



## Project Proposal

4 May 2007

Graham Hunter

Department of Computer Science  
University of Cape Town  
Rondebosch, 7701  
South Africa  
ghunter@cs.uct.ac.za

Marc Pelteret

Department of Computer Science  
University of Cape Town  
Rondebosch, 7701  
South Africa  
mpeltre@cs.uct.ac.za

Tshifhiwa Ramuhaheli

Department of Computer Science  
University of Cape Town  
Rondebosch, 7701  
South Africa  
tramuhah@cs.uct.ac.za

Project Supervisor: Dr Hussein Suleman (Computer Science, UCT)

## 1. INTRODUCTION

Grocery shopping can be a nightmare, especially when one person decides what needs to be bought but another person has to do the buying. Typical problems include:

- Difficulty in sharing the shopping list – what is the list written on and where is the list kept?
- One person adding something to the list and another wondering who added it and why
- Going shopping, only to realise that the shopping list has not been brought
- The buyer not knowing which brand of item to buy
- Co-ordination: who does the shopping and when?
- When the buyer gets to the store, they do not know where items can be found and spend a long time wandering around the store looking for them. Also, shoppers may visit the same section several times because they do not realise that several of the items on their lists can be found in that section.

The key aim of this project is to make grocery shopping easier by using technology to aid the process. This does not mean that technology will be used to automate the shopping process entirely; rather, the aim is to use it to provide a tool for shoppers to use to make shopping simpler.

Two types of communication technology will be used: cellular telephony and the Internet. The goal is to allow a household to share and manipulate a shopping list stored on a central server via a Web interface or a cellphone. The typical use case is that of one user creating the shopping list for the current day or week and another person views that list when they do the shopping. If any changes are made to the list, all the users with access to the list can see them.

The project will involve the creation of a database to store the shopping list details, the setting up of a Web application to manipulate and read from the database, and the design and implementation of the Web and cellphone end-user interfaces. The Web interface will be developed using modern Web-development techniques and technologies, such as AJAX, and the cellular telephone interface will be developed in the Micro Edition of the Java Platform (J2ME).

The two interfaces will allow the creation and modification of a user's shopping list. The Web application will handle the related database changes and synchronise between the interfaces.

## 2. RELATED WORK

The works that concern this project are those that deal with mobile applications, more specifically, communication between a central server and a mobile device.

There have been a number of works related to a server sending information to a mobile interface. Seedorf et al. [2005] developed a system for Java-enabled handheld devices with Internet connectivity to query a server. Communication was limited to HTTP and messages were sent using XML. An XML schema was used to describe the message formats. This architecture could possibly be used in the project.

Works that deal with delivering tailored content to users via mobile devices are also applicable. Olsson and Nilsson [2002] created a system for providing event notifications to a mobile device. Their mobile device can communicate using WAP or SMS. The content is tailored based on the user's location which is determined via cell triangulation or GPS.

Ozen et al. [2002] describe a system that delivers highly personalized information from XML resources to mobile clients. The system allows for users to be sent data based on queries on XML data. The results are converted into the mark-up language of the mobile device such as CHTML, XHTML and WML.

Finally, work that covers the behaviour of people when shopping or using shopping lists is of interest. Ludford et al. [2006] discuss how people create pre-planned information resources (lists) and then take these to refer to at the place where the task is performed. An idea they cover is that mobile devices can be used to provide these information resources. This idea is the basis for this project.

## 3. METHODOLOGY

To find out what is required in a mobile shopping aid, user studies will be done. People from different household types (for example, student and family households) will be interviewed to see what their shopping behaviour is. From this we will determine the most important features they require.

To test the usability of the system, prototype mobile and Web interfaces will be developed. These will then be tested by users, who will be observed while performing specific tasks such as adding to a list and viewing a list.

Finally, the final system will be tested through a simulation where several user "households" will carry out a specified set of typical tasks using the system. The households will be mixed groups of people, rather than actual households. The groups' progress will be monitored and they'll be asked to give feedback after the simulation.

## 4. OUTCOMES

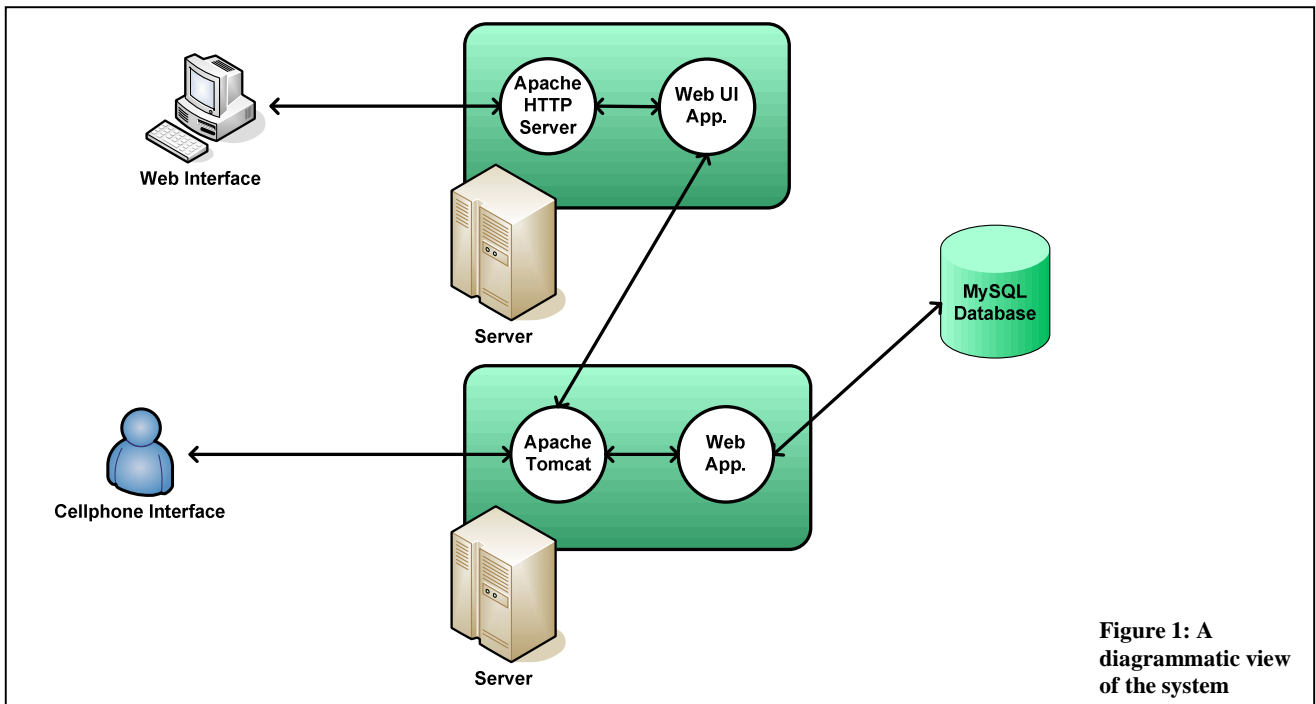
### 4.1 System

The system comprises two layers: the front-end, which has two interfaces (one Web-based, the other cellphone-based) and the back-end, which is made up of a database and a Web application through which the front-ends can access information stored in the database. The two layers will communicate using SOAP.

The system will support multiple users and lists. Each list will have an owner who can:

- Make the list accessible to specific users, the idea being that all members of a family or household can access the list
- Set the operations each user can perform on the list.

Users can add private items to the list. Such items are put on the list but are shown only to those the user gives permission to (the list owner will automatically be able to see them). This feature is included because certain items are private by nature and also because users may wish to prevent certain other users from seeing specific items for some reason. For example, an item on the list might be a gift for someone else who can view the list.



**Figure 1: A diagrammatic view of the system**

The system will have the following key features:

- Create, view and manipulate the current list
  - Add and remove items
  - Per item annotation facility, so that users can leave notes for other users about items in the list
  - Make items private.
- Add new products to the database.
- View a full history of previous lists
  - Pre-populate the current list based on the list history
  - View statistics about the list history, such as the most frequently bought item.
- Manage the user information, such as preferences and who can access specific lists.
- Provide shop layouts and in-store shopping routes, including the grouping of list items according to their location in a specific store.
- Users can check items off the current list as they are bought.

There are several other features that could be added to the system. These may be implemented if there is sufficient time to do so.

- Product suggestions based on users' list histories
- Community features – the ability to inform others about various things, like a shop being out of stock of an item or item specials and discounts
- "State" management, particularly the management of the lifetimes of products (to warn of expiration).

Features specific to parts of the system are listed as follows.

#### 4.1.1 Back-end Web application and database

- Store all information used by the system
- Perform calculations for system
- Provide a SOAP-based service point.

#### 4.1.2 Web-based end-user interface

- Communicate with the server over the Internet
- Present a Web interface for users.

#### 4.1.3 Cellphone end-user application

- Communicate with the server via SOAP over General Packet Radio Service (GPRS)
- Present a mobile interface for users.

### 4.2 Questions tackled

1. Can we use a combination of Web and mobile technology to create a system to simplify the process of grocery shopping?
2. Can the system be designed to be easy for first-time and non-technical users to use?
3. How well will such a system perform on a cellphone?
4. Does J2ME provide what is needed for the cellphone interface?

### 4.3 Expected impact of project

The result of the project should be a system that allows users to perform routine grocery shopping more easily by reducing the

amount of time it takes to compile and manage a shopping list. The system would allow users to easily share information about shopping tasks, leading to a higher level of co-ordination.

## 4.4 Key success factors

### 4.4.1 Back-end Web application and database

- The features specified for the system are fully implemented and function properly.
- The server is able to perform operations in an acceptable time and bear an acceptable workload.

### 4.4.2 Web-based end-user interface

- The features specified for the system are fully implemented and function properly.
- Users consider the interface to have good aesthetics and be easy to use.
- The interface has good performance and can bear an acceptable workload.

### 4.4.3 Cellphone end-user application

- The features specified for the system are fully implemented and function properly.
- Users consider the interface to have good aesthetics and be easy to use.
- The interface performs operations in an acceptable timeframe.

## 5. WORK DETAIL

### 5.1 Risks

Listed below are risks that could potentially be encountered during the project. For each we list the consequences of the risk, ways it can be mitigated, what monitoring can be performed to avoid it, and how it can be managed if it occurs. Each risk is also awarded a rating, based on its probability and impact. A matrix of possible values is shown below.

		Probability		
		Low	Medium	High
Impact	Catastrophic	C	B	A
	Critical	D	C	B
	Marginal	E	D	C
	Negligible	F	E	D

#### 5.1.1 The Java-enabled phone is not obtained in time

- Consequences: The phone application cannot be fully tested. Testing the application on a computer may be enough, but such testing should be verified by testing on the target hardware.
- Mitigation: Ensure that an appropriate cellphone is obtained early on in the project.

- Monitoring: Check with the cellphone source that delivery of the device is on track.
- Management: Source another cellphone from somewhere else, if possible.
- Rating: D (impact: critical, probability: low).

#### 5.1.2 The server cannot run the back-end Web application

- Consequences: The back-end is needed by both the web and cellphone applications since it does all the data processing, storing and retrieving of data from database. The front-end applications will be unable to retrieve and perform any operations on the list without it.
- Mitigation: Create stubs for use by the interfaces. Also, ensure that the computer used as a server meets the required specifications of the software that will be run on it and install and test the software early on in the project.
- Monitoring: Continue to test the system as the project progresses to make sure that it can cope with the back-end tasks.
- Management: Upgrade the machine, optimise the software installed and/or the software produced, and/or re-locate the back-end system to another machine. Having the stubs means that development of the interfaces can still continue, as they will be able to function without the server.
- Rating: E (impact: marginal, probability: low).

#### 5.1.3 The server cannot run the front-end Web application

- Consequences: Various Web-application-specific functions will not be able to be used. The only way to access and manage the shopping lists will be through the cellphone.
- Mitigation: Ensure that the computer used as a server meets the required specifications of the software that will be run on it and install and test the software early on in the project.
- Monitoring: Continue to test the system as the project progresses to make sure that it can cope with the Web application tasks.
- Management: Upgrade the machine, optimise the software installed and/or the software produced, or re-locate the back-end system to another machine.
- Rating: D (impact: critical, probability: low).

#### 5.1.4 The database is not available

- Consequences: The system will not be functional at all if the database is not available, since all the information used by the system resides in the database.
- Mitigation: Implement the database early on in the project and ensure that it is stable. Create test data that can be used during development.
- Monitoring: Regularly check that the database is functional and its data is not corrupt. Also make a regular back-up of the data.
- Management: Restore a previous backup. Re-locate the database to another server if necessary. Use the test data while the database problems are addressed.

- Rating: C (impact: catastrophic, probability: low).

## 5.2 Timeline and milestones

A Gantt chart detailing the timeline for the project is attached at the end of the proposal. The major milestones are as follows:

### Project preliminaries

- Final proposal 04 May 2007
- Project presentation 11 May 2007

### Project work

- Report: Background Chapter 25 May 2007
- Report: Chapter on Design 30 July 2007
- User evaluation 24 September 2007
- Report: Chapters on implementation and testing 01 October 2007
- Final implementation completed, testing completed 01 October 2007
- Outline of complete report 01 October 2007

### Project finalisation

- First draft of report 15 October 2007
- Final report 19 October 2007
- Poster 23 October 2007
- Web page 24 October 2007
- Project demonstration to supervisor 24 October 2007
- Final project presentations 14 November 2007

## 5.3 Resources

### 5.3.1 Equipment

- Java-enabled cellular phone
- A Web server for running the back-end server
- Three standard computers with Internet access, one for each developer

### 5.3.2 Software

- Java Platform, Micro Edition (J2ME) compiler
- Apache Tomcat
- Apache HTTP Server
- MySQL

### 5.3.3 People

- Developers to design and implement the database, the SOAP back-end, the Web interface and the cellphone interface.
- Users who might use an application such as Cellphone Shopper. Since the service is aimed at a broad range of people, as diverse a sample of users as is possible is required. The users will be involved in the requirements-gathering and testing phases of the project
- Dr Hussein Suleman, who will oversee the project and provide various resources and advice.

## 5.4 Deliverables

The project has the following deliverables:

- A back-end SOAP-based server.
- A database containing basic information, together with software to insert more information into the database.
- A Web-based end-user interface.
- A J2ME cellular phone end-user application.

## 5.5 Work allocation

### 5.5.1 Graham

- Design and implement the back-end, which will include an interface for communication with the user interfaces.
- Implement and manage the database.
- Validate the back-end system.

### 5.5.2 Marc

- Design and implement the Web-based end-user interface.
- Perform the user evaluations of the interface.

### 5.5.3 Tshifhiwa

- Design and implement the cellphone application.
- Perform the user evaluations of the interface.

### 5.5.4 Joint Work

- Gathering of user requirements.
- Design of the database.
- Design of the communication protocols used between the back-end and user interfaces.

## 6. REFERENCES

- [1] Seedorf, S., Korthaus, A., and Aleksy, M. Creating a Topic Map Query Tool for Mobile Devices Using J2ME and XML. *Proceedings of the 4th international symposium on Information and communication technologies.* (2005), 111 – 116.
- [2] Olsson, D. and Nilsson, A. MEP – A Media Event Platform. *Mobile Networks and Applications.* Volume 7, Issue 3 (June 2002), 235 – 244.
- [3] Ozen, B., Altinel, M., Dogac, A. and Kilic O. Highly Personalized Information Delivery to Mobile Clients. *Wireless Networks.* 10, (2004), 665–683.
- [4] Ludford, J. P., Frankowski, D., Reily, K., Wilms, K. and Terveen, L. Because I Carry My Cellphone Anyway: Functional Location-Based Reminder Applications. *Proceedings of the SIGCHI conference on Human Factors in computing systems: Everyday use of mobiles.* April 22-27, 2006, 889 – 898.

