ASCUS - Collaboration Finder

Saad Arif

Marticulation Number: H00011110

Fourth Year Dissertation - First Deliverable

Supervisor: Ruth Aylett

Second Reader: Patricia Vargas

April 17, 2014

Contents

1 Introduction **1**

1.1 Aims and Objectives 1

2 Literature Review **2**

2.1 Organisational need for Expert locator 2

2.2 Why do individuals want to seek experts? 3

2.2.1 Why people do not want to seek experts? 3

2.2.2 Stages of finding Expertise 4

2.2.3 Traditional Approach 5

2.2.4 Contemporary Approach 6

2.2.5 Recommendation Systems 7

2.3 Conclusion 9

3 Technology Analysis **10**

3.1 Database 10

3.1.1 MySQL 10

3.1.2 SQLite 10

3.1.3 Conclusion 10

3.2 Server Side 11

3.2.1 Ruby on Rails (RoR) 11

3.2.2 Active Server Pages (ASP) 11

3.2.3 PHP Hypertext Processor (PHP) 11

3.2.4 Conclusion 11

3.3 Testing 12

3.3.1 PHPunit 12

3.3.2 SimpleTest 12
3.3.3 Conclusion 12

4 Project Methodology 13
4.0.4 Development Methodology 13
4.0.5 Detailing Iterative Development 14
4.0.6 Testing Strategy 14

5 First Iteration 16
5.0.7 Requirements 16
5.0.8 High Priority 16
5.0.9 Medium Priority 18
5.0.10 Low Priority 18
5.1 Design 18
5.1.1 User Interface 18

6 Risks Assessment 23
6.1 Initial Database Design 24

7 Initial Evaluation and Testing Strategy 25
7.1 System Evaluation 25
7.2 System Testing Strategy 26

8 Project Plan 27

List of Figures

5.1 Search Page 19

5.2 Profile Page 20

5.3 Home Page 21

5.4 Log in Page 22

6.1 Initial ER-Diagram 24

List of Tables

6.1 Risk Assessment 23

8.1 Project Plan 27

Chapter 1

Introduction

ASCUS(Art Science Collaborative) is a non profit volunteer organisation that is made up of a network of artists and scientists who are looking for interdisciplinary collaboration. ASCUS' aim is to advocate and facilitate improved art science collaboration and science communication (ASCUS n.d.). However ASCUS does not have any tools to meet such aims. Currently ASCUS has a single static site which provides information on past events as well as information on the organisation. This site allows users who wish to participate in collaboration to get initial ideas about the organisation and how to get involved. However the site does not actually help to find potential collaboration opportunities. Thus in order to advocate and facilitate collaboration it must be possible for members of ASCUS to have a central location where they can easily and quickly find collaboration opportunities. This project will focus on implementing a application that will allow ASCUS to meet its aims of art and science collaboration.

1.1 Aims and Objectives

The aim of this project is to develop a web application in order to allow artists and scientists to find collaboration opportunities. The main objectives of the new application are:

- Develop a web application that will reduce work load on artists and scientists when looking for collaboration opportunities.
- Develop facilities that allow artists and scientists to discuss ideas for collaborations.
- The application should be designed in a way that allows both artists and scientists to use it effectively.

Chapter 2

Literature Review

This section will cover the background literature consulted for this project. The literature that was consulted was focused on Expert locator systems, Recommendation Systems and Ontologies.

Expert locator systems allow users to find experts in a particular area. This project is focused on building a subset of an expert locator system, that is focused on facilitating the ability to find collaboration opportunities. The areas that will be looked at are, what are the reasons for using expert systems, how users find experts without tools and different approaches to building an expert locator systems.

2.1 Organisational need for Expert locator

Many organisations have identified the need to locate knowledgeable individuals within there organisations. It is important for organisations to effectively use their knowledge in order to enable organisational learning, providing better technical assistance and creating teams to deal with critical situations among other goals (Ackerman, Pipek, and Wulf 2003). Furthermore an organisation may end up "reinventing the wheel" even though a solution had already been made for a similar problem before. Thus it becomes necessary to catalogue skills and expertise of individuals, who knows what, in way that it can later be queried (Becerra-Fernandez 2000).

Examples of organisations developing expert finding systems:
Hewlett Packard (HP), a company in the computers and electronic equipment market developed an Expert-Finder. The goal of the project was to build a network of experts which consisted of a database user profiles. The user profiles gave a summary of the users knowledge and skill (Becerra-Fernandez 2000).

The National Security Agency (NSA) has also attempted to build a system to locate experts within the organisations. The goal of the his project was similar to HP, identification and cataloguing of knowledge and skills within the organisation (Becerra-Fernandez 2000).

ASCUS has similar needs to the previous companies named. The organisation requires a tool that allows its members to find collaborators. Thus it needs to catalogue its members' expertise, so they can be mined by members who are looking for collaborators.

2.2 Why do individuals want to seek experts?

Yimam-Seid and Kobsa(2003) offer a few reasons as to why individuals may seek experts. They state there are two major reasons why individuals seek experts. (a) They need specific information from the expert and (b) They need the expert who to perform some function. The people seeking experts for the first reasons are usually looking to replace or complement other sources of information such as documents. Some scenarios for this reason include seeking information that is not documented, using experts to minimize ones own effort or individuals may prefer interacting with humans rather than documents or computers.

People seeking experts for the second reasons need experts for a continued period of time where the expert will be working for them or with them. Usually the search for this type of reason is performed more carefully than for obtaining information from experts.

ASCUS members looking for collaboration fall into both categories. While members may be be looking for active collaborators to participate with in their project, they may also be seeking advice or information into a field they have no expertise in. Thus it is important for the system to facilitate both types of members.

2.2.1 Why people do not want to seek experts?

Allen (1977) noted some reasons as to why information seekers may not want to use their colleagues as a source of information and prefer the use other information channels. In his study of 19 engineers, he found a higher correlation between frequency of use with accessibility than with quality of the source of information. He further found that information seekers, found the transaction of information seeking as a costly one. The cost was perceived in the chance of a response that maybe "ego threatening", a loss in status and seeming incompetent. For these reasons engineers would first look at documentation as a source of information.

While it has been shown there are issues when seeking experts, they are not of primary concern. ASCUS is organisation for collaboration thus it is assumed that members are looking for interaction. Further it can be assumed that many members have already been part of collaborations and thus the previously stated issues should not apply to these members.

2.2.2 Stages of finding Expertise

McDonald and Ackerman(1998) identified three stages individuals went through to find expertise with in an organisation. These stages were Expertise Identification, Expertise Selection and Escalation.

Expertise Identification

Expert identification is defined as “the problem of knowing what information or special skills other individuals have.” It is further noted that expertise identification is difficult problem to solve. It contains many varying factors such as what is expertise , how will it be used within the given context and the problem of handling the change of individuals skills and expertise as time goes on. One solution to such expertise identification is to “consider the types of historical artifacts that are employed by local users as resources and then incorporate use of those within the system.” (McDonald and Ackerman 1998)

Expertise Selection

After determining who has what expertise it is intuitive to then select the most appropriate individual(s) that will solve the problem. McDonald and Ackerman(1998) define expertise selection as “appropriately choosing among people with the required expertise.” Furthermore they observed that expert seekers usually used three expertise selection criteria, “organizational criteria, load on the source and performance.” Expert seekers tried to find experts that were local and when that failed they went to different departments within the organisations. Expert seekers, further more took into account how busy experts were, approaching the least busy first. Finally they firstly approached experts that were better at explaining solutions or had better “attitudes”.

Expertise Escalation

Escalation is the process by which people resolve the failure of the expertise identification or selection mechanism. The expertise seeker may try to identify other experts or pursue other experts that maybe able to solve the problem. This does may involve asking members higher up in the organisation hierarchy, asking help from less desirable experts or even searching for experts in a different department within the organisation (McDonald and Ackerman 1998).

The system being built should emulate the first two stages and try to avoid the third stage as much as possible. When the system carries out expertise identification, it should aim to asses if individual is an expert in a particular area with as much accuracy as possible. In the second stage the system should order the found candidates in an ranked list with the top candidate being the most useful to expert seeker. The ideal solution would be to never reach the expertise escalation stage at all but their is no known system that is 100% accurate in expertise identification and selection. Furthermore it is unclear from the literature as to how a system can help the user in the expertise escalation stage.

2.2.3 Traditional Approach

Many organisations contain roles which act as expertise or information locators. In Allen's(1977) discussion he presented a highly linked role within an organisation, which served to bring relevant information to informations seekers. Other researchers have found similar roles within different types of organisations. Ehrlich and Cash(1994) found what they called an "information mediator", who because of his breadth of knowledge and interpretation skills was the go to person in case of any problems. They also noted that the information mediator was a critical part of the organisation. McDonald and Ackerman(1998) found role that they called "expertise concierge". The expertise concierge has the knowledge of who within the organisations knows what. When a person was looking for expertise, they would ask the expertise concierge about people who maybe be able to help them. The concierge will use their knowledge to suggest individuals that match the query.b

One way in which the previously stated roles can be emulated and automated is by building an expert database. Such an idea works by manually entering expertise data into the database, which can then be queried. The expert database will return a set of individuals that may be of help. Furthermore the expert database may return the individuals in a ranked list. Individuals higher up the list will be more likely to be of help. However such a system do have limitations (Yimam-Seid and Kobsa 2003).

1. Developing the databases is a labour intensive and expensive.

2. For the such a system to work , it relies on the experts willingness to spend time initially providing information about their expertise.

3. Due to a continuous change in peoples expertise it is hard to keep the databases up to date.

4. There is usually a disconnect between expertise description entered into the database and the expert related query. The expertise description are usually general and incomplete while the expert related queries are very specific.

2.2.4 Contemporary Approach

In order to combat the limitations of the traditional approach the expert finding process can be further automated. This is done by automatically obtaining information on experts from many different sources and not relying solely on human sources (Yimam-Seid and Kobsa 2003). Using this method the experts do not need to update their profile manually and thus it is always up to date.

However this approach has many difficulties. Firstly it is can be difficult to find sources on which to judge the skill level of an expert in a particular area. An example source for scientist could be articles published but for artists this much more difficult as their work is more difficult to analyse. A common interdisciplinary source that can be used to within an organisation, are email communications. Emails can demonstrate expertise, as queries can be answered by the expert and communication patterns can help determine who has what knowledge with in the organisation (Campbell et al. 2003). This becomes particularly difficult to achieve within a non profit organisation where communication is private between volunteers and does not go through through an organisational email server. Furthermore using such a emails as source of information requires dealing with privacy issues. However even if a source of information was available it would still be difficult to determine the weight of the source. For example if a scientist co-authors an article does that make the scientist an expert in that area? Did both authors equally contribute to the article?. Such questions are difficult to answer without further information and if sources are not properly weighted then it is more likely for incorrect experts to be identified.

2.2.5 Recommendation Systems

In the previous sections the techniques described are based on a user querying a system and getting back a result. These approaches rely on the system to make the decision on who the expert is. However it is easier and quicker for humans to make that decision when compared to a computer system. Thus it should be possible to let the users aid the system in the expert finding process. Furthermore the system should be able to use the history with the user in order to find a suitable expert. This system is known as a recommendation system. It is based on the idea of using information that other users have found and the evaluation they have made. It involves the user providing recommendation for an item.The recommendations can be anything from detailed review to simply a numerical score. The system then aggregates those recommendations, which are then used to the direct the item to the appropriate users. They can also be used to match recommender with those seeking recommendation. Furthermore the recommendation system can use previous history of the user in order to fine tune the recommendations.

The recommendation system can complement the previously discussed systems. It is able to give the users extra information by showing them what experts have been recommended by other users. Furthermore the expert finder no longer has to decide if a user is an expert but rather only has to make sure if the expert is appropriate to the query. However Resnick and Varian (Resnick and Varian 1997) noted several issues that must be considered when designing a recommendation system.

Volume The volume of items that will be evaluated has to be considered. The process of recommendation has to match the volume of items. It may be possible to write detailed reviews and recommendations on restaurants however this not practical on momentary news article because they put an emphasis on quick distribution and evaluation.

Benefit and Cost When designing a recommendation the domain of the system has to be considered for the following questions. Is it more costly to miss a good item that could be recommended to the user or to recommend an item that is a poor match to the user?. How do the benefits of finding a good match for the user compare to cost of mismatching the user with an item?. These questions will effect the design of the system. An example of this is within an medical recommendation system, where the cost of incorrect recommendation with be very costly. Thus this system has to be more accurate and more intelligent in its recommendations then a recommendation system for restaurants.

Consumers and Producers The consumers and producers of the recommendation system have to be considered. What type of people produce the recommendation and consume them. How many consumers are their compared to the producers?. Do the produces evaluate similar products or do the products vary? Identifying the consumers and producers will help in deciding how to aggregate the recommendation. For example aggregating the recommendations by taste is more useful when the consumers and producers do not know each other.

Social Implications Once the user has identified the areas they are interested in, it easy to consume evaluations made by others and not contribute. Thus system must have some form of incentive to produce recommendations. Otherwise if not enough recommendations are made then system will not be used. However even if incentives are given it still possible for users to only give positive recommendations for their own items and negative recommendations for competitors. Furthermore the system can be further ”gamed” by a users offering monetary gains for giving their item positive recommendations. Therefore some methods should be in place to discourage this behaviour.

Privacy Recommendation systems may cause privacy concerns for their users. The more information the user has on a recommendation the better the evaluation on that recommendation. However users may not wish to share information about themselves and their habits. A solution to this, is to allow anonymous recommendations. However this does not entirely solve the problem as user may wish for a mix anonymity, as they may want to be rewarded for their recommendations but still maintain privacy.

In conclusion even though recommendation systems at first seem to complement expert finding systems their as still major issues that need to be addressed. Furthermore it should be noted that a recommendation system is not simply a sub system that is part of an expert finding system but in its self a large stand alone system. Thus combining the two will have implications on the complexity of the system as well as the time taken to develop it. This factor as well as previous discussed factors have to be addressed in order for the combination of systems to be successful.

2.3 Conclusion

Through the research carried out on expertise finding systems as well as recommendations system knowledge was gained into the system designs. By

Chapter 3

Technology Analysis

This chapter will look at the major technologies that will used within this project. This analysis is based of the initial requirements.

3.1 Database

The database will be used to store user details such as log in and password as well as the users information displayed on the users profile.

3.1.1 MySQL

MySQL server is open source relational database with extensive customisation options for performance tuning. It also offers capabilities to set user accounts and permissions. The author has experience in working with MYSQl.

3.1.2 SQLite

SQLite is a relational database in the public domain. Unlike the MyQl, SQLite does not operate in a client-server model as the database is contained within a file. Thus the SQLite databases can be embedded into an application while other databases works by having a separate database server with which to communicate.

3.1.3 Conclusion

Both database have similar features and both perform slightly different tasks. MySQL offers more control, while SQLite offers easy of set up. Due to the short time in which this project will take place, the MySQL database is chosen as the author has previous experience in it, allowing for immediate productivity.

3.2 Server Side

3.2.1 Ruby on Rails (RoR)

Ruby on Rails is a framework built on the ruby programming languages and uses a MVC arhictecture. The RoR philosophy is convention over configuration. By following the set of conventions the developer can be more productive and the framework ”just works”. Thus it is possible to be very productive but it does have a steep learning curve. Further it is relatively new, being created in 2005, it lacks the mature documentation.

3.2.2 Active Server Pages (ASP)

Active Server Pages (ASP) is a web framework from Microsoft, that you can use to create and run dynamic web applications (Neudesic 2013). In order to use ASP a the server must run a Microsoft I.I.S. (Internet Information Server) operating system. Further more the database must be Microsoft SQL Server(MS-SQL). Though it is only limited to the windows platform, without considering third party tools that allow for cross platform functionality, it does integrate well with other windows technologies.

3.2.3 PHP Hypertext Processor (PHP)

PHP is a general purpose language suited for web development. It provides many features suited for web development and can be used as scripting language. PHP is one of the most widely used programming languages for the web, thus has mature documentation as well as plethora of tools to support its development. It also considered to the have lowest barrier to entry as it is very easy to set up. Furthermore the author has previous experience in PHP.

3.2.4 Conclusion

PHP will be used for server side scripting, as it is mature and well documented. Furthermore the previous experience allows the author to be productive sooner than with other technologies.

3.3 Testing

In order to preform thorough testing it is important to automate the testing by using a testing framework. By automating it becomes easy to perform tests after every change to the application and any issues will be immediately reported. Thus it saves the developer from wasting time manually running algorithms on data. Furthermore Automated tests can be easily execute many different types of tests thus providing a wide coverage which is difficult to do manual tests.

For PHP two popular unit testing frameworks will be considered, PHPUnit and SimpleTest. Both testing frameworks provide similar features such as creating unit tests, running them automatically and showing failed tests and passed test. Therefore evaluating the frameworks will focus on other aspects such as documentations, integration with other tools.

3.3.1 PHPunit

PHPUnit is the standard unit testing framework that is included in many PHP framkeworks such as Zend Framework, Cake PHP. PHPUnit is well maintained as updates have been done regularly. PHPUnit seems to have larger user base as well as more online tutorials. PHPUnit is integrated into many PHP IDEs such as Eclipse, Netbeans, Zend Studie, PHPStorm. PHPUnit requires a terminal in order to run tests making it difficult to use on a remote server.

3.3.2 SimpleTest

SimpleTest is another unit testing framework with the focus on simplicity. SimpleTest is not well maintained with last update over a year ago. SimpleTest does offer the ability to run tests in a web browser thus not needing a terminal, which can be useful when working on a remote web server.Furthermore adding SimpleTest to a project is easy as simply using the include command in a PHP script. While many tutorials are available for SimpleTest, many of them are over a year old. Eclipse IDE offers a SimpleTest plugin.

3.3.3 Conclusion

While SimpleTest and PHPUnit offer similar in features in terms of creating tests and automatically running them. PHPUnit is better maintained thus easier to find documentations and help with issues should any arise.

Chapter 4

Project Methodology

This section will describe and justify the development methodology that was used in this project. It will also describe how user feedback was gained through prototyping and questionnaires.

4.0.4 Development Methodology

Due the user centric nature of the project, it was best to use iterative development and avoid the traditional waterfall methodology. By Using iterative development, the project focused on cycles of prototyping small parts of the project, implementing the prototype and finally getting feedback. The original prototype was constantly improved as more feedback was obtained.

The use iterative development helped manage the unpredictable nature of working with users. It helped as it was flexible to adapt to any changes needed by the users. Furthermore, due to continually prototyping and implementing small parts, the user would still have some functionality even if the project could not be finished. If the waterfall methodology was adopted and significant difficulties were encountered, it would have been likely that project would not finish and have no functionality implemented at all. Lastly any issues regarding requirements or user interface design could be solved early on in the development process. If the waterfall method was used and their were issues with requirements and/or user interface it would take significant time in order to correct the issues.

In conclusion many of the problems encountered when developing interactive software were dealt with by the use of iterative development.

4.0.5 Detailing Iterative Development

When using iterative development, the requirements of a project develop over time during each iteration. Thus it is not appropriate to report the requirements as whole as it does not show the evolution of the system and does not clearly show the need of the users at each stage of the iteration. Furthermore it does not show how the system was changed based on the feedback received. A more common and valuable approach is to to report the details of each iteration separately. The details will include what was done at each iteration, what feedback was gained from the previous iteration and how it affected the current iteration. By using this approach a clear evolution of the project can been, which will act as document to describe how the final release version was reached. However a full summary is still available in the appendices which states the full requirements and their final status.

4.0.6 Testing Strategy

Their are many different methods that can be used to test software and to ensure the program operates correctly. This chapter describes the testing strategy employed in this project to achieve validation and verification of this web application.

Validation of an application describes if the application does what the user wants. While verification describes if the the application matches its specification. In order to ensure that was the web application was thoroughly tested both white box and black box testing methodologies were used.

The black box testing methodology focuses on checking if the functionality operates correctly i.e. Does the code take in an input and give the correct output. While white box testing methodology looks at the code internally in order to determine if the code operates correctly.

The following testing strategies were used, unit testing, integration testing and acceptance testing.

Unit Testing Unit testing involves testing individual components and objects which make up the system. Each object or component is an individual unit which needs no other objects in order to operate. For unit testing black box testing methodology was used. Black box testing

is much less time intensive than white box testing and thus suited this fixed time scale project more appropriately.
PHPUnit unit testing framework was used in order to automate the process and produce formal results for each individual unit. However PHPUnit works well when the the units being tested does not interact with outside units such as a database. For example the registration process in involves validating and inserting the user data into the database. While the validation components can be tested independently, manually testing is need to ensure that correct data is entered into the database. Since this project is primarily focused on manipulating the database thus this type of testing was only useful in a limited scope.

Integration Testing This type of testing was carried out when a new functionality was added. The unit tests for previously added functionality were run as well as the new functionality was tested manually using test data. Furthermore any previously added functionalities that did not have unit tests, were also manually in order to ensure previous functionality had not broken. This type of testing was quite time intensive and after consideration, a framework should have been chosen to automate this type of testing.

Alpha Testing The alpha testing was carried out by a single student at Heriot-Watt University. The purpose of this testing was to put the web application under the stress of a normal user. Though the student was using the application normally, the student was asked to intentionally attempt to break functionality. By doing this the testing would give a large coverage and unveil subtle bugs. This testing was very successful as it unveiled many critical bugs.

Chapter 5

First Iteration

This chapter explains process by which the first initial prototype was created. The first prototype created was a concept prototype i.e. their were no implementations of it. As this was the first iteration, it was mainly focused on gathering requirements and focusing on the user interface of the prototype.

5.0.7 Requirements

The requirements of the project were gathered by speaking to the client and then prioritised order to manage the project effectively. The most important requirements were given the tag High. These requirements must have been completed before the end of the project. The next priority tag was the Medium tag, any priorities with the Medium tag must be started only after the High tag requirements have been finished. While the Medium tag priorities were not essential, it was recommended that at least some if not all of them should be implemented. Finally the least important requirements were given the tag Low. These requirements were not needed for the main functionality and only provided extra features for convenience.

5.0.8 High Priority

Search for Collaborators

This is has been identified as the most important requirement. The users shall be able to search for collaborators in the database. The criteria for search shall be:

Area of work: The user must be able to search for collaborators in the database who are Artist or Scientists. Those areas will be further split into different types of Artists and

Scientists. These must also be searchable.

Location: Search for collaborators in a particular city.

Distance from user: Search for collaborators with in a certain distance from the city the user is in. The user will not need to register to use this functionality.

Register and Log in

User shall be able to register as member and customize their profile. To register the user must provide a first and last name as well choose a user name and password. Once the user has registered they can then customize their profile. The profile shall contain the following items:

- first and last name (Required)
- age (optional)
- email (optional)
- Phone/Mobile number (optional)
- Biography (optional)
- Artistic and/or Scientific interest (optional)
- Profile Picture (optional)
- Extra pictures - in order to demonstrate some previous work (optional)
- External Links - in order to show previous work done, such as blogs and journal articles (optional)

The username and password provided will be used to login, which will allow the members to edit their profile. Members can only edit their own profiles. While only administrators can remove members, which in turn will also remove the profile.

High usability

This as been given a high priority as an application with high usability and consistent design will attract more users. Even with a functional application, in order to for the application to be successful and help collaboration, the user must find it intuitive and easy to use.

5.0.9 Medium Priority

Map visualization

This feature allows users to visually see the location of members in a map. The map will show icons representing members on map. The dots will become bigger as more members are in a particular area. There should be a zoom functionality, as the user zooms in the more detailed density of the members becomes clear. This is likely to be the most difficult requirement and is likely to scoped down in order to be manageable with in the project deadlines.

Projects

This feature will allow members to create projects, that users can search for. The project should have a Title, description,images to demonstrate ideas, goals and what skills are required. The original member to start the project, can edit the project and delete the project. There will be a comment section for each project allowing members to give feedback and ask questions. Administrator can only delete the project.

5.0.10 Low Priority

Forum

User will have the standard forum functionality available . Users will be able to create threads and post comments. Moderators maybe assigned with the chosen privileges.

5.1 Design

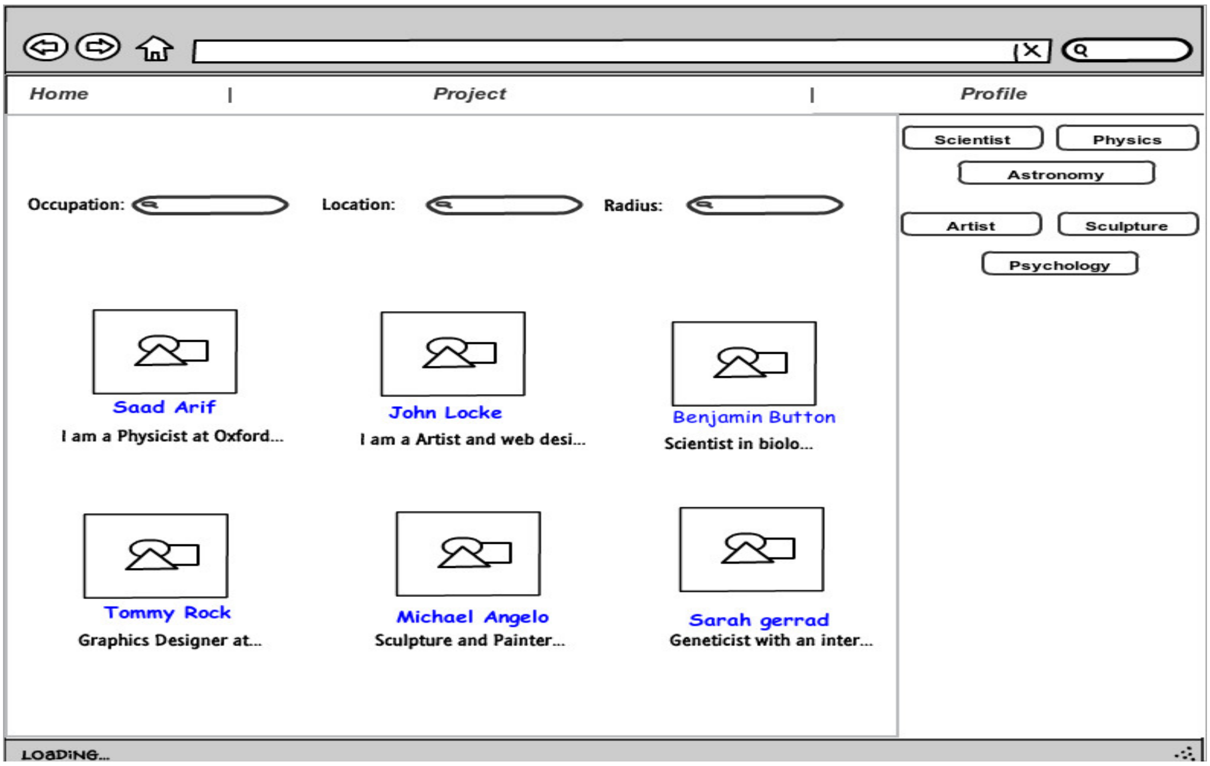
The design of the system was split into two parts. The prototype which would show how the user interface would look and how the user would interact with it. The back end design which describe how all the features would be implemented and how the database architecture would be designed. The design are explained below.

5.1.1 User Interface

The requirements detailed the system to be highly usable, which meant it was important that the user interface was intuitive, simple and aesthetically pleasing. However as this was an initial prototype and low fidelity, this meant that the aim was of the first prototype was to focus on intuitiveness and simplicity.

Though many prototyping applications were available, Lumzy was chosen because it offered all the important features and it was free. Though alternative would have been to use paper prototyping, it would have been much more difficult to maintain.

Search Page



Saad Arif
I am a Physicist at Oxford...

John Locke
I am a Artist and web desi...

Benjamin Button
Scientist in biolo...

Tommy Rock
Graphics Designer at...

Michael Angelo
Sculpture and Painter...

Sarah gerrad
Geneticist with an inter...

Figure 5.1: Search Page

This interface design is for allowing users to search for collaborators. The search bars allow the users to search by area of work, location and radius of the location. Once the search is completed, a grid of member information will appear. Each member that matched the search criteria will have their profile picture, name and first few words of their biography shown. The far right hand side shows popular areas in which members have claimed they have expertise in.

Profile Page

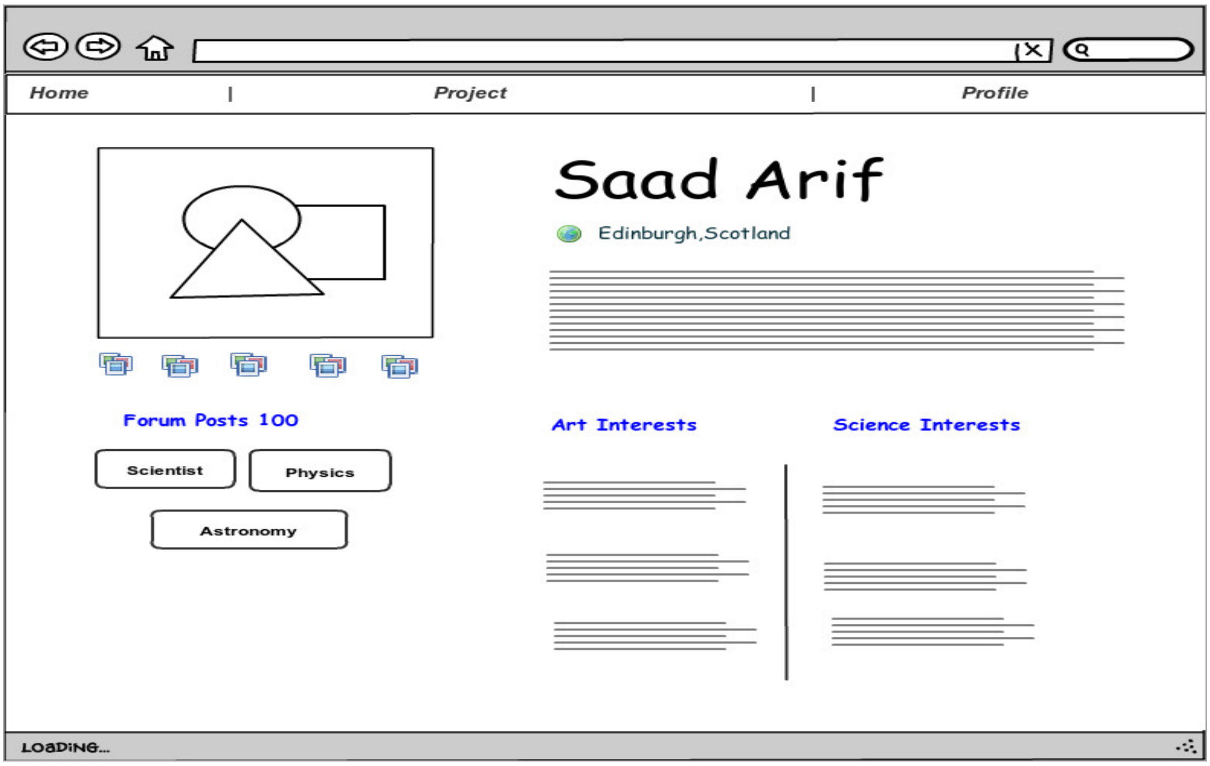


Figure 5.2: Profile Page

The figure above shows the interface design for the second main requirement, the profile page. This shows how the profile page will look after the user has entered his/her details.

The profile contains useful information that other users are likely to be interested in. It includes location, small biography. Also specific part is dedicated to simply talk about artistic and scientific interests. This will allow for more freedom to say what interests the member regardless of occupation.

On the bottom left it also shows the areas in which the member has claimed to be an expert. Expertise area coupled with the artistic/scientific interest should give the user enough information to judge if the registered member is appropriate for collaboration.

Home Page

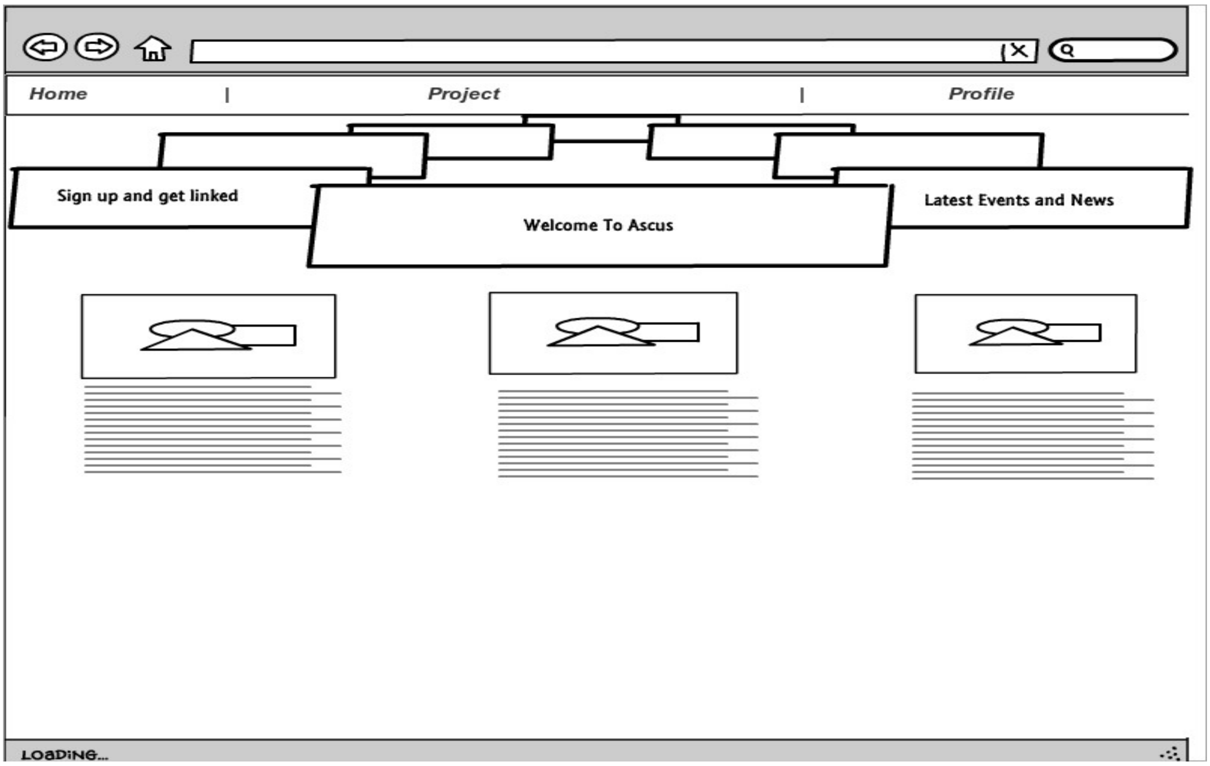


Figure 5.3: Home Page

The figure above shows the home page for the application. The purpose of this page is to introduce the user to the application as well as act as marketing tool to new users. This page should be aesthetically pleasing and contain information to hook the user in and use the application. The carousel acts as navigation tool as well as gives the user knowledge of what is available on this site quickly.

Log in

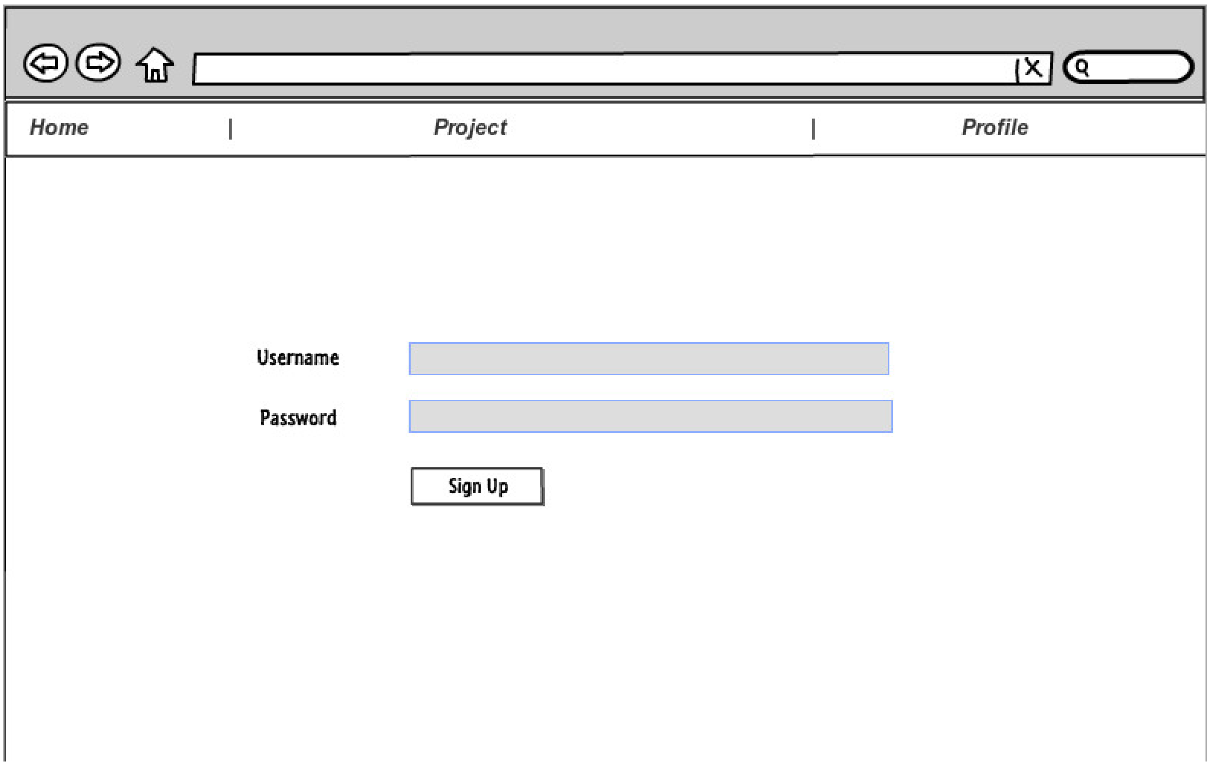


Figure 5.4: Log in Page

The above figure shows very standard log in page.Thought not shown in the figure, this page will also show error message when the user name and/or password is incorrect.

Chapter 6

Risks Assessment

This section will give an an overview of the potential risks to project development.

Risk	Probability	Severity	Mitigation Strategy
Browser compatibility issues	High	Moderate	Constantly test browser compatibility through out the development cycle. Leverage automatic test frameworks in order to supplement manual testing.
Fall behind on plan	Moderate	Moderate	Give small leeway time for each part of the application. If significantly behind schedule then only complete vital parts of the application.
Person of interest is unavailable	Moderate	Moderate	Another client is kept up to date with the main development of the application.
Users find the system unusable	Moderate	High	Follow user centred design approach. Maintain regular meetings with the client. User testing must be carried out extensively and feedback should be integrated into the next iteration.
Low quality feedback	Low	Moderate	Pilot all methods used to gain user feedback. If low quality feedback is given reassess previous methods for weaknesses and integrated any finding.

Table 6.1: Risk Assessment

6.1 Initial Database Design

In order for the database design to reflect the requirements, it is important to identify the separate entities with in the requirements. Three entities were identified, member who has registered, the member's profile and areas of expertise the user has. The figure below shows the relation ship between the entities.

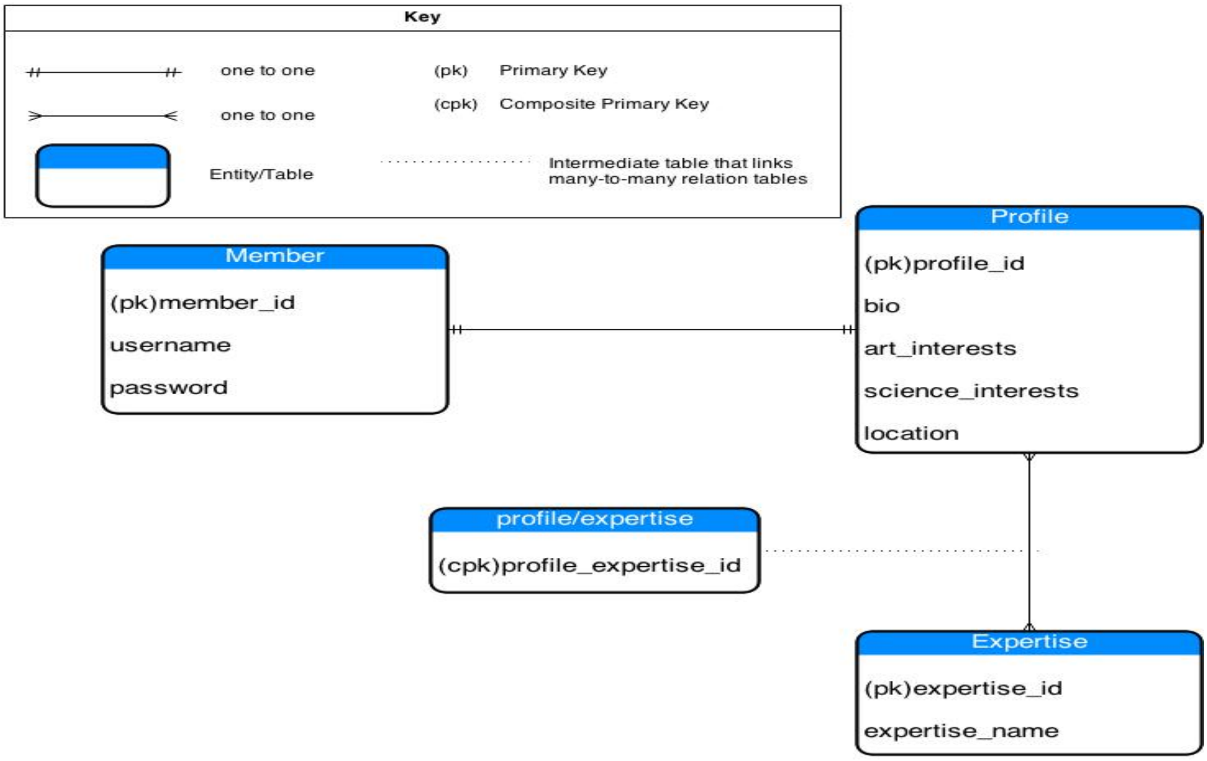


Figure 6.1: Initial ER-Diagram

The relationship between an member and his/her profile is a one-to-one. Each member must have a profile and vice-versa.

Every member has many expertise and many expertise can be shared by many members. Thus we have a many-to-many relationship between member and expertise areas. However the member table is used to keep log in information thus the expertise table has a relationship with the profile, since the profile is used to display the profile in web application.

In order for the many-to-many relationship to be implemented it must require an intermediate table. The profile/expertise table will act as link between the profile and expertise table. It will use an composite primary key consisting of the primary key from the profile table and the primary key from the expertise table.

Chapter 7

Initial Evaluation and Testing Strategy

This chapter describes the plan for testing and evaluating the system.

7.1 System Evaluation

In order to meet the high usability requirement, it is important to involve users when developing the system and user interface. If the system is not usable it will lead to a system that no one will use thus rendering the project unsuccessful.

In order to evaluate the system, prototypes will be used. The system evaluation will be conducted by allowing expected system users to interact with the prototypes. Afterwards the user will be asked to fill out an questionnaire, which will include open and closed questions. The closed questions will be used focus on a particular aspect of the design, while open questions will be used to get in depth feedback.

In order to gain even more feedback a semi structured interview will be used. It will be decided in advance which themes and areas are needed to be explored during the interview. Using a semi structured interview allows for more freedom to discuss various areas and explore areas based on the users answers. The interviews will be recorded, as writing the users answers will likely not capture exactly what the user said. It also frees the interviewer to focus on the users answers rather than with writing them down.

Before any questionnaire and interviews are conducted the user will be given an consent form detailing what will the questionnaire/interview involve, all information gathered will be anonymised, the user can withdraw their participation as they see fit and any recordings will only be kept as long as necessary for the project.

7.2 System Testing Strategy

In order to sufficiently test the system the testing strategy will be split into two parts, testing during development and testing after development. Rather than testing being a separate phase in development, it will be integrate into the development cycle. After each feature and or the smallest independent part is completed, tests will be written for said part. The tests will be unit tests and will be written using the PHPUnit test framework.

Using the automated test framework, this allows us to carry out integration testing as well. As development progress, tests can be run after the integration of different parts of the application. If the integration causes any unforeseen, bugs the automatic testing should signal a problem. This way the bugs can be caught early on when integrating and thus in theory the system should be bug free throughout the development.

It is impossible to foresee all potential problems with the system and test for them. Thus it is important to carry out testing involving the users before releasing the final system. After all features that are to be implemented have been finished the beta testing will begin. The system will be released to a subset of the users and the users will be given an email address to which they can send there issues. The issues will be prioritised and the highest priority issues will be dealt with first.

Sample data will be added in order to typical use of some features e.g. sample profiles will be added so the searching functionality can be used. It is likely that this testing will happen for a short and fixed amount of time due to the static deadlines.

Chapter 8

Project Plan

A high level plan below describes the key tasks to be achieved.

Due Date	Version	Task
15/10/2013	R1	Gather initial requirements.
01/11/2013	N/A	Conduct Literature review and research into relevant technologies
22/11/2013	Final	Hand in final version of the first deliverable
03/12/2013	R2	Create Questionnaire to gather further requirements. Pilot the questionnaire and improve it if needed.
31/12/2013	WD1	Create Website design prototype.
31/12/2013	DD1	Create initial database design and implementation.
31/12/2013	IMP1	Create an initial implementation of high priority requirements
10/01/2014	N/A	Gather feedback on initial prototype and implementations. Improve the current system according to feedback.
17/01/2014	WD2	Create website design prototype for high and medium priority requirements.
27/01/2014	DD2	Update database design in order to facilitate high and medium priorities.
07/02/2014	IMP2	Implement medium priority requirements and integrate with previous requirements.
17/02/2014	N/A	Gather feedback and improve system based on feedback.
27/02/2014	Final	Create final website design prototype for high,medium and low priority requirements.
07/03/2014	Final	Final update to database design in order to facilitate high,medium and low priorities.
17/03/2014	Final	Implement low priority requirements and integrate with previous requirements.
01/04/2014	Final	Test the systems, fix any bugs and release the system.

Table 8.1: Project Plan

Bibliography

Ackerman, Mark S, Volkmar Pipek, and Volker Wulf (2003). *Sharing expertise: Beyond knowledge management.* MIT press.

Allen, Thomas J (1977). "Managing the flow of technology: Technology transfer and the dissemination of technological information within the R & D organization(Book)". In: *Research supported by the National Science Foundation. Cambridge, Mass., MIT Press, 1977. 329 p.*

ASCUS (n.d.). *ASCUS - Art Science Collaborative.* URL: `http://ascus.org.uk/` (visited on 11/07/2013).

Becerra-Fernandez (2000). "The role of artificial intelligence technologies in the implementation of People-Finder knowledge management systems". In: *Knowledge-Based Systems* 13.5, pp. 315 –320. ISSN: 0950-7051. DOI: `http://dx.doi.org/10.1016/S0950-7051(00)00091-5`. URL: `http://www.sciencedirect.com/science/article/pii/S0950705100000915`.

Campbell, Christopher S. et al. (2003). "Expertise Identification Using Email Communications". In: *Proceedings of the Twelfth International Conference on Information and Knowledge Management.* (New York, NY, USA). CIKM '03. New Orleans, LA, USA: ACM, pp. 528–531. ISBN: 1-58113-723-0. DOI: `10.1145/956863.956965`. URL: `http://doi.acm.org/10.1145/956863.956965`.

Ehrlich, Kate and Debra Cash (1994). "Turning information into knowledge: Information finding as a collaborative activity". In: *Proceedings of digital libraries.* Vol. 94, pp. 119–125.

Just talk to me: a field study of expertise location (1998). (New York, NY, USA). CSCW '98. Seattle, Washington, USA: ACM, pp. 315–324. ISBN: 1-58113-009-0. DOI: `10.1145/289444.289506`. URL: `http://doi.acm.org/10.1145/289444.289506`.

Neudesic (2013). *Get started with asp.net.* URL: `http://www.asp.net/get-started` (visited on 11/09/2013).

Resnick, Paul and Hal R. Varian (Mar. 1997). “Recommender Systems”. In: *Commun. ACM* 40.3, pp. 56–58. ISSN: 0001-0782. DOI: 10.1145/245108.245121. URL: http://doi.acm.org/10.1145/245108.245121.

Yimam-Seid, Dawit and Alfred Kobsa (2003). “Expert-Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach”. In: *Journal of Organizational Computing and Electronic Commerce* 13.1, pp. 1–24. DOI: 10.1207/S15327744JOCE1301_1. eprint: http://www.tandfonline.com/doi/pdf/10.1207/S15327744JOCE1301_1. URL: http://www.tandfonline.com/doi/abs/10.1207/S15327744JOCE1301_1.