyrics
   Look at your children song
2. You've lost touch with an old school friend (you went to St. Mary's School). What key words/queries would you use to find them?
   **Persons name* and St Mary's school on Facebook*
   Full name of friend, St Mary's School, year of graduation (i.e. 'class of')
   The friend's full name and "St. Marys"
   Me and St Marys
   I wouldn't want to know. I wouldn't bother
   St Mary's school with the year we left
   Google their name
3. How would you identify what this is using a search engine (pretend you don't know what it is called). What key words/queries would you use?

Google image search, or failing that: dried herb star 8 points
Star shaped seed pod
Star eight spice wikipedia
Star shaped flower
Wooden flower
Star shape hard shell
Post a picture on Facebook
No answer

4. How would you find out the name of this famous person using a search engine? What key words/queries would you use?
Pale brunette brown(or hazelnut) eyes freckles natural beauty earring
Not something I would search for
Model actress long brown hair, brown eyes, big lips
Celebrity female young long dark hair earring
I wouldn't expect to be able to find out their name unless I had more information than just a physical description
Famous females
Pretty white brunette

5. How would you identify what breed this animal is using a search engine? What key words/queries would you use?
Small dog breeds terriers
Small dog fluffy head moustache breed
Dog breed grey furry miniature
Yorkie pooh
Small pet dog
Small dog fluffy head moustache breed
'Small dog breeds'

6. Your friend and you can't agree on how Thandie Newton pronounces her first name. How would you resolve this using a search engine?
Interview with Thandie Newton
Pronounce 'Thandie Newton'
How to pronounce Thandie Newton
Thandie Newton pronunciation
'Thandie Newton pronunciation'
Thandie Newton
YouTube Thandie Newton

7. What would you search for to identify this pattern's name, and which country it originates from?
Some combination of: tiling symmetry
Tessellating "interlocking s"

Square spiral repeating pattern name origin
Square spiral originates from
Repeating square patten
5 part wave pattern square
Celtic patterns

8. How would you search for delay's relating to your (imminent) flight to Paris?
   No answer given
   Airport website
   Flight number plus the date and time
   Aeroplane
   Departures from airport X
   Search for confirmation email in my inbox
   Paris departure Heathrow

Search strings formed by users with Autism, <u>with Jellibeans</u>.

1. You hear a song on the radio with the lyrics, 'Look at your children', and you want to download it. What would you type into search on your favourite search engine to find out what song it was?
   Name of song song name Look at your children Music
   Song lyrics look at children Listen
   Lyrics for song look at your children
   Song name look at your children song
   Name of song name of song look at children

2. You've lost touch with an old school friend (you went to St. Mary's School). What key words/queries would you use to find them?
   School friend name st marys school Female Brown hair Middle aged
   Name of friend from st marys name class of x Female
   Friend name st marys school Female Middle aged
   Name st mary school Male Facebook black hair Middle aged Modest
   Name of friend from st marys name

3. How would you identify what this is using a search engine (pretend you don't know what it is called). What key words/queries would you use?
   Name of star shape brown shell edible item
   Wikipedia name of star eight spice
   Dried herb star eight spokes and spice
   Organic brown star name flower
   Name of organic spice wikipedia natural wooden

4. How would you find out the name of this famous person using a search engine? What key words/queries would you use?
   Famous lady with brown hair name Brown Young Chic Sexy

Brown eyes woman Famous Female Brown Young Modest Moody
Actress pale brunette natural Female Brown Middle aged
Celebrity female long hair dark earring name Famous Female Brown Middle aged Serious
Pretty white brunette name Famous Female Brown Young Chic

5. How would you identify what breed this animal is using a search engine? What key words/queries would you use?
   Name of small white fluffy dog name
   White dog small poofy hair name
   Name of dog white small yorkie
   Grey and white dog small name of dog
   Pet dog name name white small moustache breed

6. Your friend and you can't agree on how Thandie Newton pronounces her first name. How would you resolve this using a search engine?
   Thandie Newton interview YouTube
   How to pronounce Thandie Newton name pronunciation
   Pronounce Thandie Newton name
   Thandie Newton's name how to say it
   Thandie Newton pronunciation YouTube

7. What would you search for to identify this pattern's name, and which country it originates from?
   Name of pattern country of origin name blue spiral
   Origin of pattern country name repeating pattern spiral Wikipedia
   Name repeating pattern symmetry tile Wikipedia
   Tessellating interlocking s pattern origin Wikipedia
   Repeating pattern tile symmetry Wikipedia name of pattern

8. How would you search for delay's relating to your (imminent) flight to Paris? No answer given
   Flight number information Ryan air to Paris Homepage
   Departures from London to Paris airlines information delays
   Aeroplane flight delays paris departure Heathrow
   Departure to paris flight delay information
   Airport website flight delay to Paris

8.2.1 Result 1

The first question was whether participants found results they would be satisfied with using both search tools, Jellibeans only, Google only, or neither.

- For each question the defining criteria of a succesful search was defined according to the base model (from typically developing individuals):

  A reference to the ‘name of the song’ / ‘lyrics’ and ‘Look at your your children’.

  A reference to ‘St Marys School’, [year of graduation] and a description of the person.

  Reference to the ‘star’ shape, the colour ‘brown’, and the fact it was a ‘plant’/‘organic’.

  Reference to a famous woman with brown hair and young.

  Reference to a ‘small’ ‘dog’, and ‘fluffy’ breed.

  A query with the keywords ‘Thandie Newton’ and ‘Pronunciation’

  A reference to a ‘square pattern’ which is ‘repeating’ (‘tiles’ or ‘spiral’ interchangably used), with ‘origin’.

  A reference to ‘Paris’, ‘airport/aeroplane’, ‘flight time/information’.

A categorical analysis was conducted to validate the search efficacy of Jellibeans. There 40 queries in total. Participants were asked to consider the top 3 results from Google (for comparison with Jellibeans), and to explain if their search was satisfied by Jellibeans (only), their other search tool of choice (only), both, or neither. The graph below demonstrates that Jellibeans has a good precision rate (30%) compared to other search tools (25%), see asterix Figure 33, although this was not a significant difference (see Figure 34).

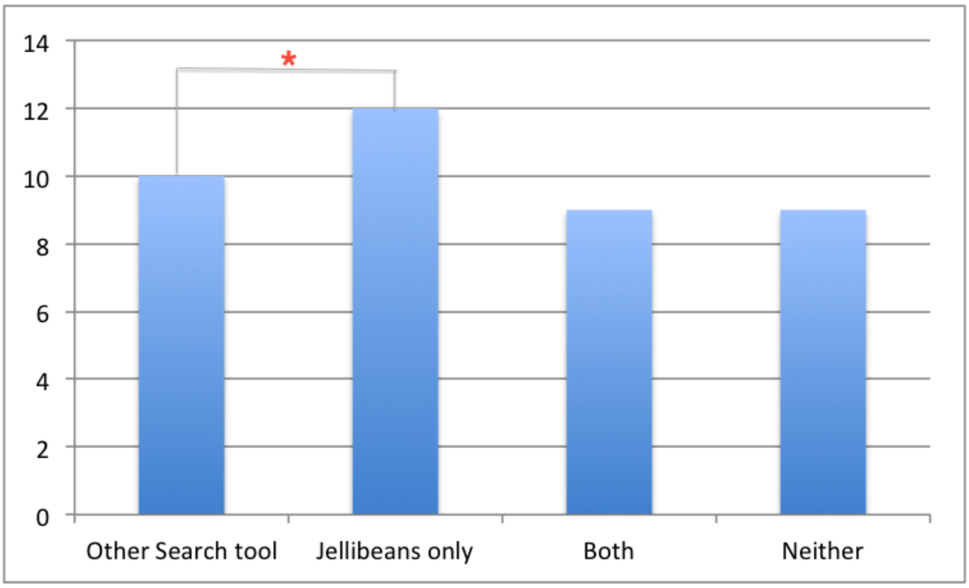


Figure 33: Y axis represents the number of search queries correctly retrieved (out of a possible total of 40). X axis shows number for each search tool, both or neither.

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	OtherSearchToolOnly - Jellibeans_Only	-.050	.749	.118	-.290	.190	-.422	39	.675

Figure 34: No significant difference in precision of Jellibeans compared to other search tools, when considering only the top three results.

The full data file can be found in Appendix 10.6.

This means that Jellibeans is successfully reducing the number of results the participant has to sift through - down to just 3 results - without reducing precision of the search tool within those items.

8.2.2 Result 2

Participants were asked to count the number of results (in the top three) that they felt were relevant to the search query. These data are presented in Figure 35. The results show that the number was comparable (albeit variable) between Jellibeans and the usual search tool for the participant.

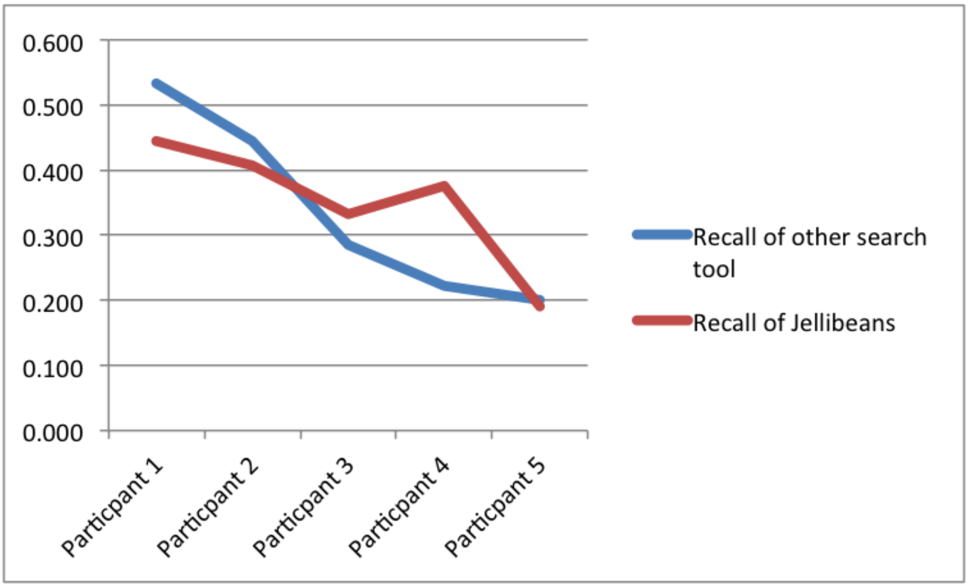


Figure 35: Number of queries considered relevant for users: Jellibeans versus usual search tool (for each participant).

8.2.3 Result 3

There was a significant difference in the number of words used to form search queries when comparing Jellibeans to other search tools. Significantly more words were used to form search queries using Jellibeans (mean number of words in Jellibeans = 3.53, mean number of words in other search tools = 6.55). A paired-samples t-test was run to confirm this observation ($t = -7.79$, $df = 39$, $p < 0.001$, see Figure 36). This is interesting because it means that participants used more words in their search query, which in turn narrows down the scope of their search. This is useful for increasing precision scores of the Jellibeans search tool.

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	No_Words_in_Other_Search_Tool - No_Words_in_Jellibeans	-3.025	2.455	.388	-3.810	-2.240	-7.794	39	.000

Figure 36: Result of t-test comparing number of words used to form search queries. Participants used more words to define their queries using Jellibeans compared to other search tools.

There was also more repetition in words used to form search queries in Jellibeans. This is probably due to the nature of the search process using this tool – there are more search inputs to complete and drop down input boxes (to select 'keywords') which are added to the search in Jellibeans. To remove this confound from the analysis, keywords that were repeated were dropped from any calculations.

This suggests that individuals are defining their search queries better, and using more keywords in Jellibeans.

8.3 Focus Group

During the focus group users were asked to comment on:

1. Whether the colour coding for each category of search was helpful.
2. Whether they liked the parametric search style of Jellibeans
3. Whether they felt Jellibeans 'narrowed' the scope for irrelevant detail.
4. If they would choose Jellibeans over other search engines, for example, Google, and why.
5. The speed and efficiency of search with Jellibeans
6. Their thought on search results.
7. The integration of Jellibeans with the LEAP motion controller.

8.3.1 The Positives

All 5 users verified that the 'Dashboard' colour coding was helpful (see Figure 38), and liked that the colour of the chosen type of search remained on screen for its duration.

Figure 37: Colour coding on Jellibean dashboard.

The modals (floating window reminders) for each of the colour themes were useful for all new users, as they were originally unfamiliar with the colours. The associations for all user groups, including users with Autism, were quick to learn, these were single link visual (rather than verbal) associations.

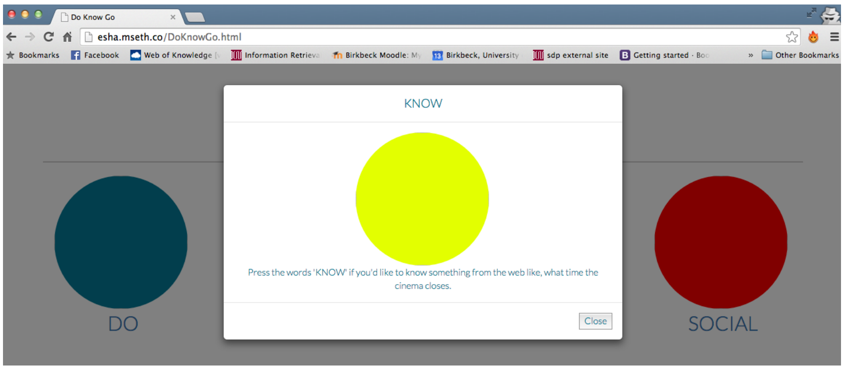


Figure 38: Modals Assist New Users To Lean the Colour Associations.

Four out of 5 participants agreed that one of the major positives of Jellibeans was that unlike Google, users did not receive back hundreds and thousands of mostly irrelevant results. Jellibeans held their interest for longer because of the reduced number of results that users had to sift through. Testing also revealed that the parametric search style meant that they did not need to try and remember all the key words to search for, rather the cues helped them remember points they would have missed otherwise.

A common comment (3 out of 5 users) was the time/reward payoff from completing a detailed search form. Users agreed that although they would spend more time formulating their search query, they were likely to save time that would usually be spent sifting through search results. These participants indicated that they would use Jellibeans over other search engines if continued use on other queries also showed such benefit.

All users positively commented on the reduced number of search results presented (top 3). The amount of textual information that was presented back on the results page is significantly reduced in Jellibeans.

During the focus group one user with high functioning Autism described that the direct questions (e.g., ‘What do you want to do on the web?’), with suggestions (‘Shop’, ‘Watch’, ‘Listen’), was useful when he was “at a total blank”. He disclosed that often his need for perfectionism and sameness could have him staring at a blank screen until he came up with the ‘perfect’ query. However, the suggestions were helpful for him to get started, and so his experience was that the queries were completed faster than usual.

8.3.2 The Negatives

Two users however, had considerations about the 'missing' results from their search and commented that for them, 3 results were too few and perhaps 5 or 6 results would have been a better choice. It must be said that these individuals were amongst the lowest AQ scorers, suggesting that the amount of text individuals with Autism find optimal is a function of the severity of Autism traits in that individual.

One participant commented that the suggestions that appear on the sides of the 'Do', 'Know', 'Go' and 'Social' pages were not user-specific (see Figure 39). He suggested implementing something that is user-centred, user-driven, or, trending, rather than static.

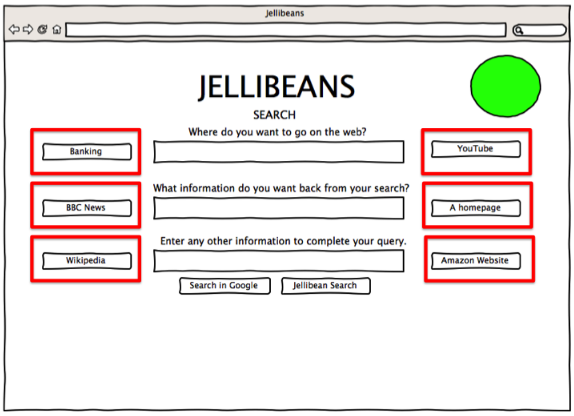


Figure 39: Suggestions on Jellibeans 'Go' page for users to help with search query formulation (suggestions are highlighted in red).

However, one of the characteristics of Autism is restricted and repetitive behaviours, and an aims of the current project was to reduce the restricted and repetitive nature of search queries for people with Autism. In other words, this comment was suggesting that Jellibeans was achieving the aim it had set out to do.

Nevertheless, one future direction for Jellibeans in order to achieve a balance between user-specific suggestions but not restricted interests, is to include a recommendation system that changes often enough for them not to be repetitive, but still within a range of possible interests for that individual.

8.3.3 Other observations

Users agreed during the focus group that with Jellibeans the time/reward payoff is really only apparent, over and beyond a traditional search engine, when the search query is complex. For example, if users are searching for something they have yet to identify, i.e., something ambiguous. To overcome this criticism, Jellibeans includes a 'Search in Google' option, so that if users know what they are searching for, they have the option to run the search outside Jellibeans.

9 Evaluation and Discussion.

Several unexpected challenges and successes occured during the course of the project's execution.

9.1 Challenges

One challenge was encountered when integrating the Apache Lucene Library with Jellibeans. The library was not well maintained and the org.apache.log4j file log was broken. I considered Apache Solr [32], and Apache Nutch [27], but these libraries requires a large document collections to create indices prior to text search. It was not feasible to index the entire document colletion on the web for Jellibeans, and to sift the web *on the fly* would have resulted in a long duration for participant user queries. To overcome this challenge I followed the mitigation protocol outlined in my project proposal that stated I should attempt to implement the methods needed. User testing showed that the implemented solution was senstive enough to retrieve documents that satisfied user search queries.

It was originally thought that user Autism Scores would be stored in their Google+ profile, to be retrieved when users sign in to Jellibeans using Google+. However, Google+ did not allow for a field to persist after the session ended to retrieve the user's score. Conversely, any data stored to Google+ would not be retrievable after the user's session had ended (i.e., after the user logged out of Google+). To overcome this challenge I implemented my own datastore for user's Autism Quotient Scores. The final prototype of the application is therefore not integrated with Google+, but instead the user can log in and out of the Jellibeans website. The users AQ score is saved in the SQL datastore that is integrated with the website (although in earlier prototype models Google+ was integrated with Jellibeans).

To deploy the application, with a database in the web browser, I learned a number of development languages that I had not used before. This included php, JQuery, JavaScript, HTML and CSS. I also worked with an unfamiliar php framework called Laravel[22] to deploy the application in the browser. Although the prototype of Jellibeans (at esha.mseth.co) is implemented using static HTML, I chose to implement the final application using this dynamic website framework. This was one of the biggest personal challenges of the project, and steepest learning curve.

Web Robots created many challenges for the current project. Robots automatically traverse the web to retrieve a document and in a recursive fashion these robots retrieve all documents that are referenced. Although this does not include web searches as these are operated by human queries (i.e, not programatically requested), when Jellibeans was implemented programatically from Eclipse IDE [8], I often received blocked requests in the console. This was because several websites implement `robot.txt` files to block web crawlers, and Jellibeans was being blocked as though it were a web crawler. This was a challenge that was unexpected and originally seemed unsurmountable at the time. After some research it became apparent that to overcome this challenge, I needed to register Jellibeans as a project in the Google Developers Console [15], after which correct privilidges would be assigned.

9.2 Successes

During the course of the project, user testing was conducted on 52 occasions (surveys = 37 participants, feedback on combined search engine = 10 participants and interviews/focus group = 5 participants). Seven of the individuals who participated in the survey had an increased Autism Score, and 5 of these individuals were included in the feedback stages of user testing for Jellibeans.

User testing demonstrated that the user model that Jellibeans implements works well in the search tool. Participants reported a number of positives with the application that shows promise for the future development of Jellibeans.

The LEAP integrated behaviour into the web browser in a very naturalistic way. The movements are smooth and the visual aesthetics of the pointer on the screen are a gret balance of fun, *and* function. The cursor is easy to use even for people with uncontrolled, or, jerky hand movements.

Although not part of the original proposal for the project, and even with a number of challenges along the way, there was time to implement some of the user feedback to drop frustations that users identified during user testing before the final prototype was implemented. Other modifications suggested by users required the purchase of data (such as ‘current trends’) which I have spoken about in Section future.

9.3 Future Directions

There are many avenues to consider for the future of Jellibeans. For example, modifications to the asthetic appeal, user needs, and functionality of the application.

A future aesthetic consideration is to improve the visual design of the website itself. In the current version Bootstrap was used, which helped considerably with visual aesthetics, however this could be further considered along with functionality as well. If I were to redesign Jellibeans, I would not present the Autism Questionnaire on the first page of the application, but instead have this on a separate page. I would also choose to implement an abbreviated version of the test, or a version for parents or carers to complete on behalf of the user (as a 50-item optional questionnaire may not have the best response rate from individuals with Autism themselves).

A future functional direction for Jellibeans would be to use an API or library for word frequencies in the written English language, which could be integrated with a thesaurus. This type of API requires significant research into the natural language processing of written (rather than spoken) English text. These frequency data could have been used to replace infrequent words in queries formed by individuals with Autism with more frequent words to suggest alternatives for the user. Of course this product would not be scalable, and only apply to users with a particular dialect. This is because word frequency is considerably variable across, and even within a language (e.g., consider South-West England compared to the North of England).

Another future direction for Jellibeans would be to implement current trending searches, which could be a contributing factor in the future for why users choose other search engines over Jellibeans.

Individuals with Autism have increased comorbidity with Attention Hyperactivity Disorder (ADHD), and Colour Blindness, and so these can be taken into account for future iterations of Jellibeans. For example, the colours used for Jellibeans are in the RGB colour scheme, but these could be replaced with gradients or patterns so that individuals with colour blindness can also also benefit.

References

[1] Baron-Cohen, Wheelrigjt, Skinner, Martin, Clubley (2001). The Autism-Spectrum Quotient (AQ): evidence from Aspergers Syndrome/high-functioning autism, males and females, scientists and mathematicians. *Journal of Autism and Developmental Disorders, 31*, 5-17.

[2] Bootstrap, http://getbootstrap.com/ Retrieved September 2 2015.

[3] Provos, Niels, MaziÃÍres, David and Talan Jason Sutton (1999). *A Future-Adaptable Password Scheme*. Proceedings of 1999 USENIX Annual Technical Conference: 81âĂŞ92.

[4] Bronwyn, M, Murray, M, and Durkin, K. (2003) Weak central coherence, poor joint attention, and low verbal ability: Independent deficits in early autism. *Developmental Psychology, 39*, (4), 646-656. http://dx.doi.org/10.1037/0012-1649.39.4.646

[5] Bowler, D. M., Gaigg, S. B., Gardiner, J. M. (2008). Subjective organisation in the free recall learning of adults with Aspergers syndrome. *Journal of Autism and Developmental Disorders, 38*(1), pp. 104-113. doi: 10.1007/s10803-007-0366-4

[6] Developmental Disabilities Monitoring Network Surveillance (2010) *Centers for Disease Control and Prevention (CDC). Prevalence of autism spectrum disorders: Autism and Developmental Disabilities Monitoring Network, United States. MMWR Surveill Summ.2009; 58*,

[7] Developmental Disabilities Monitoring Network Surveillance (2010) *Centers for Disease Control and Prevention (CDC). Prevalence of autism spectrum disorders: Autism and Developmental Disabilities Monitoring Network, United States. MMWR Surveill Summ.2009; 58*, 10:120

[8] Eclipse Luna IDE, https://eclipse.org/luna/, Retrieved 27 August 2015.

[9] eBizMBA inc, *The eBusiness Guide*, www.eBizMBA.com; Retrieved 20 March 2015

[10] Executive Function, https://en.wikipedia.org/wiki/Executivefunctions, Retrieved 27 August 2015.

[11] Elsabbagh, M., Volein, A., Holmboe, K., Tucker, L., Csibra, G., Baron-Cohen, S., Bolton, P., Charman, T., Baird, G. and Johnson, M. H. (2009), Visual orienting in the early broader autism phenotype: disengagement and facilitation. *Journal of Child Psychology and Psychiatry, 50*, 637ÃČÂćÃĆâĆňÃĆâĂIJ642. doi: 10.1111/j.1469-7610.2008.02051.x

[12] Fishkin (2015) https://moz.com/beginners-guide-to-seo/how-people-interact-with-search-engines, Retrieved 3 August 2015.

[13] Games for Health (2012) *Screen-based technologies and Autism. 1*: 248-53

[14] Leap Motion, *Java SDK Documentation*, https://developer.leapmotion.com/documentation/java/index.html Retrieved 1 April 2015.

[15] Google Developers Console, https://console.developers.google.com, Retrieved 1 April 2015.

[16] LeapStrap, http://wilkesalex.github.io/leapstrap/getting-started/, Retrieved September 3 2015.

[17] Garzotto, F., Valoriani, M. and Bartoli, L. (2014), Touchless Motion-Based Interaction for Therapy of Autistic Children, Virtual, Augmented Reality and Serious Games for Healthcare, *Intelligent Systems Reference Library, 68*, 2014, pp 471-494

[18] Google Privacy and Terms, http://www.google.com/policies/technologies/, Retrieved 27 August 2015.

[19] JavaScript Console, https://developer.chrome.com/devtools/docs/console, Retrieved September 3 2015.

[20] JS Fiddle, https://jsfiddle.net/, Retrieved September 3 2015.

[21] JSoup Java API, http://jsoup.org/, Retrieved 27 August 2015.

[22] Laravel, laravel, http://laravelbook.com/laravel-architecture/ Retrieved Sep 6 2015

[23] Laurent Mottron, Jacob A. Burack, Johannes E. A. Stauder, Philippe Robaey (1999) Perceptual Processing among High-functioning Persons with Autism. *Journal of Child Psychology and Psychiatry 40*, (2), 203211. doi:10.1111/1469-7610.00433.

[24] Lella, A., (2014). *comScore Releases March 2014 U.S. search Engine Rankings.* ComScore.com. Retrieved 21 Feb 2015

[25] Moore, D. J., McGrath, P., & Thorpe, J. (2000). Computer aided learning for people with autismÃćÂĂÂŤa framework for research and development. *Innovations in Education and Training International, 37*

[26] Novogrodsky, R. (2013). Subject-pronoun use of Children with Autism Spectrum Disorders (ASD). *Clinical Linguistics and Phonetics, 27*(2), 85-93.

[27] Apache Nutch, http://nutch.apache.org/, Retrieved September 10 2015.

[28] OpenSearch, http://www.opensearch.org/Specifications/OpenSearch/1.1, Retrieved 30 July 2015.

[29] Online PHP tester, http://phptester.net/, Retrieved September 7 2015.

[30] Shane, H. C. and Albert, P. D. (2008) Electronic screen media for persons with autism spectrum disorders: results of a survey. *Journal of Autism Developmental Disorders, 38*,8 :1499-508. doi: 10.1007/s10803-007-0527-5.

[31] Slavin, S (2013) Autism: Successful communication,10 things to make it easier, http://adultswithautism.org.uk/autism-successful-communication-10-things-to-make-it-easier/, Retrieved September 1 2015.

[32] Apache Solr, http://lucene.apache.org/solr/, Retrieved September 10 2015.

[33] Surveymonkey, https://www.surveymonkey.com/home/, Retrieved 5 August 2015.

[34] Term Frequency Inverse Document Frequency Weighting, http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html, Retrieved 9 August 2015.

[35] Word Frequency Information, http://www.wordfrequency.info/free.asp?s=y, Retrieved 6 August 2015.

[36] Shen, X., Tan, B. and Zhai, C. (2005) Implicit User Modeling for Personalized search, *Conference on Information and Knowledge Management*, Bremen, Germany.

10 Appendices

10.1 Class Diagram

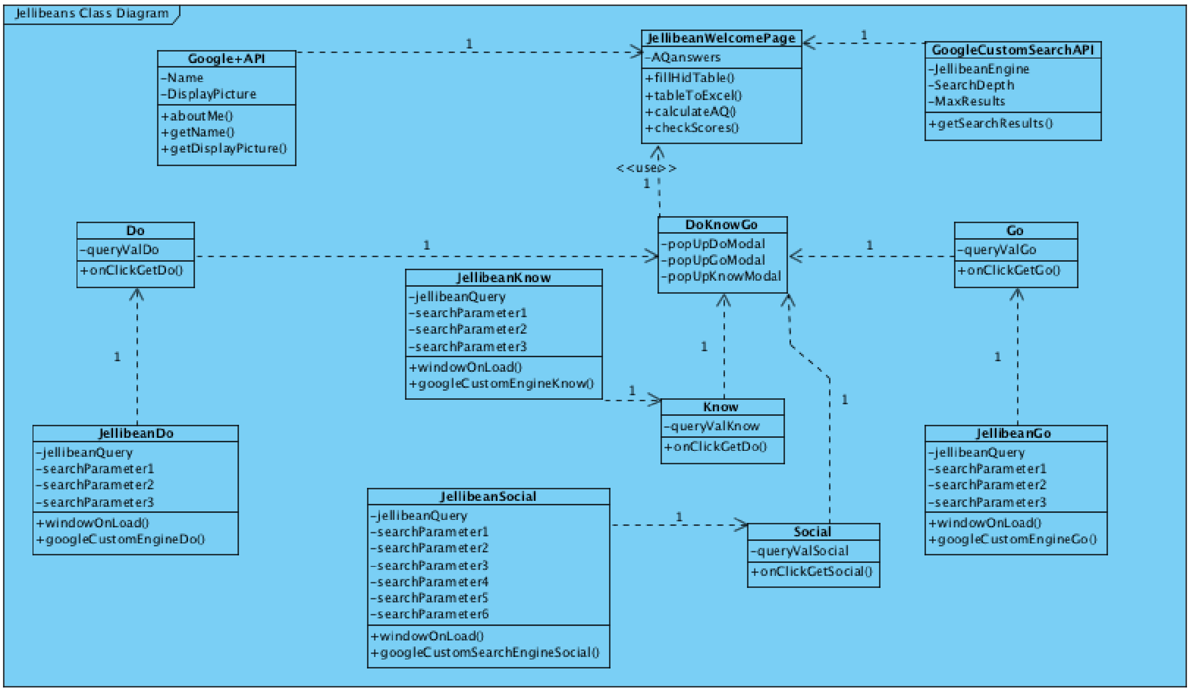


Figure 40: High Level UML Class Diagram of Jellibeans.

10.2 Use Case Diagram

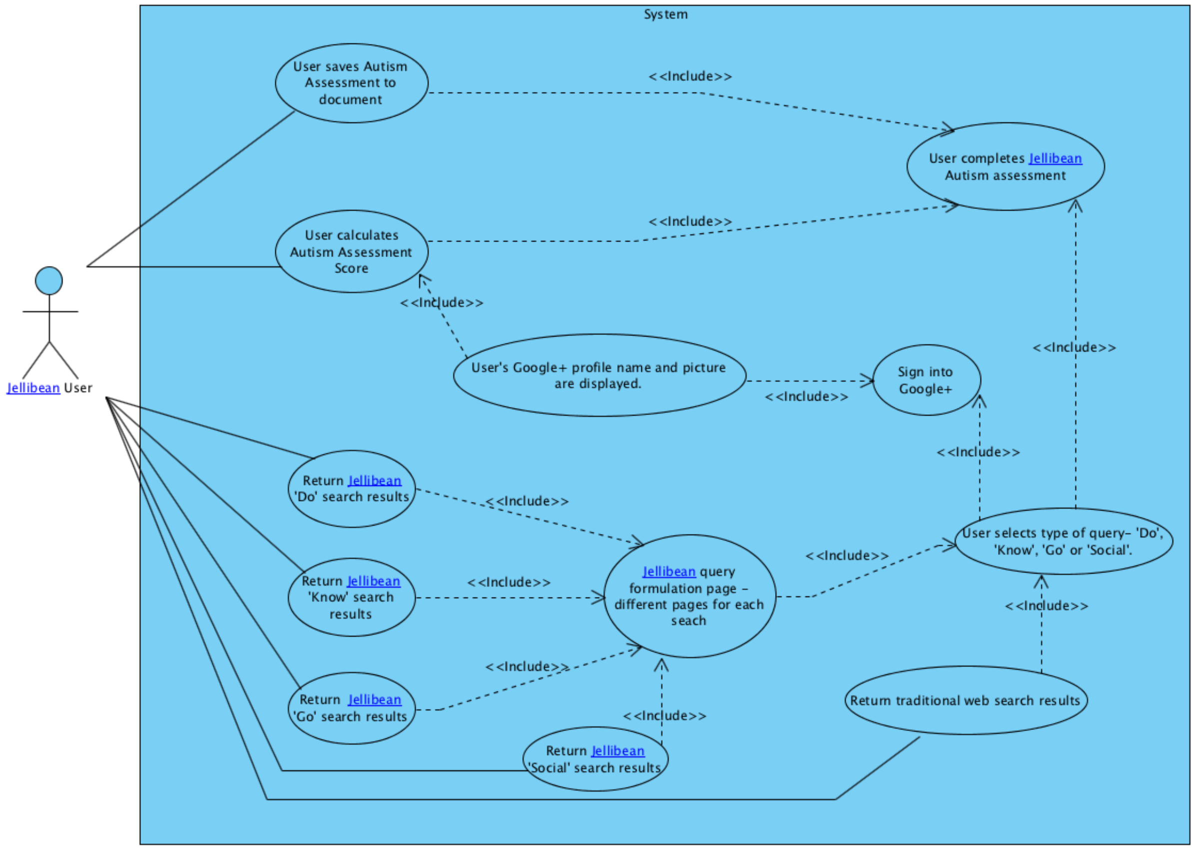


Figure 41: Jellibean Feature Uses and High Level UML Use Case Diagram

10.3 Languages and APIs Used in Jellibean Implementation

Google Search API	A RESTful API with a single method called list. The API method used was GET, and the response data is returned as a JSON type. The response consists of (1) the actual search result, (2) metadata for search like number of results, alternative search queries, and (3) custom search engine metadata. Google terms of service, state that 'screen scraping', or copying the data directly from the website is prohibited.
JSoup API	For Bing and Yahoo search results, JSoup (a Java HTML parser) was used to identify the links from the resulting query. The JSoup HTML parser was considered more efficient for retrieving search results, as it could be used to complete the task from both search engines, using a slightly different href element filter for each. JSoup also has advantages over html parsing. It contains a class representing a list of nodes called 'Elements', which implements Iterable to iterate over a list in an enhanced for loop.
Java	To test the combined output from the three search engines with a group of users, and gather their feedback on the results, I developed a Java Applet to run the programme. The search results from the Java Applet were written to a text file. Java was chosen for development of 'to be tested' packages of features (e.g., the combination search engine). The developer for the project was most proficient with Java, and could quickly develop prototypes for user testing.
JavaScript	JavaScript is commonly used in HTML as it can run locally in a browser (rather than remote server). The Google+ API provided good documentation for using JavaScript. Furthermore the JavaScript code could be easily embedded into the Jellibean HTML code to interact with the Document Object Model (DOM) of the page. The browser was able to respond to user demand quickly, which made Jellibeans more responsive. JavaScript was used in this way to return the aboutMe profile information from Google+ to the user.
HTTP Set-Up	The Hypertext Transfer Protocol (HTTP) enables communications between web-browsers (clients) and the server (the computer that hosts Jellibeans) using a request-response protocol. Jellibeans was deployed in textbfnginx HTTP web server, and digitalocean.com was used to host the web servers, so that it was ready to be used remotely by users (and for me to gather feedback easily). When the user submits a HTTP request to the server that hosts Jellibeans, it responds with the appropriate behaviour to the client so that the user can formulate their search query and retrieve the results from their search.
JQuery	
php	

HTML	HyperText Markup Language (HTML 5) was used to create the webpages for Jellibeans, so that the web browser could render the wesite.
Bootstrap and CSS	For front-end development and to style the Jellibean web pages I used the Bootstrap framework [2] for it's clear styles, good design and ease of use for the current project. I included my own Cascading Style Sheet (css file) where I needed extra styling beyond what the Bootstrap framework could offer.
LEAP SDK	The LEAP motion controller is a light-weight, portable device that detects user motions using infra-red light. The LEAP has accurate timing, and works well with the combinatorial configuration of senses (sight, hearing and touch). As this is a tool to be used with individuals with Autism, who have increased sensitivity to touch, the LEAP is the perfect choice as there is no additional sensations to the users body. The LEAP is affordable for users to integrate with search at home, and currently retails for about $60.
LeapStrap	The LeapStrap SDK was used to integrate the LEAP into the web broswer. LeapStrap is a HTML5 front-end framework for websites, and gives them fill LEAP motion functionality.

10.4 Google Terms of Servive

5. Use of the Services by you

5.1 In order to access certain Services, you may be required to provide information about yourself (such as identification or contact details) as part of the registration process for the Service, or as part of your continued use of the Services. You agree that any registration information you give to Google will always be accurate, correct and up to date.

5.2 You agree to use the Services only for purposes that are permitted by (a) the Terms and (b) any applicable law, regulation or generally accepted practices or guidelines in the relevant jurisdictions (including any laws regarding the export of data or software to and from the United States or other relevant countries).

5.3 You agree not to access (or attempt to access) any of the Services by any means other than through the interface that is provided by Google, unless you have been specifically allowed to do so in a separate agreement with Google. You specifically agree not to access (or attempt to access) any of the Services through any automated means (including use of scripts or web crawlers) and shall ensure that you comply with the instructions set out in any robots.txt file present on the Services.

5.4 You agree that you will not engage in any activity that interferes with or disrupts the Services (or the servers and networks which are connected to the Services).

5.5 Unless you have been specifically permitted to do so in a separate agreement with Google, you agree that you will not reproduce, duplicate, copy, sell, trade or resell the Services for any purpose.

5.6 You agree that you are solely responsible for (and that Google has no responsibility to you or to any third party for) any breach of your obligations under the Terms and for the consequences (including any loss or damage which Google may suffer) of any such breach.

Figure 42: Google Terms of Service

10.5 Search Query Survey

Search Behaviours

Welcome to My Survey

We are trying to understand more about how people with autism, or autistic tendencies, and people without autism search the web.

For the search queries below please indicate what search terms you would use to search the web.
There are no right or wrong answers.
Some of the searches do not have an answer but please tell us how you would go about trying to search for the answer.
If you would do many iterations of search, please indicate that below. Start with the one you are most likely to try.
Please indicate as much as you can about your search process as possible. Use as much space as you need.

There are 10 questions here and this should take approximately 20 minutes or less.
The second part asks you to complete an online questionnaire and report back the score on this sheet.

If you have any questions you can contact e.massand@bbk.ac.uk.

Next

Search Behaviours

An Example Question

EXAMPLE QUESTION:
You saw a man reading the paper today (you don't know which paper it was), and the headline was vaguely about a new discovery about the solar system. What would you type into search on your favourite search engine to find out about the story?

EXAMPLE ANSWER:
I could start by searching, 'solar system and discovery and news and today'.
If that doesn't work I could search, 'todays paper and solar system', or if I don't find my answer there I could search 'news' and manually /visually sift the page for relevant info.

Prev Next

Search Behaviours

1.

What browser do you usually use? (Google Chrome, Internet Explorer, Safari, Firefox ...)

and

What search engine do you usually use? (Google, Bing, Yahoo ...)

Prev Next

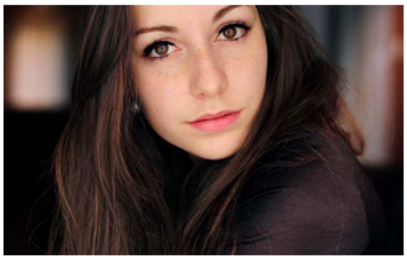
Search Behaviours

2. You hear a song on the radio with the lyrics, ”Look at your children”, and you want to download it. What would you type into search on your favourite search engine to find out what song it was?

3. You’ve lost touch with an old school friend (you went to St. Mary’s School). What key words/queries would you use to find them?

4.

How would you identify what this is using a search engine (pretend you don't know what it is called). What key words/queries would you use?



5.

How would you find out the name of this famous person using a search engine? What key words/queries would you use?

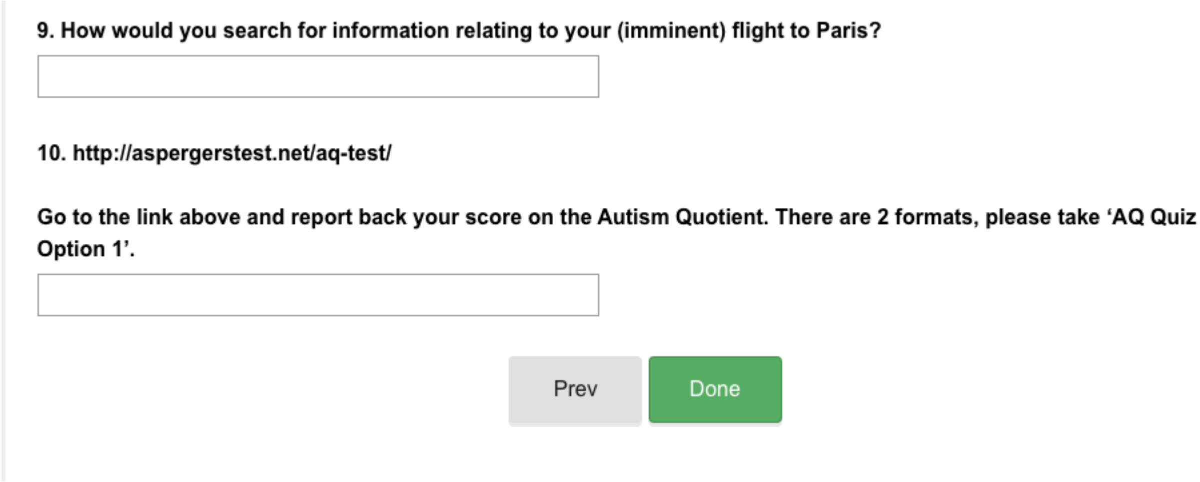
6. Two repeats shown

What would you search for to identify this pattern's name, and which country it originates from?

7.

How would you identify what breed this dog is using a search engine? What key words/queries would you use

8. Your friend and you can't agree on how Thandie Newton pronounces her first name. How would you resolve this using a search engine?



9. How would you search for information relating to your (imminent) flight to Paris?

10. http://aspergerstest.net/aq-test/

Go to the link above and report back your score on the Autism Quotient. There are 2 formats, please take ‘AQ Quiz Option 1’.

Prev Done

Figure 43: Search Query Survey on Surveymonkey.com [33]

10.6 Data Coding for Result 1 and 2

Participant	Query#	# words in query with normal search engine	# words in query with Jellibeans	Google Only	Jellibeans only	Both	Neither	How many out of top 3 were relevant OTHER	How many out of top 3 were relevant JELLIBEANS
1	1	6	8	1				1	
1	2	3	6		1				1
1	3	5	7	1				2	
1	4	8	8	1				2	
1	5	4	6		1				2
1	6	3	4	1				2	
1	7	2	7	1				1	
1	8	0	7		1				1
2	1	4	6		1				1
2	2	5	8			1		1	1
2	3	4	5				1	0	0
2	4	0	8			1		2	2
2	5	6	6			1		3	3
2	6	3	7			1		2	2
2	7	3	8		1				1
2	8	2	7				1	0	0
3	1	5	6	1				1	
3	2	3	8		1				1
3	3	3	6	1				0	
3	4	9	7			1		2	2
3	5	5	6	1				1	
3	6	3	4			1		1	1
3	7	6	6	1				1	
3	8	4	6				1	0	0
4	1	5	6			1		2	1
4	2	2	11		1				1
4	3	3	5		1				1
4	4	7	11		1				2
4	5	2	5				1	0	0
4	6	3	7			1			2
4	7	4	6			1			2
4	8	4	6				1	0	0
5	1	4	5				1	0	0
5	2	0	5				1	0	0
5	3	2	6				1	0	0
5	4	0	9		1				1
5	5	3	7		1				2
5	6	3	4		1				1
5	7	3	7	1				3	
5	8	0	5				1	0	0

Figure 44: Data File for Results 1 and 2

10.7 Questions on the Autism Spectrum Quotient [1]

Participants are asked to read each statement very carefully and rate how strongly they agree or disagree with the statement (Strongly Disagree, Slightly Disagree, Slightly Agree, or, Strongly Agree).

I prefer to do things with others rather than on my own.
I prefer to do things the same way over and over again.
If I try to imagine something, I find it very easy to create a picture in my mind.
I frequently get so strongly absorbed in one thing that I lose sight of other things.
I often notice small sounds when others do not.
I usually notice car number plates or similar strings of information.
Other people frequently tell me that what I've said is impolite, even though I think it is polite.
When I'm reading a story, I can easily imagine what the characters might look like.
I am fascinated by dates.
In a social group, I can easily keep track of several different people's conversations.
I find social situations easy.
I tend to notice details that others do not.
I would rather go to a library than a party.
I find making up stories easy.
I find myself drawn more strongly to people than to things.
I tend to have very strong interests which I get upset about if I can't pursue.
I enjoy social chit-chat.
When I talk, it isn't always easy for others to get a word in edgeways.
I am fascinated by numbers.
When I'm reading a story, I find it difficult to work out the characters' intentions.
I don't particularly enjoy reading fiction.
I find it hard to make new friends.
I notice patterns in things all the time.
I would rather go to the theatre than a museum.
It does not upset me if my daily routine is disturbed.
I frequently find that I don't know how to keep a conversation going.
I find it easy to read between the linesÂÎ when someone is talking to me.
I usually concentrate more on the whole picture, rather than the small details.
I am not very good at remembering phone numbers.
I don't usually notice small changes in a situation, or a person's appearance.
I know how to tell if someone listening to me is getting bored.
I find it easy to do more than one thing at once.
When I talk on the phone, I'm not sure when it's my turn to speak.
I enjoy doing things spontaneously.

I am often the last to understand the point of a joke.
I find it easy to work out what someone is thinking or feeling just by looking at their face.
If there is an interruption, I can switch back to what I was doing very quickly.
I am good at social chit-chat.
People often tell me that I keep going on and on about the same thing.
When I was young, I used to enjoy playing games involving pretending with other children.
I like to collect information about categories of things (e.g. types of car, types of bird, types of train, types of plant, etc.).
I find it difficult to imagine what it would be like to be someone else.
I like to plan any activities I participate in carefully.
I enjoy social occasions.
I find it difficult to work out people's intentions.
New situations make me anxious.
I enjoy meeting new people.
I am a good diplomat.
I am not very good at remembering people's date of birth.
I find it very easy to play games with children that involve pretending.