Travel Search: Extreme Search – Searching for Destinations

LUQMAAN SALIE, University of Cape Town
SHUAIB PARKER, University of Cape Town
DYLAN HENDERSON, University of Cape Town
NGONI CHOGA, University of Cape Town

1. INTRODUCTION

Search engines are very good at simple lookup searches. They provide accurate results to people who know what they are looking for. However, users who have fuzzy information requirements are not presented with decent options when using these typical search engines.

This limitation in search capabilities is a frustration to users who are not exactly sure of their information needs. Many researches such as Marchionini [2006] recognise this problem, and call for the need of exploratory search systems. Exploratory search systems allow users to explore the data available to them, without knowing exactly what they want to query.

This is true for travel search engines, since most of these search engines only allow users to search for flights if you supply the engine with very specific information, such as the desired arrival city.

This is the exact predicament that a flight booking company called Travelstart finds itself in. Travelstart's current website allows users to search for destinations by specifying the exact departure and arrival city. However, Travelstart want to extend the search functionality of their website to cater for customers with unclear information needs.

2. PROBLEM STATEMENT

People often want to travel to a certain place for a holiday or special occasion without having a clear idea of where they want to go. Modern travel search engines, such as Travelstart's search engine, are good at catering for users who know where they want to travel to, but are not good at catering for these users who are unsure of their destination. Thus, users who only have a clear budget with no set destination do not receive optimal search results. This eventually leads to the customer getting frustrated and deciding to use a different service to solve their problem.

3. PROJECT DESCRIPTION AND REQUIREMENTS

We are required to create a search engine for Travelstart which caters for users who are unsure of where they want to travel to, but only know a few pieces of information about their potential destination. For example, if the user only knows that he or she wants to travel to an island for under R8000, the system should be able to support this query and return relevant results.

The system will be embedded in a website, which means we are also required to access Travelstart's current API and design a new user interface in order to incorporate this new search engine. The system needs to be as efficient and user-friendly as possible because the system will be used by a large number of end users.

Author's addresses: L. Salie, S. Parker, D. Henderson, and N. Choga, Department of Computer Science, University of Cape Town.

4. RELATED WORK

Some design recommendations for facets have been stated by Hearst et al. [2002]. They found that usability testing needs to go hand in hand with the technology used for the design of the interface. Hearst found that users like fly-away menus for hiding facets and prefer columns of labels over lists. Colour should also be used to differentiate website components without confusing the user.

One of the major issues found when implementing faceted search systems is a conflict between user queries and stored facets. This is because users do not understand the way in which the information has been stored. One recommendation [Suominen et al. 2007] is to approach the problem from a user-centric point of view, where facets are found for specific users and then these facets are mapped to categories.

Other related work includes a semantic search that was introduced by Cheap Air, which allowed users to make queries with cities and dates. For example: “Chicago to LA on Oct 5th, back on the 10th.” This is similar to what we are trying to achieve but is different in that it is not as fluid in what the users can type to get results.

Salton et al. [1983] founded an information retrieval model called extended boolean model. The traditional boolean model treats search terms as boolean expressions, whereas the extended boolean model represents a document and a query as n-term vectors. It then takes the inner product of these vectors, and normalises it to create a "rank" value. This value represents how closely the search query matches the document. In this way, if a document only matches a few of the search terms, it will be returned as a result. We will be using the extended boolean model in our search engine.

Query expansion is a technique used to enhance the operations of information retrieval, which can be applied to different languages and objects [Bernardi et al. 2006; Chum et al. 2007; Chum et al. 2011; Croft et al. 2010]. Between the old techniques such as global analysis, local feedback and local context analysis, local context analysis is the most proficient method [Croft et al. 1996]. Newer methods using query logs, expansion based on semantics, real-time query expansion, and manually created lexical database expansion have also been implemented [Cui et al. 2002; Fang 2008; Massoudi et al. 2011; Navigli et al. 2003; Pal et al. 2014]. We will implement local context analysis and some of the newer methods to compare effectiveness.

5. SYSTEM FEATURES AND PROCESSES

The system will be developed as a website and used through modern web browsers. The main feature of the website would be a text based interface that will allow users to type in complex queries which will be interpreted by the system. The step by step process the system will go through is described in figure 1 below.

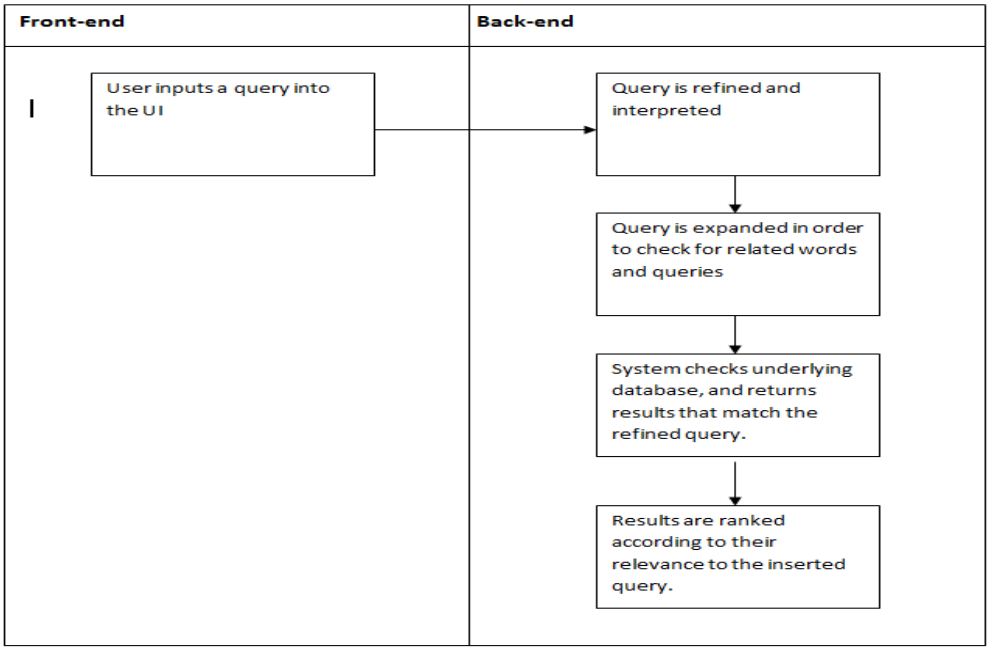


Fig 1. System and subsystems of the project.

With these processes in mind, we can break down the creation of the system into four separate subtasks. Although some of these tasks are performed sequentially, the subsystems required to perform each of these tasks can be developed independently by using dummy data to replace the actual data required.

5.1 Faceted Database Creation

The flight data needs to be stored in a new database that will allow extra information to be associated with destination. The data needs to be grouped into facets such as “fun” and “romantic” in order to enable the search functions to pull relevant results. The system needs to access pictures for destinations as well as group destinations to specific keywords.

5.2 Query Formulation and Expansion

Once the user enters the query, the query will be transformed into a structured query that can be accepted by the search engine. Query formulation will fix any spelling errors if possible, perform stemming and normalisation on query terms, and omit stop words. Thereafter, query expansion will take place to increase recall. Terms from the original query will be expanded using similar words from the database.

5.3 Ranking of Search Results

Results need to be ranked and given a weight in order to determine the order in which they appear to the user. This process needs to be carefully optimized, since the flight database is fairly large, and users typically expect results to be obtained near instantaneously.

5.4 User Interface Design

An intuitive user interface needs to be designed and developed in order to ensure that users find it easy to use the website.

6. METHODS

6.1 Overall project methodology

We will be using Agile methodologies during the project. This means that we will work on designing a working system quickly, testing it, and then changing it based on the client's feedback.

The main reason for choosing Agile instead of the waterfall method is the fact that our task does not have a clearly defined set of requirements. If we design a single version of our system, we might not give the client a desirable product. Agile also provides our team with more client input since we will be able to show them our iterations of the product and make changes as necessary. Since testing will occur at the end of each sprint, bugs in the system will be found earlier. Agile ensures easy adaptation to unforeseen problems, which is difficult with the rigid waterfall method.

6.2 Subsystem implementations and testing plans

Faceted Database creation

Faceted search is a system that allows users to change as well as narrow results from a query without re-entering it. The results of a faceted search are indexed along multiple taxonomies and then displayed to the user in the form of categories. The fact that these facets exist in multiple taxonomies is quite useful as it can lead to complex search queries. The end user has no knowledge of the way the data they are viewing is stored and interacts with the data based purely on the given search categories. Data regarding flight information will be accessed directly from the Travelstart API. Destination information is, however, largely unavailable. A new database or UML/REF file will need to be created to store this information. If Travelstart agree, this information can be stored as an extension to their API. Destinations will be grouped into facets, which include categories such as Beach, Romantic, Adventure and family fun. API's such as Trip Advisor which already groups destinations into such categories will be used, and later, we will allow users to enter their own categories for destinations. Images will be requested at query time from sources such as DBpedia on specific destinations.

A testing suite such as Mocha will be used for testing the system. One of the major aspects of testing is to check that data in the front end and from other retrieval sources is mapped consistently to stored data. All transactions that take place in the database will be tested according to the ACID principles, transactions will either fail or pass, only complete and valid data will be saved to the database, transactions will not affect each other and any data that is saved is never lost. Above this, internal methods will be tested against expected values.

Query formulation and expansion

The query entered by the user will most likely be a natural language query or a query based on categories selected. After the user enters a query, the query will be processed using a variation of the extended boolean model. The extended boolean model is a method of determining how close a document – or, in our case; flight destination – matches a query. More information on boolean ranking can be found in [Salton 1983].

We will implement some additional techniques to refine the query after it has been submitted. Some of the possible refinement techniques we could use are: stemming, stopping, normalisation, and relevance feedback. In addition to the refining process, the query will be expanded using local context analysis. This means that the expansion will happen within the database of possible results. However, we will not restrict ourselves to this method, since others could be more effective.

Semantic expansion or real-time query expansion – in which the user chooses the terms for expansion manually – may be implemented as well.

This section will be tested with recall and precision measurements. The team will implement the techniques, mine through the results, and calculate the precision and recall for each method. Each method's F-measure (the average between the precision and recall) will be calculated and the results will be compared to find the highest F-measure and therefore the best methods. We will also test expansion term retrieval speed to see which methods are the most efficient, as well as ask users to evaluate the expansion implementations to find out which methods are most appropriate.

Ranking of search results

Each search result needs to be given a relevance rank or weight. In order to do this, we need to map a search term to various flight destinations and assign a "weight" value to this destination. In order to determine how closely the query matches the destination, we will use an inverted file in order to store these weights. A simple example of an inverted file is shown in table 1 below.

Table I. Inverted file example.

Romantic	Paris : 5 Zanzibar: 3
Holiday	Jamaica : 6 Australia: 3
Island	Hawaii: 6 Philippines: 5

Once this inverted file is made for every term, we will create a 'relevance rank' by using these weights in order to determine how closely related a destination is to a query. We are currently considering employing this technique, but we may have to revise it if we feel it is providing unsatisfactory results.

Optimisation techniques also need to be employed on these inverted files. Initial weights for terms such as "romantic" will be taken from places such as Trip Advisor. These weights will be dynamic, and will increase if users click on the results. Dynamic weights will ensure that the search results will get more and more refined as more people use the system.

The testing of the ranking system will be done in stages. Initially, test subjects will be recruited and asked to do a test case based test. The user will be asked to search using a predetermined set of parameters, and will then be asked to assess the relevance of each search result. However, when the system is successfully pulling flight destinations from the actual database, users will be asked to perform any search they like, (using a dummy UI), and assess the relevance of results returned.

In order to evaluate our ranking we will be using measures like Mean Average Precision (MAP) and Normalised Discounted Cumulative Gain (NDCG).

User Interface Design

The user interface design is going to be carried out using Human-Computer Interaction methodologies, and hence, most of this section will be made up of:

– Finding out what extra features frequent fliers would want from current travel agency sites in the future.

– Designing an experiment that tests our hypothesis.
– Choosing our user population.
– Constant interaction with the frequent fliers to see whether the system design is turning out to be what they described or what they meant/wanted (users are involved in the entire design process).
– The development phase of the website will follow, still involving a lot of user input in this development stage.
– The biggest part of this subsystem would be the testing of the design. We will have to do a substantial number of tests ranging from usability all the way to efficiency testing. We will also need to make sure that different user groups are involved in the testing.

For this part of the project, most of the evaluations will be qualitative because we are going to observe the flow and interaction of the user with the subsystem, and find out how satisfied users are with the interface. Methods such as user compensation and usability expert hours would be useful, as well as empirical and inspection techniques. The main goal of this section is simplicity which leads to ease of use. We shall be testing a number of hypotheses, all of which will be to do with the usability of the website. We shall be doing full hypothesis testing which includes the null hypothesis and significance levels. The hypotheses are listed below.

1. Does the usability of the site directly affect online booking?
2. Will asking users to create user profiles affect their willingness to use the site?
3. Will the standard menu bar or a no menu approach help people find destinations easier?

6.3 Overall evaluation plan

The system will be evaluated through an initial temporary integration into the Travelstart website. The website will display a "Try Me Button" for users that want to participate in the initial version of the extreme search. Users will be able to rate their experience of the system when they complete their search. The user-interface will be tested separately with usability experiments as mentioned above. The key indicator of success for the project would be whether Travelstart likes the system or not. Moreover, it would be whether they actually implement it on their website or not, how often users use it, and how well received it is by users.

7. RESOURCES AND MATERIALS

Since we are developing a website, the user interface will need to be coded in HTML with CSS or C# with ASP.NET. As a starting point, we are considering a bootstrap to aid the initial design, and thereafter edit the skeleton to suit our needs. JavaScript will be used to facilitate user interactions.

Due to the differences of the sections, we may use different languages for the different subsystems. For the back-end of the system, we will most likely use Java, but we will experiment with Node.js in the database section since it seems most appropriate.

We will also be granted access to the Travelstart API via SOAP/REST. All flight information will be taken from Global Distribution Systems (GDS). Images, and additional destination information will be taken from DBpedia.

8. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

There are a number of ethical and legal issues that come into play with any project, the main one being plagiarism. Referencing work is going to be of paramount importance as credit needs to be given to the original authors. Copyrights need to be respected; especially in a project of this nature, since we may use parts of other websites as we complete our own.

The use of APIs from the tourism industry is another issue. The use of such APIs may be restricted for academic use. There may be compatibility issues between the API and the intended development platform. Nearing the end of the project, a decision will be made by Travelstart, to determine whether they will incorporate the system into their website or not. In order to use the system, they will need permission from the students.

One important ethical issue we have to consider is that of intellectual property. Who will own the website and its associated technologies at the end of the project? If Travelstart decided to implement our website, will we be able to get compensation for it? Ideally, we would want full intellectual property rights over the website (including the back-end) with the option to sell these rights to Travelstart. We also have to make sure that the non-disclosure agreement (NDA) presented to us by Travelstart is fair, and respects our intellectual property rights.

9. ANTICIPATED OUTCOMES

By the end of this project, we expect to have a fully functional flight search system. This system will work via user input in form of natural language queries, image queries or faceted queries. The system's key feature is the natural language search function, but behind it will be query expansion, term weighting and a faceted search. We also expect to have an innovatively designed user interface that makes the use of the system easy for users while providing them with all the information they require.

The expected impact of this project would be a new way of searching for holiday destinations and booking flights. The purpose of this project was to help Travelstart find the competitive advantage over their competition by implementing a search feature that is currently unavailable.

10. PROJECT PLAN

10.1 Risk assessment

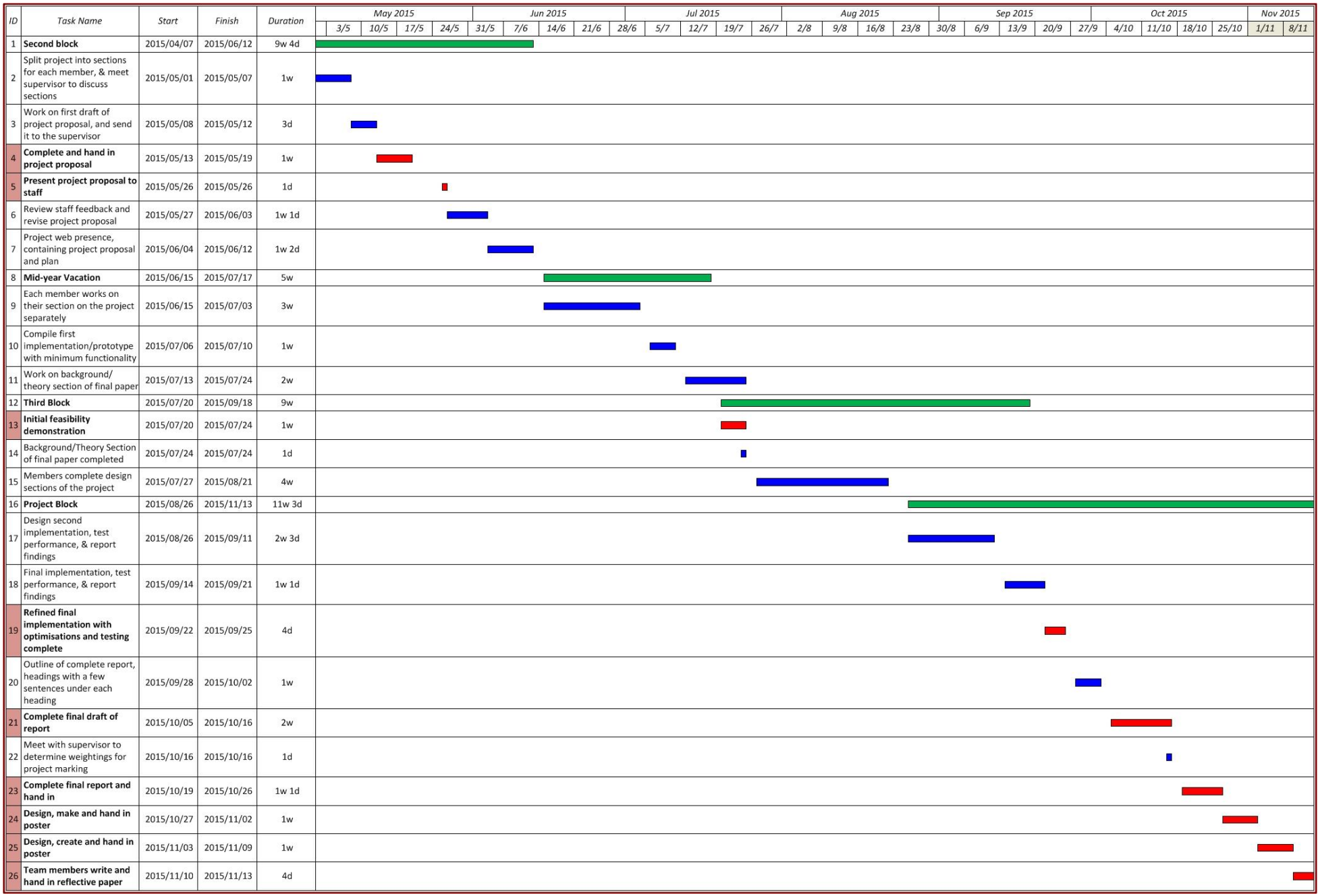
The risks that may occur during the project's process are documented below. The probability of occurrence, the severity of impact, and a category of severity based on probability and impact are given for each risk. Ways of mitigating each risk to avoid it, ways of monitoring the risk during the project, and finally, ways of managing the risk if it does occur are specified.

Table II. Risk assessment.

Risk Condition	Consequence	Category	Probability	Impact	Mitigation	Monitoring	Management
Failure to meet project deadlines for hand-ins.	A late penalty of 10% is enforced for every day the hand-in is late. For some hand-ins, resubmission is not allowed and the team will get 0 for the hand-in.	Medium	Low	High	The team needs to stick to a rigorous schedule for each task. The schedule should include extra time for each task in case of unforeseen obstacles.	Regular check-ups on the schedule should be done to ensure the team is on time.	If the team does fall behind, extra hours will need to be put in to complete tasks on time, or as soon as possible. The team will also have to increase their speed of work. If this is not possible, the team will have to decrease the scope of the task.
Incorrect estimation of scope	The proposed project could be incomplete, and some of the features will not be implemented, or the project could be too simple, and will not be enough for an Honours level project.	High	Medium	High	The proposed project should be well outlined and well thought out so that the scope is correct. Feedback from the presentation of the project should be taken into consideration and the scope should be adjusted if necessary.	The first iteration or implementation should be done over the course of the mid-year vacation. The time it takes for the team to complete this iteration will be used to monitor whether the project's scope is correct.	If the scope has been incorrectly estimated, the team will have to consult the project supervisor and find out whether they can change the scope to better suit their capabilities. If the scope cannot be altered, the team will focus on the most important components of the system and implement them first.
Group member does not complete his part of the project	The final product will have significant issues in some aspects. The team may not meet the deadlines due to other members having to do extra work.	Medium	Low	Medium	Each team member should have equal work to do. All the team members should have an idea of what the other team member's work is so that they can help each other.	Regular meetings with the group should happen to keep track of progress, and interventions should be made as necessary.	The group member that is not pulling his weight should be confronted by the other members. If the problem persists, the team should consult the project supervisor.
Misunderstanding of project requirements between client and development team	The final product will be something that the client does not want or need.	Medium	Low	Medium	The team should have clear instructions from the client. The team should stick to the outline given by the client and not develop based on their own preferences.	Meetings with the client to determine whether the team is on the right track should be organized.	The team will have to change the product on the next iteration if the feedback from a meeting is negative. The client's feedback should be used to create a final product that the client wants.
The project can be very resource intensive due to searching large databases	The performance of the search functionality will be very low and the end product could prove to be unusable.	Medium	Low	Medium	The final project will have to be optimised continuously throughout iterations so that it is always usable. This means that the schedule should allow time for this.	The first implementation should be tested on performance to determine whether changes need to be made, and to allow enough time to make the changes.	If the product performs slowly in the early stages, the underlying code will have to be optimised or changed so that it performs better before adding more features for the next iteration.
Tasks required to complete the project is beyond the team's skillset	Some features will not be implemented, or will be partially implemented, resulting in an incomplete final product.	High	Medium	High	Carry out research to decide whether features can be implemented with the skills that the team has.	The first prototype implementation would also give the team a better idea of the difficulty of the project.	The features that the team cannot implement should be excluded or the team can implement them within limits.

Client and group cannot agree on terms in the NDA	The group will not get access to Travelstart's API.	Medium	Medium	Medium	Try to agree on terms of use of information early.	This will be determined at the start and will not need monitoring.	Group develops their sections in a separate environment, with dummy data.

10.2 Timeline with Gantt chart



ID	Task Name	Start	Finish	Duration
1	**Second block**	2015/04/07	2015/06/12	9w 4d
2	Split project into sections for each member, & meet supervisor to discuss sections	2015/05/01	2015/05/07	1w
3	Work on first draft of project proposal, and send it to the supervisor	2015/05/08	2015/05/12	3d
4	**Complete and hand in project proposal**	2015/05/13	2015/05/19	1w
5	**Present project proposal to staff**	2015/05/26	2015/05/26	1d
6	Review staff feedback and revise project proposal	2015/05/27	2015/06/03	1w 1d
7	Project web presence, containing project proposal and plan	2015/06/04	2015/06/12	1w 2d
8	**Mid-year Vacation**	2015/06/15	2015/07/17	5w
9	Each member works on their section on the project separately	2015/06/15	2015/07/03	3w
10	Compile first implementation/prototype with minimum functionality	2015/07/06	2015/07/10	1w
11	Work on background/ theory section of final paper	2015/07/13	2015/07/24	2w
12	**Third Block**	2015/07/20	2015/09/18	9w
13	**Initial feasibility demonstration**	2015/07/20	2015/07/24	1w
14	Background/Theory Section of final paper completed	2015/07/24	2015/07/24	1d
15	Members complete design sections of the project	2015/07/27	2015/08/21	4w
16	**Project Block**	2015/08/26	2015/11/13	11w 3d
17	Design second implementation, test performance, & report findings	2015/08/26	2015/09/11	2w 3d
18	Final implementation, test performance, & report findings	2015/09/14	2015/09/21	1w 1d
19	**Refined final implementation with optimisations and testing complete**	2015/09/22	2015/09/25	4d
20	Outline of complete report, headings with a few sentences under each heading	2015/09/28	2015/10/02	1w
21	**Complete final draft of report**	2015/10/05	2015/10/16	2w
22	Meet with supervisor to determine weightings for project marking	2015/10/16	2015/10/16	1d
23	**Complete final report and hand in**	2015/10/19	2015/10/26	1w 1d
24	**Design, make and hand in poster**	2015/10/27	2015/11/02	1w
25	**Design, create and hand in poster**	2015/11/03	2015/11/09	1w
26	**Team members write and hand in reflective paper**	2015/11/10	2015/11/13	4d

10.3 Deliverables

All the deliverables are included in the timeline/Gantt chart. They have been marked with a red block on the left, and a red bar for the time it needs to be worked on. The red was used to help the team easily identify important parts on the timeline.

10.4 Milestones

Along with the deliverables of the project, there are a few milestones we have that are of importance. They have also been noted on the Gantt chart.

After completing the project proposal, we will need to reflect on the feedback that we got from the staff, and revise our project proposal. The new proposal will then be the project we will carry out. This proposal and the project plan will be posted on our project's website before the end of the semester which will mark the initial web presence of the project. The next milestone we have is the first implementation/iteration. This will be done during the mid-year vacation and will only contain minimal functionality. The prototype can then be sent to the client to review so that we can find out if we are on the right track.

Due to how short the third block is, we will not be able to do an iteration in addition to our modules. Therefore, the second iteration will be done during the start of the project block. This iteration should have most of the functionality present and should be close to the final implementation. The final iteration and the rest of the milestones are all deliverables that are included in the Gantt chart.

10.5 Work allocation

Our project is equally divided among our group members since each of us are taking a section from the procedures.

Dylan will be creating the faceted database, the query formulation and expansion will be done by Luqmaan, the weighting of results will be done by Shuaib, and Ngoni will design the user interface and connect it to the back-end of the system.

REFERENCES

Raffaella Bernardi, Diego Calvanese, Luca Dini, Vittorio Di Tomaso, Elisabeth Frasnelli, Ulrike Kugler, and B. Plank. 2006. Multilingual search in libraries. The case-study of the Free University of Bozen-Bolzano. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*. 2006, Genova, 2287-2290.

Claudio Biancalana and Alessandro Micarelli. 2009. Social tagging in query expansion: A new way for personalized web search. In *Proceedings of the International Conference on Computational Science and Engineering (CSE'09*). IEEE, Vol. 4, Vancouver, BC, 1060-1065.

Ondrej Chum, Andrej Mikulik, Michal Perdoch, and Jiri Matas. 2011. Total recall II: Query expansion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, Providence, RI, 889-896.

Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. 2007. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*. IEEE, Rio de Janeiro, Brazil, 1-8.

W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. Search engines: Information retrieval in practice. Addison-Wesley.

Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web (WWW'02)*. ACM, New York, NY, 325-332.

Hui Fang. 2008. A Re-examination of Query Expansion Using Lexical Resources. In *Proceedings of the Association for Computational Linguistics (ACL'08)*, Vol. 2008, Columbus, Ohio, 139-147.

Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. 2002. Finding the flow in web site search. *Communications of the ACM* 45, 9 (Sept. 2002), 42-49.

Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Communications of the ACM - Supporting exploratory search* 49, 4 (April 2006), 41-46.

Kamran Massoudi, Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. 2011. Incorporating query expansion and quality indicators in searching microblog posts. *Advances in Information Retrieval,* 6611 (2011), 362-367.

Roberto Navigli and Paola Velardi. 2003. An Analysis of Ontology-based Query Expansion Strategies.

In *Proceedings of the 14th European Conference on Machine Learning, Workshop on Adaptive Text Extraction and Mining*. Cavtat-Dubrovnik, Croatia, 42-49.

Gerard Salton. 1984. The use of extended Boolean logic in information retrieval. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data (SIGMOD'84)*. ACM, New York, NY, 277-285.

Osma Suominen, Kim Viljanen, and Eero Hyvänen. 2007. User-centric faceted search for semantic portals. *The Semantic Web: Research and Applications* 4519, 356-370.

Dong Zhou, Séamus Lawless, and Vincent Wade. 2012. Improving search via personalized query expansion using social media. *Information retrieval* 15, 3-4 (June 2012), 218-242.