G52GRP Final Group Report

Mobile Auction System

19-03-2013

GP12-MVR2

Group Members:

Jake Howland (jxh01u)

Thomas James Hadkiss (tjh01u)

Marike Helen Littlefair (mhl01u)

Ryan Lobo (rxl01u)

Yao Zhang (yxz02u)

Supervisor:

Milena Radenkovic

Contents:

Contents 2

Figures List 4

1. Introduction 7

2. Criteria Set 8

3. Related Work 9

3.1. Survey of Existing Systems 9

3.2. Market Research 13

4. Project Proposals 14

4.1. Design Proposal 14

4.2. Implementation Proposal 22

4.2.1. Source Code Layering 22

4.2.2. Design Patterns 23

4.2.3. Database 24

5. Implementation 26

5.1. Programming Language 26

5.2. Run Environment 26

5.3. Software 26

5.4. Source Code Architecture Design 27

5.5. SQL Implementation 28

5.6. HTML5/ jQuery Implementation 28

5.7. Implementation Steps 30

5.8. Code Implementation Samples 34

5.8.1. Proxy Bidding 34

5.8.2. Peer to Peer Network Reputation Management System 35

5.8.3. Recommended Item ` 36

5.9. Establishing a Server 38

6. What Was Achieved 39

6.1. SYSREQ 1 – Users can create their own personal account 39

6.2. SYSREQ 2 – Users can view items on auction 45

6.3. SYSREQ 3 – Users can follow items on auction 49

6.4. SYSREQ 4 – Users can bid on items 51

6.5. SYSREQ 5 – Users can manage their transactions 52

6.6. SYSREQ 6 – Users can sell items 53

6.7. SYSREQ 7 – System will implement a feedback system which rates the quality of a user in both buying and selling 57

6.8. SYSREQ 8 – The system should monitor buying and searching patterns of users 58

6.9. SYSREQ 9 – The system is secure in the interest of its users an maintaining integrity 60

7. Testing 61

7.1. JUnit Testing 61

7.2. System Testing Log 63

8. Project Reflections 66

8.1. Project Management Perspective 66

8.2. Problems Encountered 66

9. References 68

10. Appendices 69

10.1. Complete System Requirements Specification 69

10.2. Project Time Plan 76

10.3. Example Minutes 78

Figure List:

Figure 1.1	BidApp! Logo	7
Figure 2.1	Criteria set	8
Figure 3.1	Survey Demographics	9
Figure 3.2	Online services	9
Figure 3.3	How often online services are used	10
Figure 3.4	Prefer online methods as opposed to high street retailers	10
Figure 3.5	Loyalty to an online service	10
Figure 3.6	Used an online auctioning service before	11
Figure 3.7	Current limitations	12
Figure 3.8	Market Research	13
Figure 4.1	Initial Home Page Design & Final Home Page Result	14
Figure 4.2	Flow Chart showing the main page transitions of the application	15
Figure 4.3	Welcome Screen	16
Figure 4.4	Logged-In Screen	17
Figure 4.5	Search Screen and Search Results Screen	18
Figure 4.6	Individual Item Screens	19
Figure 4.7	Messages Screen	20
Figure 4.8	My Account Screen	21
Figure 4.9	Layered model of source code	22
Figure 4.10	Model View Controller Design Pattern	23
Figure 4.11	DAO Design Pattern	24
Figure 4.12	ER Diagram	25

Figure 5.1	Project Folders	27
Figure 5.2	Source Code Packages	27
Figure 5.3	HTML5 and jQuery example Source Code	28, 29
Figure 5.4	Content produced by Figure 5.3	29
Figure 5.5	HTML5 meta-tags	29
Figure 5.6	Implementation Steps	31, 32
Figure 5.7	Test Driven Development diagram	33
Figure 5.8	Partial proxy bidding code	34, 35
Figure 5.9	Reputation Calculation equation	35
Figure 5.10	Partial trust and reputation management code	36
Figure 5.11	Partial Recommended Auction code	36, 37
Figure 6.1	Evidence of satisfied SYSREQ 1.1	39
Figure 6.2	Evidence of satisfied SYSREQ 1.2	40
Figure 6.3	Evidence of satisfied SYSREQ 1.3	40
Figure 6.4	Evidence of satisfied SYSREQ 1.4	41
Figure 6.5	Evidence of satisfied SYSREQ 1.6	41
Figure 6.6	Evidence of satisfied SYSREQ 1.7	42
Figure 6.7	Evidence of satisfied SYSREQ 1.8	43
Figure 6.8	Evidence of satisfied SYSREQ 1.9	44
Figure 6.9	Evidence of satisfied SYSREQ 2.1	45
Figure 6.10	Evidence of satisfied SYSREQ 2.2	46
Figure 6.11	Evidence of satisfied SYSREQ 2.3	47
Figure 6.12	Evidence of satisfied SYSREQ 2.4	48
Figure 6.13	Evidence of satisfied SYSREQ 3.1	49
Figure 6.14	Evidence of satisfied SYSREQ 3.2	50

Figure 6.15	Evidence of satisfied SYSREQ 4.1	51
Figure 6.16	Evidence of satisfied SYSREQ 4.2	52
Figure 6.17	Evidence of satisfied SYSREQ 6.1	53
Figure 6.18	Evidence of satisfied SYSREQ 6.2	53
Figure 6.19	Evidence of satisfied SYSREQ 6.3	54
Figure 6.20	Evidence of satisfied SYSREQ 6.4	54
Figure 6.21	Evidence of satisfied SYSREQ 6.5	54
Figure 6.22	Evidence of satisfied SYSREQ 6.7	55
Figure 6.23	Evidence of satisfied SYSREQ 6.8	56
Figure 6.24	Evidence of satisfied SYSREQ 7.1	57
Figure 6.25	Evidence of satisfied SYSREQ 7.2	58
Figure 6.26	Evidence of satisfied SYSREQ 8.2	59
Figure 6.27	Evidence of satisfied SYSREQ 8.4	60
Figure 7.1	Results of JUnit Testing	61, 62
Figure 7.2	System Testing Log	63, 64, 65

1. Introduction

The purpose of the system was to provide users with a website, accessible via web browsers on mobile devices, which aimed to improve the process of online auctioning. When accessed by a mobile device's browser, the site resembles a mobile application. Figure 1.1 shows the proposed logo for 'BidApp!'.

Figure 1.1 - *BidApp! logo*

The system aimed to promote interaction between users who were both buying and selling items.

The system also supports proxy bidding so bids on the user's behalf may be made.

We support buyer and seller rankings which aim to ensure that a user is trading with reliable nodes. The ranking system is based on recommendations and feedback from other users and on proxy behavioural monitoring.

User profiles are created based on buying/selling behaviour. Using these profiles, a personalised approach for all users is created, and strategically targeted marketing is produced. For example, users who target one or more specific areas for buying and selling is recommended items based in those areas.

A messaging feature enables users to communicate quickly and easily, in order to share product information and ask questions. This provides a safe way for users to communicate, as this channel is monitored by BidApp!'s administration team.

Items for sale have their own listing page, in which full details of the item are displayed, with the option to include images also. Users have a 'Followed Items List', which tracks items that they have bid on or are interested in, enabling them to keep updated with these auctions.

Financial transactions is handled as a back-end service unavailable directly to users. This will increase the security of the application.

2. Criteria Set

To ensure consistency throughout conducted research and designing, a multi-criteria set shown in Figure 2.1, based on the problem description was made These will be referenced throughout the document by the codes assigned to each of them.

Code	Name	Description
CS-1	Rich Interactivity	The site should provide much functionality for the user with a great amount of interactivity.
CS-2	User Profiling	The site should build up user profiles for each of its users, allowing accurate and well-targeted recommendations to be made.
CS-3	Security	Financial Transactions and a users' personal information must be secure at all times.
CS-4	User-Friendly	The site should be simple fast to use, with no overcomplicated interfaces or functionality.
CS-5	User Communication	The site should encourage communication between users and feedback and should promote a diverse and eclectic user base.
CS-6	Reliable	The site should be reliable and all functionality should be operational at all times.

Figure 2.1 – *Criteria set*

3. Related Work

3.1. Survey of Existing Systems

The survey of existing systems consisted of an online survey which was agreed upon by the group.

Figure 3.1 shows the demographic of people who took the survey.

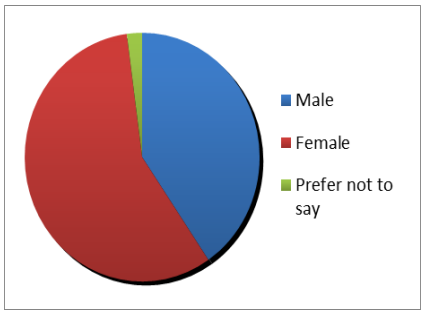


Figure 3.1 – *Survey Demographics*

From this figure we can see that 40.4% were Male, 57.4% were Female and 2.1% preferred not to disclose this information.

Of these people, around 95% said they had used online services for buying or selling goods. This is shown in Figure 3.2.

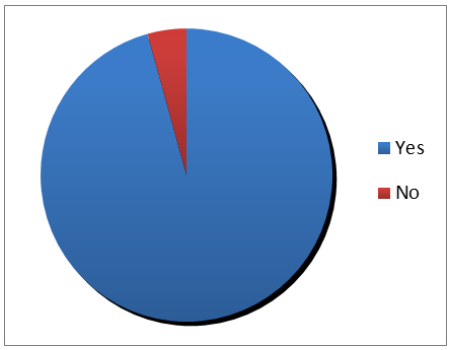


Figure 3.2 – *Online Services*

Furthermore, in answer to “Approximately how often do you use online shopping services each month?” the most popular reply was between 1 and 4 hours. The full results of this can be seen in Figure 3.3.

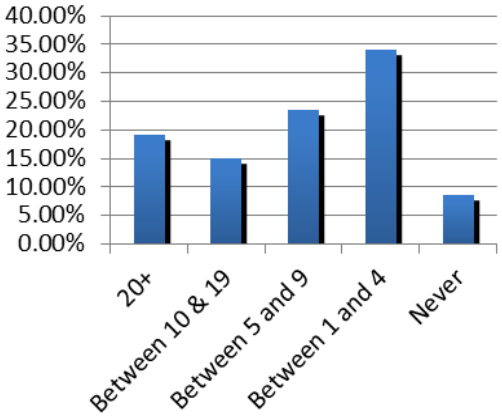


Figure 3.3 – *How often online services are used*

In answer to "Do you prefer this online method as opposed to high-street retailers?", the survey suggested that around ¾ of people preferred to use online shopping methods as opposed to high-street retailers. This is shown in Figure 3.4.

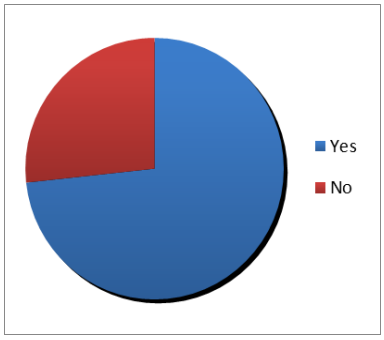


Figure 3.4 – *Prefer online methods as opposed to high street retailers*

When asked: "Do you feel a loyalty to an online service?", over ¾ of people said No, which is shown in Figure 3.5. This would insinuate that if they don't feel loyal to a single site, they would be interesting in using a variety of sites, possibly including BidApp!

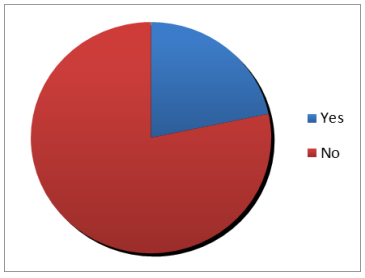


Figure 3.5 – *Loyalty to an online service*

When asked: “When using online trading services, what sort of item are you most likely to search for? (e.g. electronic goods)”, the 4 most popular answers were as follows:

1) Electrical Goods
2) Clothes
3) Books
4) DVDs

When specifically asked about using online auctioning services, just under 4/5 of those surveyed said they had used an online auction service before. This is shown in Figure 3.6.

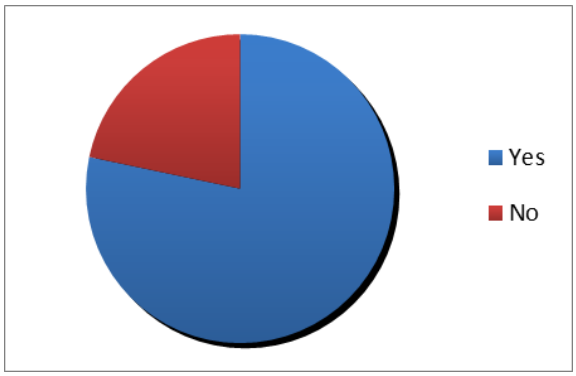


Figure 3.6- *Used an online auctioning service before*

Of all those who answered affirmatively to the previous question, when asked to name exactly which ones they had used in the past month, the answer was unanimously “eBay”.

When asked about the current limitations of the online shopping services, over 50% believed that there were no limitations.

None	56.80%
Poor choice of recommended item listings	5.40%
Search facilities are difficult to use	5.40%
Inaccurate search results	21.60%
Too much advertising	27%

These results can be seen clearly in Figure 3.7.

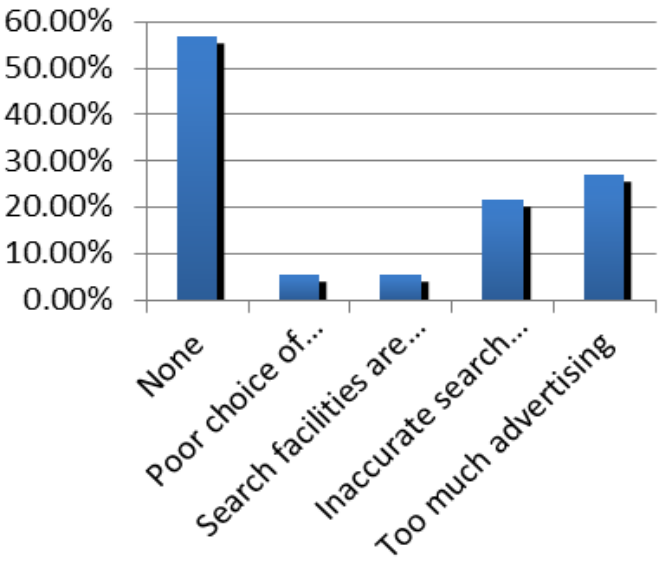


Figure 3.7 - *Current limitations*

When asked to comment on any improvements that could be made to currently used auction sites, most answers centred around searching on these sites, and that they would like to see "More accurate results". Our Criteria set items of: Rich Interactivity, User Profiling and User-Friendliness, should ensure that this is not an issue with the system we will be creating.

3.2. Market research

The market research for the project consisted of talking to several regular users of online auction and trading sides. From the conversations, the following positives and negatives for eBay, Gumtree and Craigslist were amassed. The results of this research are shown in Figure 3.8.

eBay	
Positives +	**Negatives -**
Easy to create an account for the site.	Too many unrelated items appear for some searches.
Simple to navigate the site to find what you want.	The site can look busy and messy at times, too much happening on screen at any one time.
Good to see customer reviews and an average of these as a percentage.	Sometimes you get charged money by eBay unexpectedly, as it is not always clear about insertion fees and commission etc.
Cheap to use, so no high listing fees and high commission payments.	
Being able to set minimum prices for a listing is a good function.	
Users like being able to contact sellers with questions.	
Easy navigation to items through good filters of products (e.g. Womenswear → Trousers → Jeans).	
The mobile application for the site is easy to use for buying and selling.	

Gumtree	
Positives +	**Negatives -**
Allows for search by location so it is easy to see what is geographically closest.	There is an option to meet people in person which could be potentially dangerous.
Easy to contact people through the site, if they provide email addresses or phone numbers.	
Has capability to keep users anonymous.	
Features Google Maps integration allowing you to see exact location of item.	

Craigslist	
Positives +	**Negatives -**
Website operates by locations.	Site looks incredibly dated, uninteresting layout.
Very simple format of website.	Sometimes can be difficult to see images of items.
Wide variety of categories available.	Not as reliable as other options such as eBay

Figure 3.8 – *Market Research*

4. Project Proposals

4.1. Design Proposal

To design the user interface for BidApp!, throwaway prototyping was used. There were many initial design ideas, with the first one pictured below in Figure 4.1, which shows the first and last homepage designs. There were many designs in between.

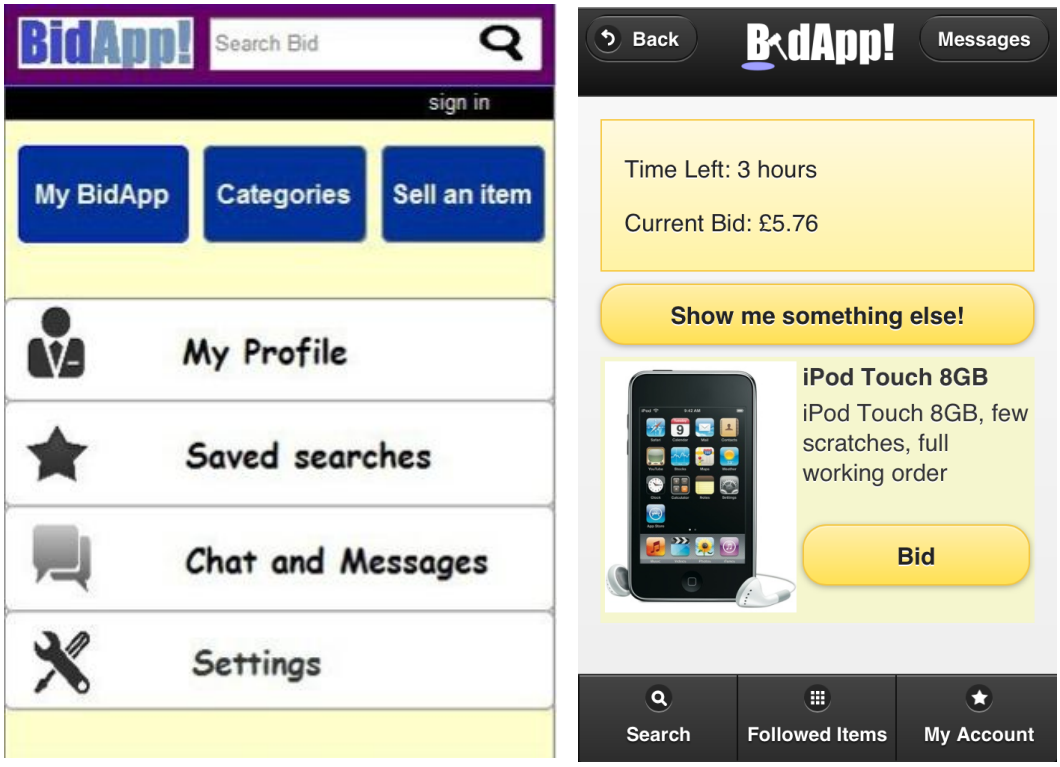


Figure 4.1 – *Initial Home Page Design & Final Home Page Result.*

The requirements document was used to make a list of functions that would need to be included in the application. These functions were then formulated into a page transition diagram as seen in Figure 4.2. This page transition diagram shows the main pages in the application and how they can be navigated to. The diagram focuses on the paths that can be taken by a registered user when using the application.

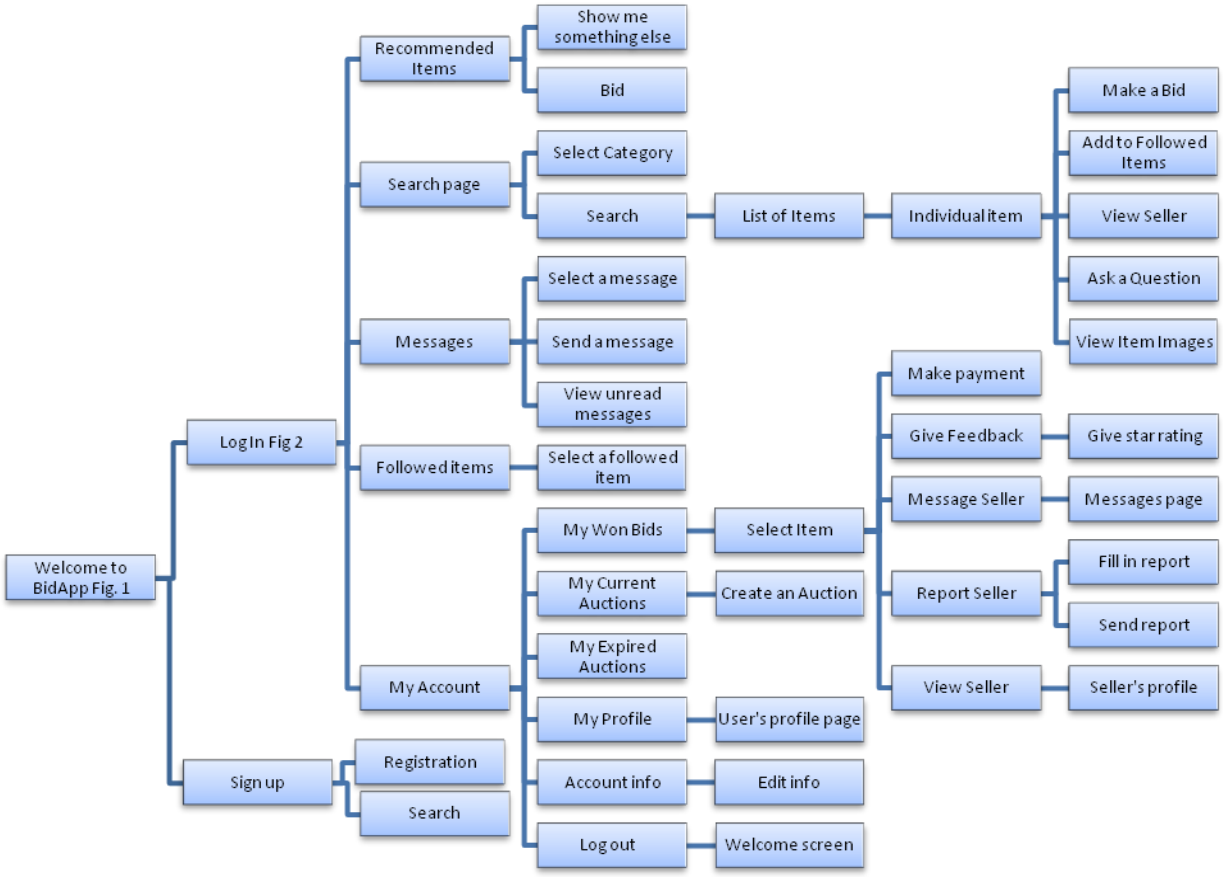


Figure 4.2 – *Flow Chart showing the main page transitions of the application*

There are 3 types of user. The above diagram focuses on the registered user and the guest user. When opening the application, users are taken to the welcome screen, and from here can either log in or sign up.

Guest users can search, with the option to refine their search by category. All other functions are only available to registered users.

From the welcome screen a guest user can ‘Search’, ‘Log in’, ‘Sign up’ or go ‘Back’. While a guest user of the system, these buttons remain static throughout the application because they enable simple navigation whist encouraging the user to create an account. The Welcome Screen is shown in Figure 4.3.

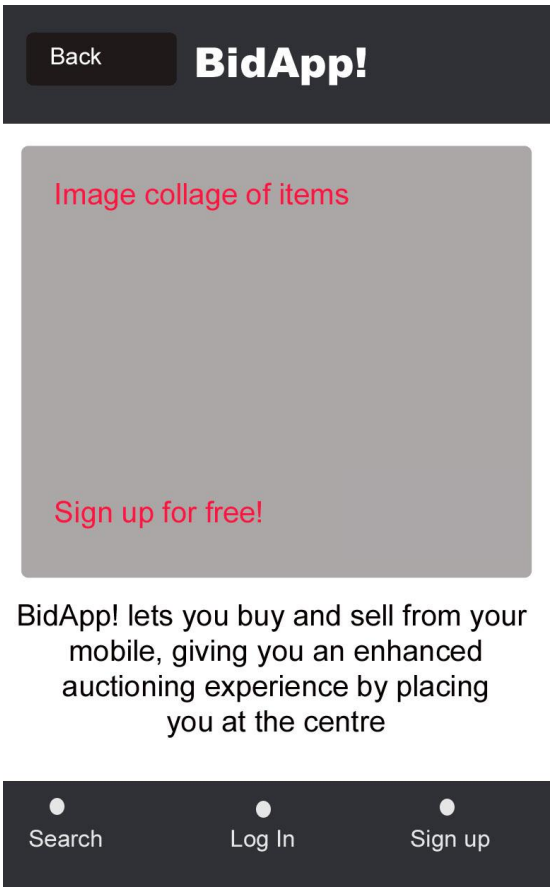


Figure 4.3 – *Welcome Screen*

There are two different ways to log in, either sign up as a new user, or by logging in as a returning user. Once they are logged in, the user is taken to a screen displaying a recommend item, from which they can view another by selecting 'Show me something else', or they can immediately bud on said recommended item. From this screen, they can also select 'Search', 'Followed Items', 'My Account', 'Messages' or go 'Back'. These buttons are static and remain in the same place for each screen of the application when a user is logged in. They maintain ease of use, as well as consistency while creating a frame for the rest of the application. The Logged-In Screen is shown in Figure 4.4.

Figure 4.4 – *Logged-In Screen*

When a user selects 'Search', they can refine the search by category and select the order in which the results are listed. The resulting page is a list of items showing the main item image, the name of the item, the current bid and the time left in the auction. At the bottom of this screen is a 'Previous' and a 'Next' for the user to navigate through the list of items for auction. These screens can be seen in Figure 4.5.

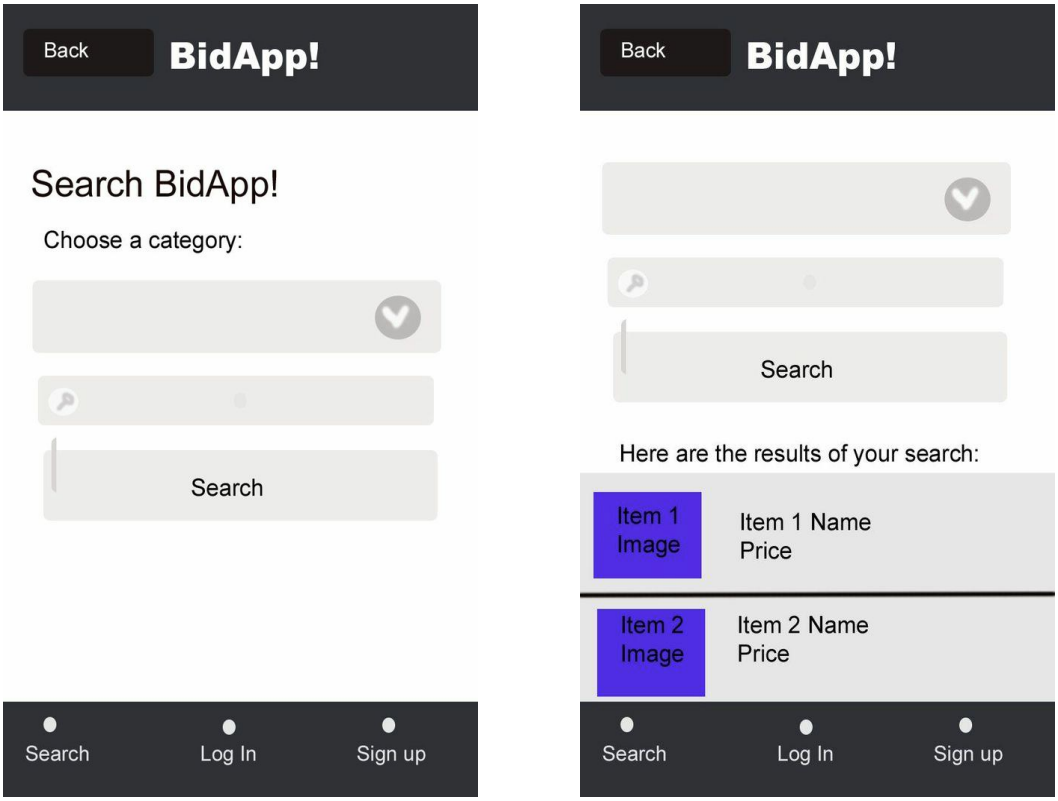


Figure 4.5 – *Search Screen and Search Results Screen*

When the user selects an item from the list of search results they are taken to the item page. This page contains images and a description of the item as well as the current bid, the time left, and the option to 'Make Bid'. Clicking 'Make Bid' takes them to a page where they enter their bid and their maximum bid, and are given the option to enable proxy bidding and 'Add to followed Items'. They are also given the option to 'Add to followed items' directly from the individual item page, as well as 'View Seller' and 'Ask a question'.

'Add to followed items' adds the item to the list of followed items specific to that user, but does not change the page. 'View Seller' takes the user to the seller's profile page. Here they can see reviews of the seller and the seller's current items for auction. From the seller's profile page, the user can also select 'Report Member' which takes the user to a page where there is a short form to fill in and the option to 'Send Report'. This is shown in Figure 4.6.

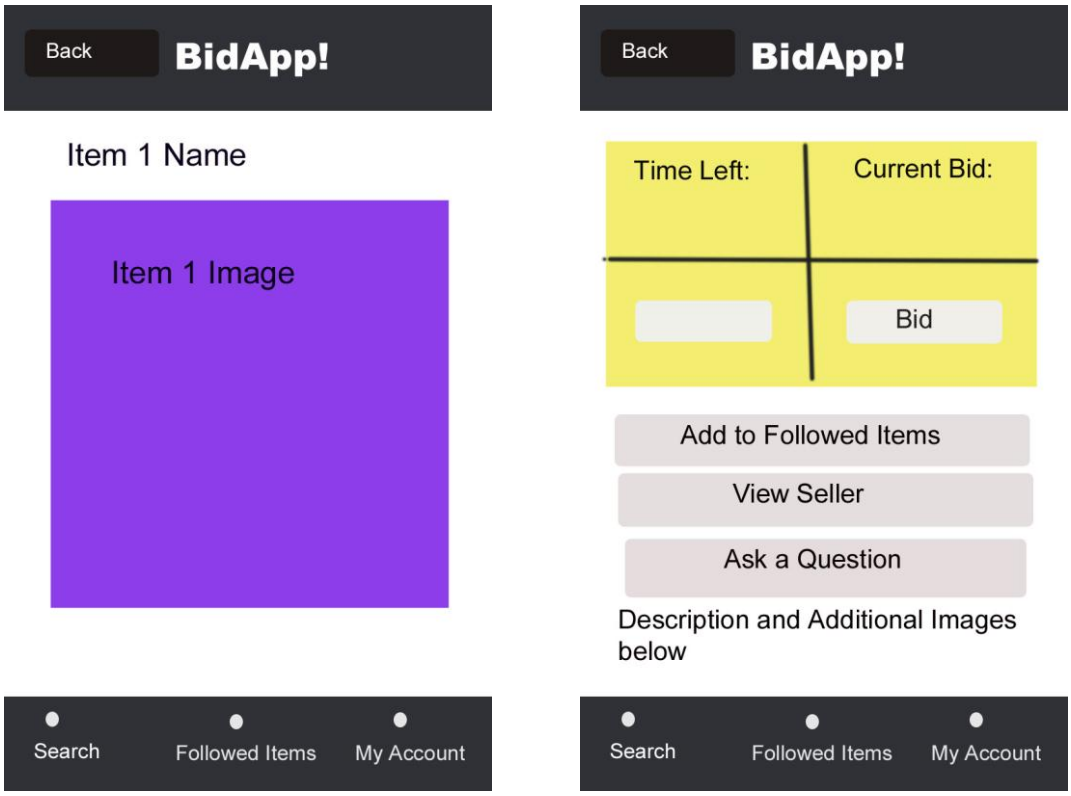


Figure 4.6 – *Individual Item Screens*

Users are also given the option to 'Message Member' which takes the user to the messages page, where the user can start or continue a conversation with that member, as shown in Figure 4.7.

Figure 4.7 – *Messages Screen*

'My Account' contains links to all the pages directly relating to the user's account i.e.: 'My Current Auctions', 'My Expired Auctions', 'My Won Bids', 'My Profile', 'Reviews of Me', 'Account info' and 'Log out'. This is shown in Figure 4.8.

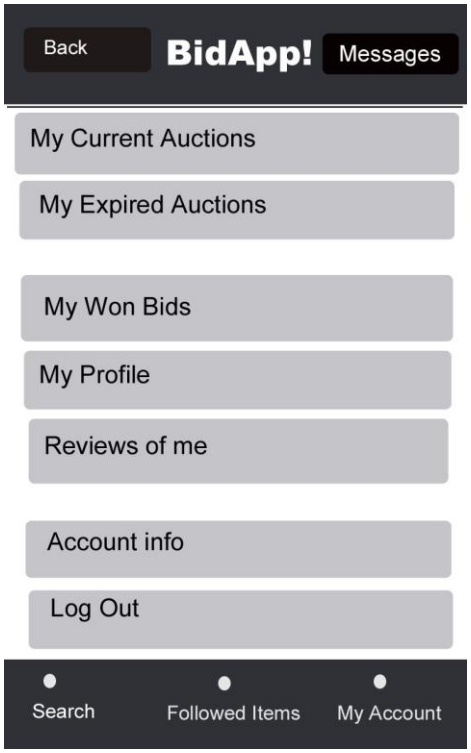


Figure 4.8 – *My Account Screen*

'My Current Auctions' takes the user to a list of items that they have put up for auction but are yet to expire. From this list they can select individual items to see the item's page. There is also the 'Create an Auction' which takes the user to a page where they can fill in a form with all the relevant information (Item Name, Description, Category, Starting Price, Postage and Packaging Price, Auction Expiry Date) as well as upload an image. 'My Expired Auctions' takes the user to a list of items they have put up for auction and have now expired. Selecting one of these items takes the user to a page which shows the winning bid, the address that the sender needs to send the item to, and the option to 'View Buyer' which takes the user to the buyer's homepage. 'My Won Bids' takes the user to a page where the user can see how much they won the bid for, 'Make Payment', 'Give Feedback', 'Message Seller', 'Report Seller', and 'View Seller'. 'My Profile' takes the user to their profile page, which is the same page a user would see if they selected 'View Seller' for that user. This information can be viewed and edited in 'Account info'. 'Reviews of me' takes the user to a page showing the average ratings that they have been given by other users in the form of 1-5 stars. 'Log out' logs out the user and takes the user back to the guest user welcome screen.

4.2. Implementation Proposal

4.2.1 Source Code Layering

The architecture of the project has three layers which are shown in Figure 4.9.

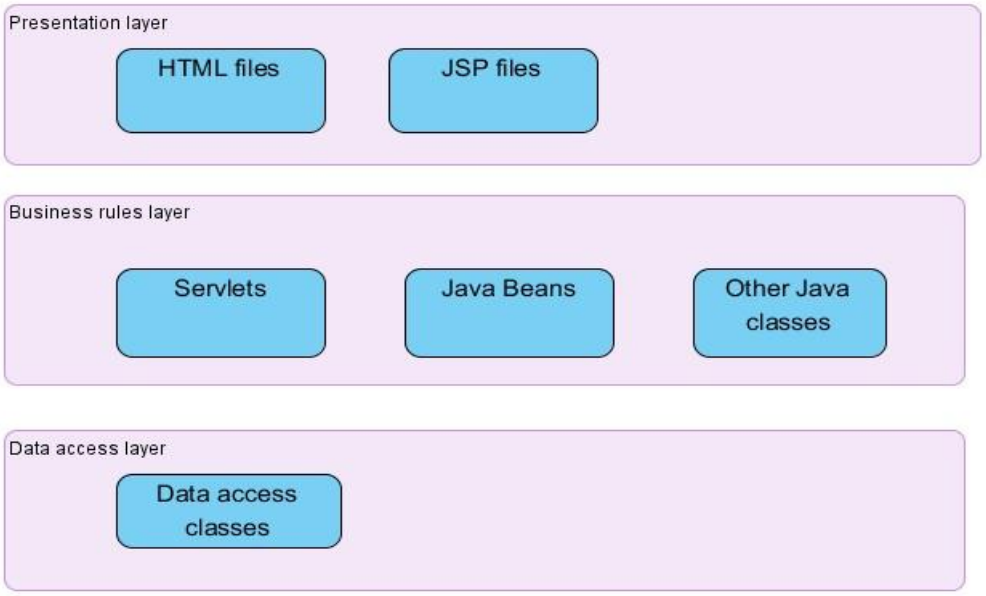


Figure 4.9 – *Layered model of source code*

The *Presentation layer* contains web pages which consist of JSPs (Java Server Pages)[1]. HTML would be implemented first to create the look and feel of the user interface. And then, these pages would be edited by the programmers so they can work properly with the servlets[2] of the application.

The *business rules layer* uses java servlets to control the flow of the application. To start, a servlet usually reads any parameters that are available from the request, which typically comes from the presentation layer (JSPs) and call other Java classes to store or retrieve data from a database. Java beans at this layer are used as a model to read data from the presentation layer and then create or update with java servlets' control, temporarily store and process data.

The *data access layer* contains some Data Access Object (DAO) classes. This layer works for the storing data and do some SQL queries to communicate data with database.

---

[1] Oracle. (2012). *JavaServer Pages Technology.* Available: http://www.oracle.com/technetwork/java/javaee/jsp/index.html. Last accessed 10th December 2012.

[2] Oracle. (2012). *Java Servlet Technology.* Available: http://www.oracle.com/technetwork/java/index-jsp-135475.html. Last accessed 10th December 2012.

4.2.2. Design Patterns

MVC, DAO **a**nd Abstract Factory design patterns are used. The Model-View-Controller pattern is used to build the application and is shown in Figure 4.10 below.

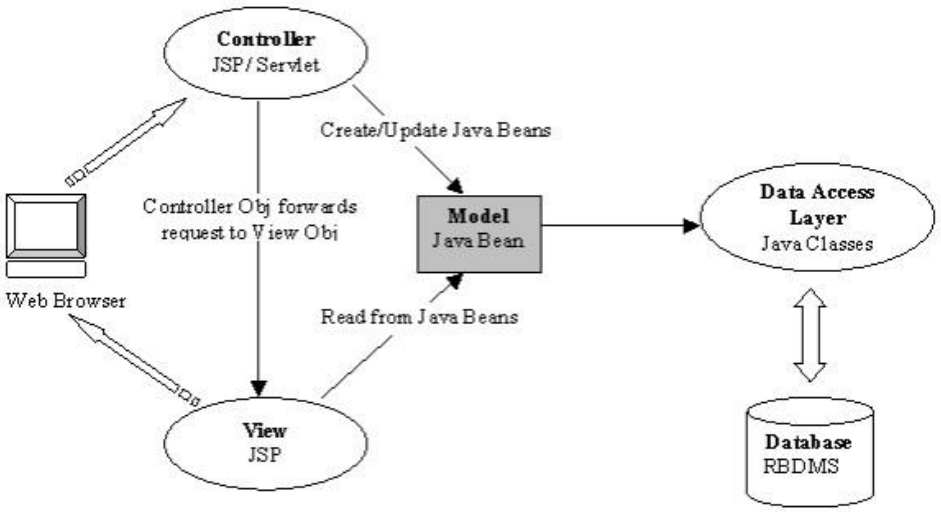


Figure 4.10 – *Model View Controller Design Pattern*

In this pattern the model defines the business layer of the application, which is implemented by JavaBeans. The view defines the presentation layer of the application, which uses HTML within JSPs to present the view to the browser. The controller uses servlets to manage the flow of the application. A servlet usually reads any parameters that are available from the request, which typically comes from the view. If necessary, the servlet updates the model and saves it to the data store. Finally, based on the logic that's coded in the servlet, the servlet forwards the model to one of the several possible JSPs for presentation.

Data Access Object and Abstract Factory design pattern are also used. A Data Access Object (DAO) is used to abstract and encapsulate all access to the database. The DAO manages the connection with the database to obtain and store data. The DAO pattern can be made highly flexible by adopting the Abstract Factory shown in Figure 4.11.

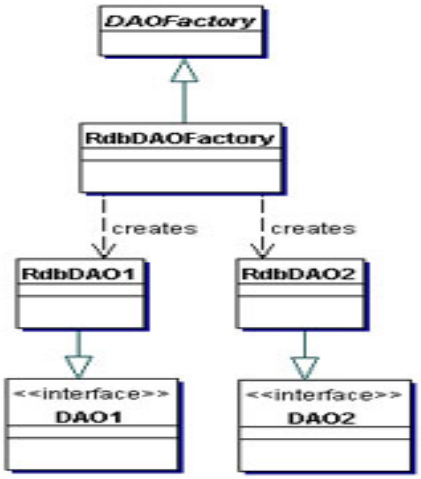


Figure 4.11 – *DAO Design Pattern*

4.2.3. Database

The Database was designed with 11 tables. Figure 4.12 shows the structure of this and how the tables are related.

- User –These are the people using the site.
- Listing – These are the items for sale.
- Bids – These are the bids that have been placed by users on listings.
- Messages – These will be messages sent between users
- Follows – This is the items that a user is following
- Search – This is all the searches that a user has made
- Keywords –These are keywords that are featured in listings to help the user search the listings more easily
- Report - Where users who have been reported and whom has reported them are stored.
- Category - Every item is assigned a category to aid in the users' searching.
- Item Images – These are additional images for each item.
- Feedback – These are all of the feedbacks/reviews given between users

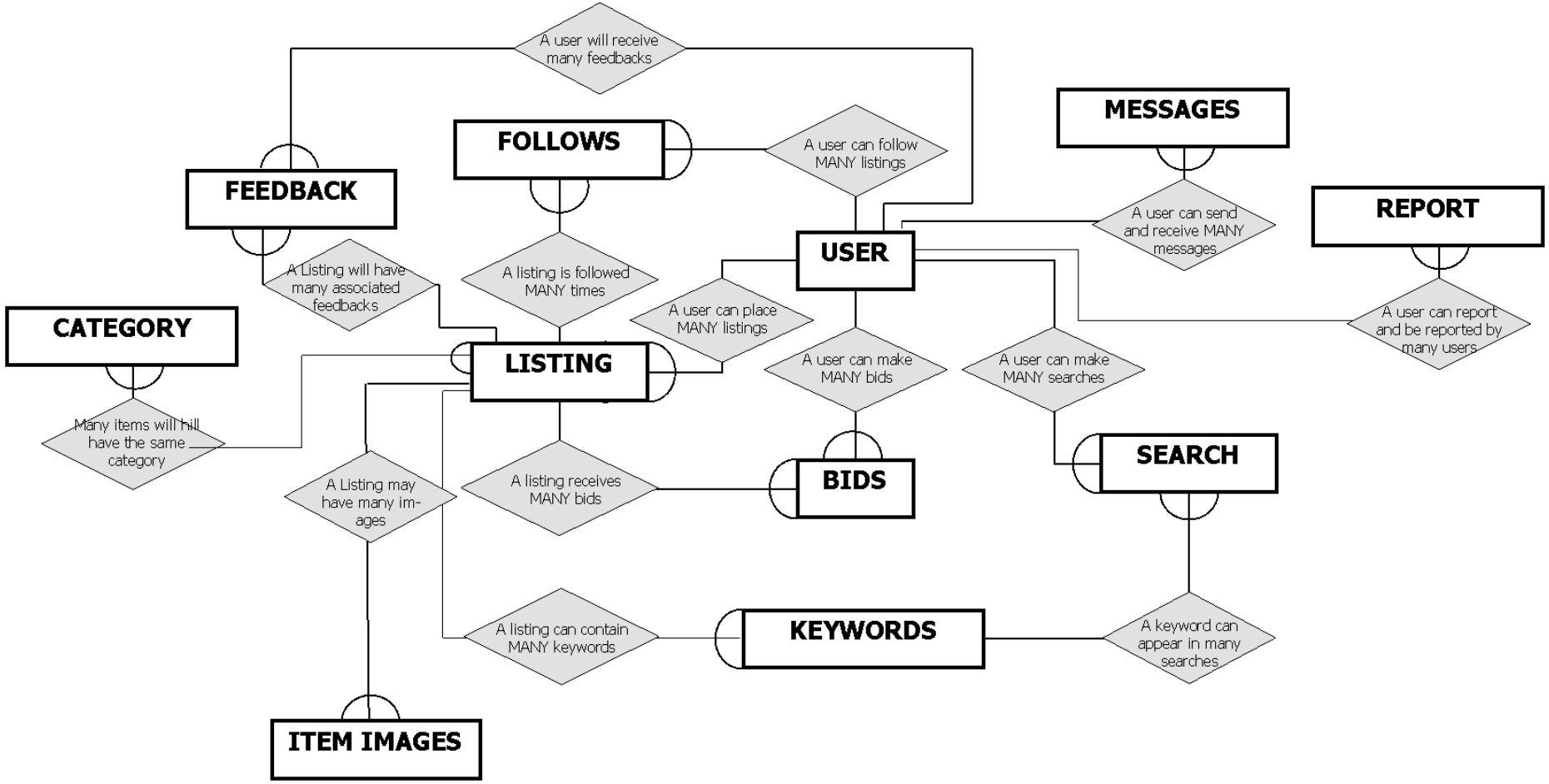


Figure 4.12 - ER Diagram for Relational Database Structure.

5. Implementation

5.1. Programming Language

Java was chosen to be the main programming language, using servlets, JavaServer Pages (JSP) and other java files to build the web application. JSP also contains HTML and JavaScript which are used to build the client side. Servlets and JSP are used to perform the data process as the web application. SQL was used to build the data store, as it is easy to maintain and widely used.[3]

The reason to use the JavaServer web application technology is based on its range of benefits. It offers a clean separation between behaviour and presentation. The separation of logic from presentation also allows each member of a web application development team to focus on his or her piece of the development process and provides a simple programming model to link the pieces.

5.2. Run Environment

To run a Java application, the server must also run a software product known as a servlet/JSP engine, or servlet/JSP container. This software allows a web server to run servlets and JSPs. Sun's Java Enterprise Edition (Java EE)[4] specification describes how a servlet/JSP engine should interact with a web server. Since all servlet/JSP engines must implement this specification, all the engines should work similarly.

For servlets/JSP to work properly, the engine must be able to access the Java Development Kit (JDK) that comes as part of the Java Standard Edition (Java SE)[5]. The JDK contains the Java compiler and the core classes for working with Java. It also contains the Java Runtime Environment (JRE) that's necessary for running compiled Java classes.

5.3. Software

NetBeans[6] was chosen as our Integrated Development Environment (IDEs). NetBeans is open-source, available for free from the NetBeans web site, and runs on all modern operating systems. In addition, NetBeans comes with a bundled version of the GlassFish[7] application server, a free Java EE

[3] sql.org. (2012). *Welcome.* Available: http://www.sql.org/. Last accessed 10th December 2012.
[4] Oracle Corporation. (2009-2011). *JavaTM Platform, Enterprise Edition 6 API Specification.* Available: http://docs.oracle.com/javaee/6/api/. Last accessed 10th December 2012.

[5] Oracle Corporation. (1993, 2012). *Java™ Platform, Standard Edition 7 API Specification.* Available: http://docs.oracle.com/javase/7/docs/api/index.html. Last accessed 10th December 2012.Oracle Incorporation. (2012). *Netbeans IDE.* Available: http://netbeans.org/. Last accessed 10th December 2012.

[6] Oracle Incorporation. (2012). *Netbeans IDE.* Available: http://netbeans.org/. Last accessed 10th December 2012.
[7] Glassfish. (2008-2012). *Glassfish Community.* Available: http://glassfish.java.net/. Last accessed 10th December 2012.

server which is very powerful. However, when distributed by Sun, this Java EE server is sometimes called the Sun Java System Application Server. For this reason it is very convenient to use NetBeans as our development tool.

MySQL was used as our project's database management system because it is one of the most popular systems for Java Web applications. It was also free for the purpose of our project. It is easier to use compared with products like Oracle Database or Microsoft SQL Server and can run on most modern operating systems, while Microsoft SQL Server runs only on Windows. Even though it's free for most uses, MySQL provides most of the features that should be expected from a modern relational database management system (RDBMS). In particular, it provides support for Structural Query Language (SQL), the industry standard. It also provides support for multiple clients and enhances connectivity and security.

5.4. Source code architecture Design

There are four folders in the bidapp web application as shown in Figure 5.1. The JSP web pages are stored in the Web Pages folder as presentation layer and the java files which process the data and implement functions are stored in the Source Packages folder.

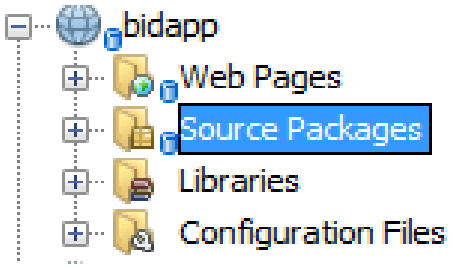


Figure 5.1 - Project Folders

There are six packages in the Source Packages folder which are shown in Figure 5.2. "com.auction.action" package store the java servlet files and "com.auction.logic" package stores the java bean files. The "com.auction.entity" package stores the entity java files corresponding to the entity of database relationship with accessor and mutator methods and some data fields as attributes. These three package are as *business rules layer*.

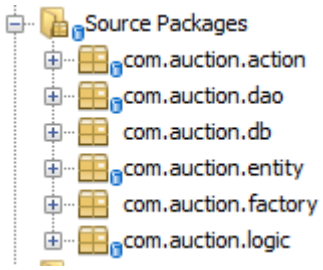


Figure 5.2 - Source Code Packages

The “com.auction.dao” package stores Data Access Objects (DAOs) which provide an abstract interface to the database and not used by JSPs or servlets directly. DAOs provide some specific operations without exposing the detail of the database by mapping calls to the business rules layer. The “com.auction.db” package is used to access and manage the linking between server and database. Every DAO file uses this package’s class to communicate with the database. The “com.auction.factory” packages store an IDAO interface so that all DAOs implement the methods in IDAO. These three packages are known as the *data access layer*.

5.5. SQL Implementation

SQL was implemented for the back end of the system, which would communicate to the system through the Data Access Layer. All the tables discussed in section 3.2.3 were implemented.

5.6. HTML5/ jQuery

HTML5 was implemented as part of the presentation layer and would act as the main implementation of the system’s user interface. In order to accommodate for the requirement of the system to work on mobile devices, a decision to combine HTML5 with jQuery mobile libraries was made. These libraries would enhance the user interface by combining HTML5 elements with JavaScript that enables dynamic content to be displayed according to the user’s interactions. Furthermore, jQuery mobile provides tested libraries that may be used to create a single uniform user interface capable of operating in an identical fashion across all mobile platforms. The libraries were designed for efficiency, requiring little amounts of memory to transfer across possibly slow mobile internet connections. The libraries would also remain easily interpreted by less powerful mobile browsers, ensuring that the speed of the user interface remained uniform across all platforms. An example of the use of jQuery in combination with HTML5 in the implementation can be seen below in Figure 5.3.

Code Sample : Website Header

```
1. <link rel="stylesheet"
   href="http://code.jquery.com/mobile/1.1.1/jquery.mobile-1.1.1.min.css"
   />

2. <script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>

3. <script src="http://code.jquery.com/mobile/1.1.1/jquery.mobile-
   1.1.1.min.js"></script>

4. <!--Header-->

5. <header data-role="header" data-position="fixed">

6. <a data-role="button" data-icon="back" data-rel="back">Back</a>
```

```
7. <h1><a href="#homePage"><img src="Logo.png" width=80%></a></h1>
8. <a href="#messagesPage">Messages</a>
9. </header>
```

Figure 5.3 - *HTML5 and jQuery example source code*

Figure 5.4 shows the HTML5 which was created using the code shown in Figure 5.3.

Figure 5.4 - *HTML5 produced by code shown in Figure 5.3*

In addition, in order to ensure that the HTML5 webpage and user interface would facilitate the full size of the phone screen and enhance the usability of the system, HTML5 meta-tags were also used to create a 'viewport'. This is shown in Figure 5.5.

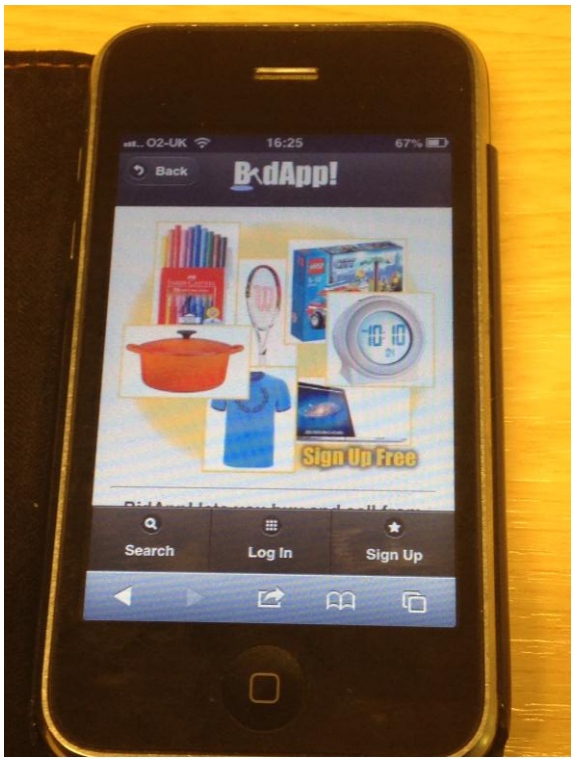


Figure 5.5 - *HTML5 meta-tags to create a viewport*

5.7. Implementation Steps

JSP web pages were created based on the HTML prototype. The Register function was implemented first as a sample to link to the database. To begin, the entity classes with attributes data fields were created, and then accessor and mutator methods corresponding to the entities of the database were also created. And then some bean files were created to link to the database. After the register function was finished, the web application could link to the database. Following this; Login, Logout and change account information functions were implemented. The implementation steps and contents show below in Figure 5.6.

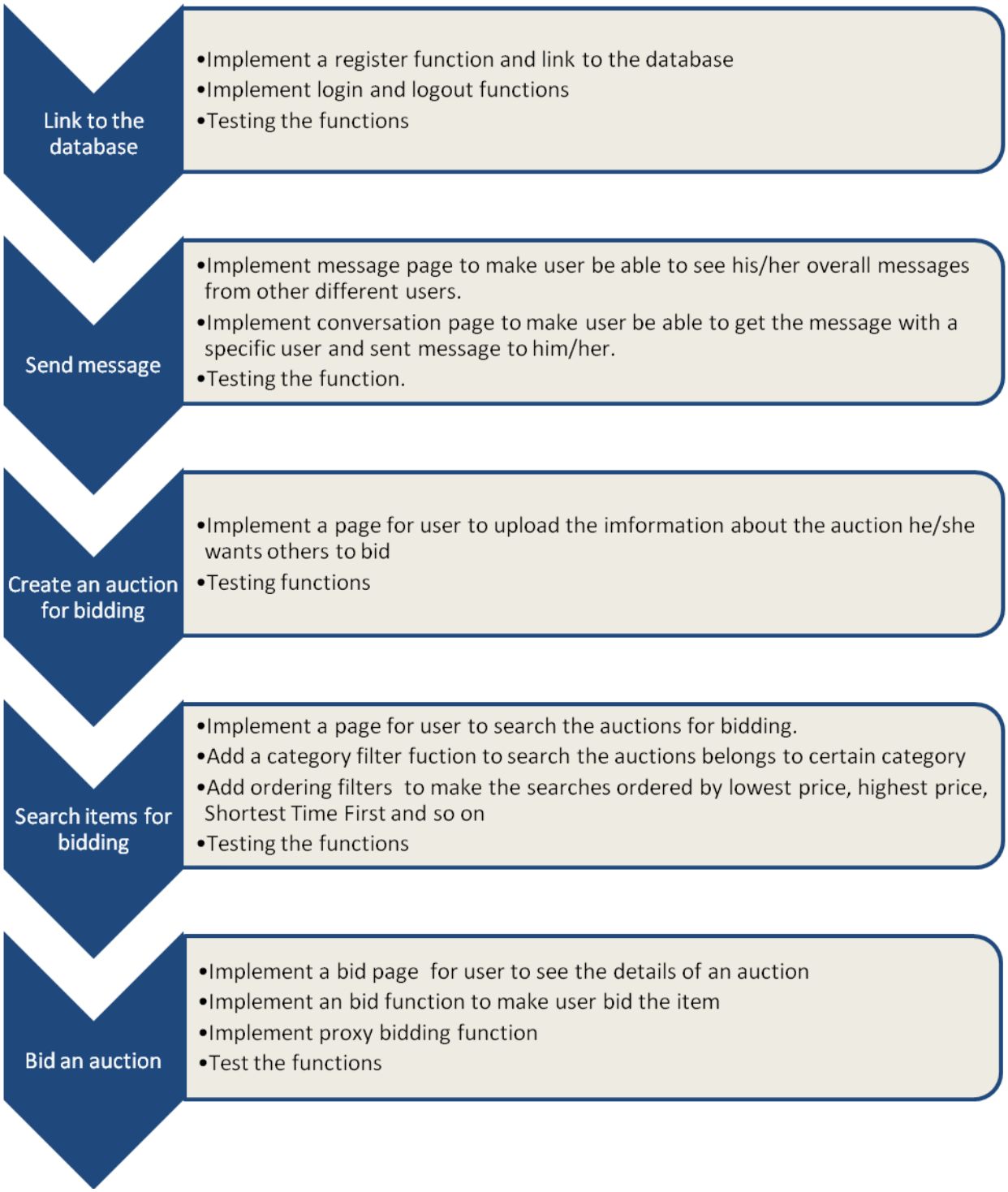
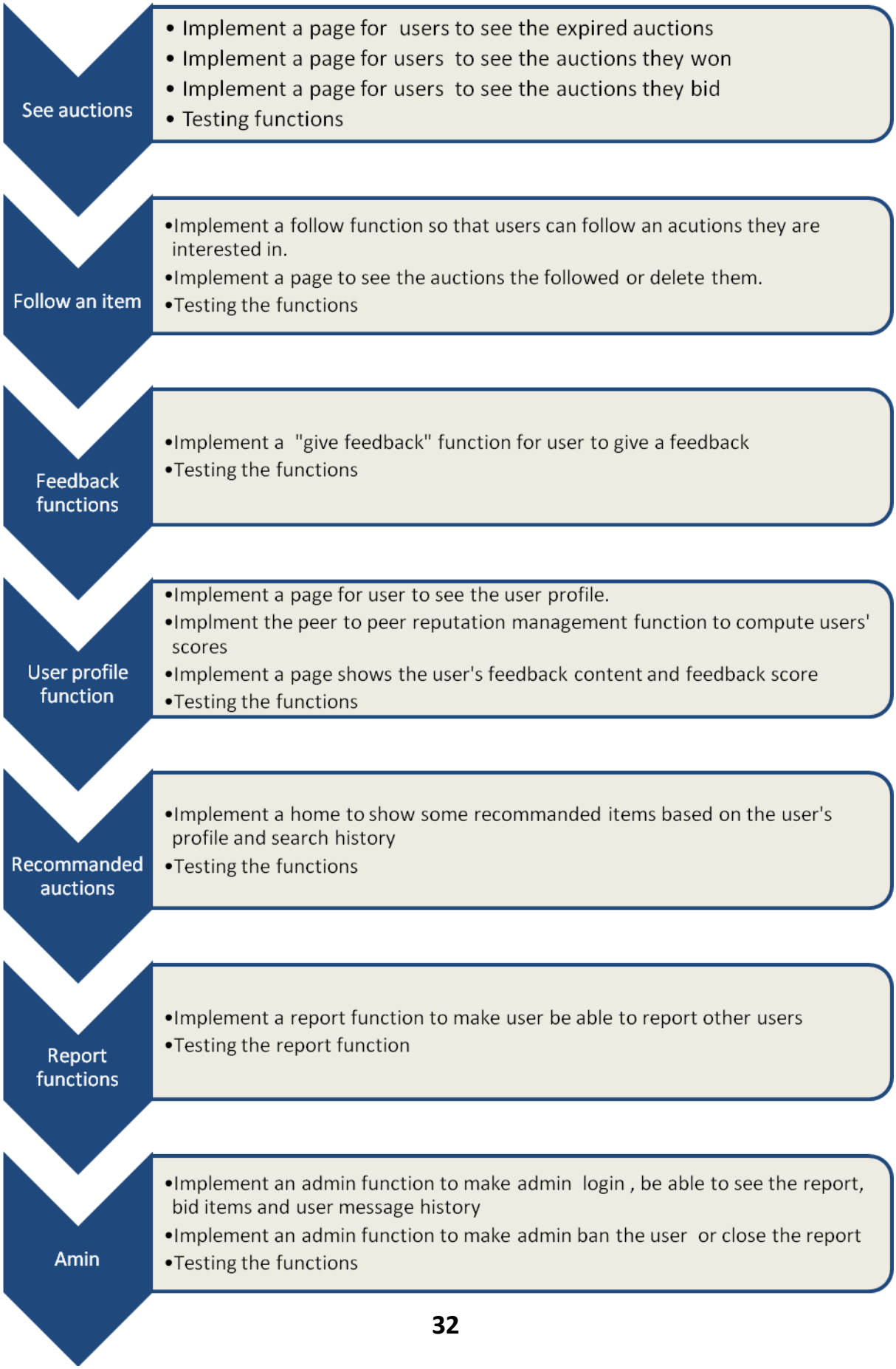


Figure 5.6 - *Implementation Steps*

(Continued on following page)


See auctions
• Implement a page for users to see the expired auctions
• Implement a page for users to see the auctions they won
• Implement a page for users to see the auctions they bid
• Testing functions
Follow an item
•Implement a follow function so that users can follow an acutions they are interested in.
•Implement a page to see the auctions the followed or delete them.
•Testing the functions
Feedback functions
•Implement a "give feedback" function for user to give a feedback
•Testing the functions
User profile function
•Implement a page for user to see the user profile.
•Implment the peer to peer reputation management function to compute users' scores
•Implement a page shows the user's feedback content and feedback score
•Testing the functions
Recommanded auctions
•Implement a home to show some recommanded items based on the user's profile and search history
•Testing the functions
Report functions
•Implement a report function to make user be able to report other users
•Testing the report function
Amin
•Implement an admin function to make admin login , be able to see the report, bid items and user message history
•Implement an admin function to make admin ban the user or close the report
•Testing the functions

Many agile practices are used in the implementation process. Pair programming was used in the programming process to produce more efficient code and though out better methods. In addition, Test Driven Development (TDD) was used to reduce the bugs and make the program more robust by using Unit Testing before programming as shown in Figure 5.7 below.

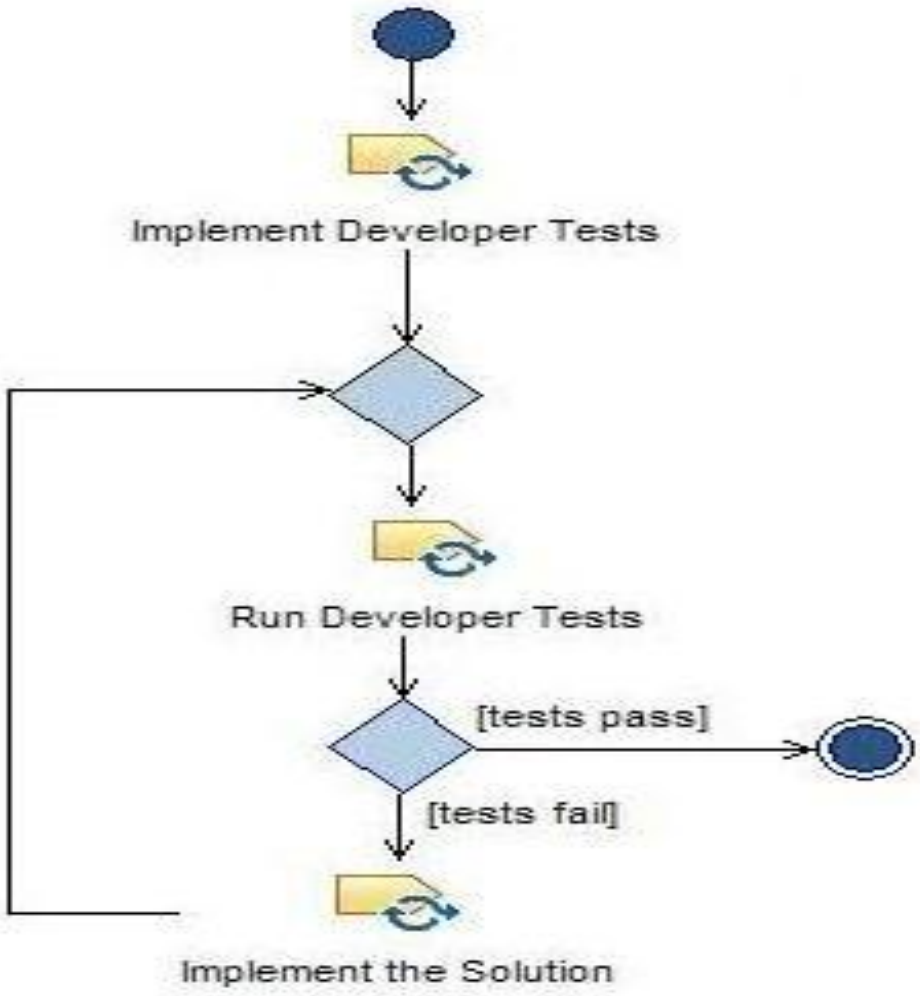


Figure 5.7 - *Test Driven Development*

5.8. Code implementation samples

5.8.1. Proxy bidding

The system also gives users the option to enable proxy bidding (also known as automated bidding). When the user places a bid, they enter the maximum amount they are willing to pay for the item. At this stage, the seller and other bidders don't know if there is a maximum bid. If another user's bid is higher than the previous bid, the system will automatically compare the other user's bid with the user's maximum bid. If the user's maximum bid is higher, they will remain the winning bidder and the current bid price will automatically increase 0.05 pounds. [8]. Partial implementation of this is shown in Figure 5.8.

Partial Code : **Proxy bidding**

```
//maxbid is the highest bid so far
1.    BidsBean bbean = new BidsBean();
2.    Bids maxbid =
   bbean.queryListingByHighestPrices(listing.getId());
3.
4.    //Get maxbid's bid amount and proxy bidding price
5.    double maxbidAmount = maxbid.getBidAmount();
6.    double maxbidProxy = maxbid.getMaximumBid();
7.    /* proxy bidding */
8. // If user choose use proxy bidding
9. if (proxy > 0){
10.      // compare user's bid value with max bid's value
11.      if (maxbidAmount < bidValue){
12.
13.         // compare user's bid with max bid's proxy bidding value
14.           if (maxbidProxy > bidValue){
15.
16.              // compare user's proxy value and max bid's proxy
   bidding values
```

[8]WiseGeek. (2013). *How Proxy Bidding Works.* Available: http://www.wisegeek.com/what-is-proxy-bidding.htm

```
17.              if (maxbidProxy > proxyValue){
18.                 maxbid.setBidAmount(proxyValue + 0.01);
19.                 bbean.updateBidAmount(maxbid.getBidId(),
   bid.getBidAmount());
20.               }else{
21.                  bid.setBidAmount(maxbidProxy + 0.01);
22.
```

Figure 5.8 - *Partial proxy bidding code*

5.8.2. Peer-To-Peer network reputation management system

Figure 5.9 shows the calculation of a user's reputation within the system.

$$\mathbf{R} = \sum_{i=1}^{n} \frac{W(i) * O(i)}{n}$$

Figure 5.9 – *Calculation of reputation*

A user's trust, is based on both the feedback scores and the buyer's reputation score who gave the feedback. The queried trust rating is the average of the evaluated trust ratings, weighted by the credibility of their senders. That is, if peer A issued a trust query on peer B, and the responses of peers R1, R2…Rn and A's credibility rating for R(i) is w(i) and R(i)'s trust rating for B is o(i), then A's queried trust score on B is R. The queried distrust rating is calculated in the same fashion, using the respondents' distrust ratings of B. This is shown in Figure 5.10.

Partial Code : **Trust and reputation management system in peer to peer network**

```
1. for (int i = 0; i < size; i++){

2.  feedback = feedbacks.get(i)  *

(newUserinfoBean().queryUserinfoByUserid(feedback.getSenderId())
.getUserScore / 5) ;

3.        score += feedback.getFeedbackScore();

    }

4. if (size == 0){

5.   score = 0 ;

6. }

7. else{

8.   score /= size ;

9. }
```

Figure 5.10 - *Partial code about trust and reputation management*

The piece of code use feedbacks.get(i) get i's trust rating for B and (new UserinfoBean().queryUserinfoByUserid(feedback.getSenderId()) .getUserScore / 5) get the credibility rating for R(i). Add them together and then divide the size to get the feedback score.

5.8.3. Recommended item

To get a recommended item for the user, the system records the keywords the user has searched for and stores them in the database. Every time a user refreshes the home page or clicks the "show something else" button, the system will get the keywords list which the user most recently searched and randomly selects one to show as a recommended item. If the user hasn't searched anything or the keywords cannot map anything, the system will show an item randomly. This is shown in Figure 5.11.

Partial Code : **Recommended auction**

```
1. // Get the Login user

2. Userinfo user = (Userinfo)session.getAttribute("loginUser");

3.

4.   // Get users' searches history as a list from database

5.    List<UserSearches> searches = new
```

```
   UserSearchesBean().queryByUserId(user.getUserid());
6.    Listing listing = null;
7.    double price = 0;
8.
9.    if (searches != null){
10.      //Get a searches keyword from the search history randomly
11.      String searchText =
   searches.get(((int)Math.floor(Math.random()*searches.size()))).getSearchText
   ();
12.      //Query listings from the database which contain the keyword
13.      List<Listing> listings = new
   ListingBean().queryListingByListingTitle(searchText);
14.
15.       if (listings != null && listings.size() != 0){
16.           // Get a listing from the listings list randomly
17.           listing =
   listings.get(((int)Math.floor(Math.random()*listings.size())));
18.
19.            // The following code get the listing's current price
20.            Bids bid = new
   BidsBean().queryListingByHighestPrices(listing.getId());
21.
22.             if (bid == null)
23.                price = listing.getStartPrice();
24.               else
25.                  price = bid.getBidAmount();
26.        }
```

Figure 5.11 - *Recommended Auction*

5.8. Establishing a Server

The entire implementation of the server, which supports the interaction between the client side HTML and server side JSP servlets, is packaged and hosted on an application server. In addition to distributing the application itself, web server functionality is required in order for users to connect across the internet and hence use the services from other non-local and public networks. We have used an implementation of Windows Server 2003, which has an allocated and unique FQDN[9] of *"www.night.cs.nott.ac.uk"*. This allows users to access the services over the internet. The chosen operating platform allows us greater flexibility in terms of remote administration (RDP[10]) and offers a more powerful and adaptable web page administration. The web server in use is Glassfish, chosen due to its small footprint and capability of handling JSP servlets. The web application itself is an implementation of Java and Java Server Pages, created and interconnected through the use of NetBeans IDE, which offers a powerful interface for coding web packages and applications. The database is a MySQL database, which is run locally on the server. It is accessed, updated and modified directly through the use of JSP servlets, embedded within the webpages; for example, when a new user creates an account, the JSP servlet will return the form data to the server, which will in turn log this and write the data to the database, to act as a permanent storage solution. The stored data is then universally accessible to all other components of the software; for example, the recent new member can view their personal details, and amend if necessary.

The easiest and most efficient method to implement the above was to install and use a version of NetBeans Enterprise Edition, which doubles up to complete a number of tasks and services required of our project. Naturally, NetBeans stores the entire archive of web pages (both HTML, CSS and all associated elements). In addition, it runs an implementation of the Glassfish web server in the background. All of our JSP servlets and scripts are executed through the IDE, so they directly interact and act as a medium between the database and the client side web application. NetBeans operates as a background process, that is continuously running on the server, providing users with 24/7 access. The FQDN allows both desktop and mobile users to easily navigate to the web page to access the application. The default web page on the server is set to always direct users to the Home Screen, the design and interface of which is different depending on whether the user is an existing member, new member, or currently logged in. For security, only the required ports (used by the necessary applications) are accessible, all other ports are blocked. This increases the security, and decreases the risk of potential security threat.

[9]Webopedia. (2013). *FQDN.* Available: http://www.webopedia.com/TERM/F/FQDN.html

[10]Microsoft Support. (2013). *Remote Desktop Protocol.* Available: http://support.microsoft.com/kb/186607

6. What Was Achieved

In this chapter, we shall look back upon the SRS document (see Appendix 9.1), and review if (or to what extent) each requirement was met, in order to gauge the overall success of the project.

6.1 SYSREQ 1 - Users can create their own personal account

SYSREQ 1.1 - Users can create a username and password ✓

Figure 6.1 shows how users are able to set their own username and password during the signup process.

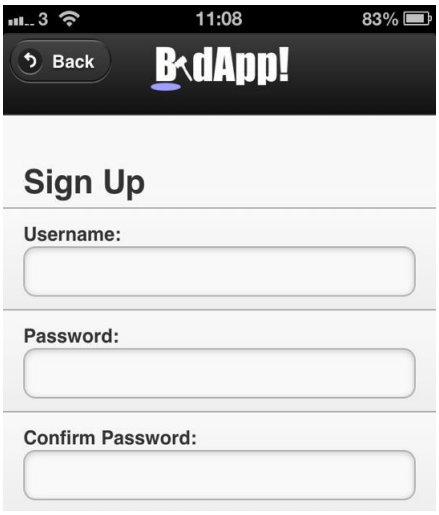


Figure 6.1 – *Evidence of satisfied SYSREQ 1.1*

SYSREQ 1.2 - Users can log in with a username and password ✓

Figure 6.2 shows how users can then use their chosen username and password to login to their BidApp account when they are not already logged into the system.

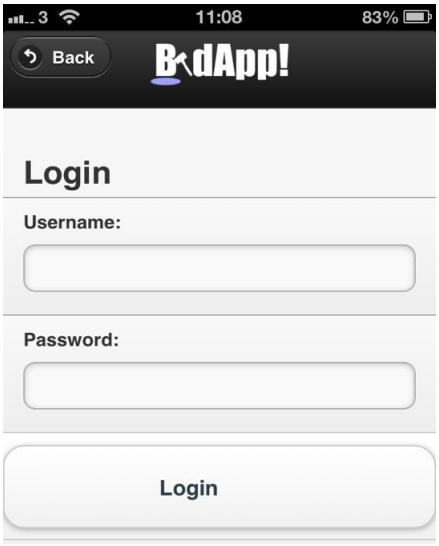


Figure 6.2 – *Evidence of satisfied SYSREQ 1.2*

SYSREQ 1.3 - Users can add personal details to their account ✓

Figure 6.3 shows how users are able to add First Name, Last Name, Email Address and Home Address details to their account during the initial signup process, and can later modify them through the "My Account" section of the site.

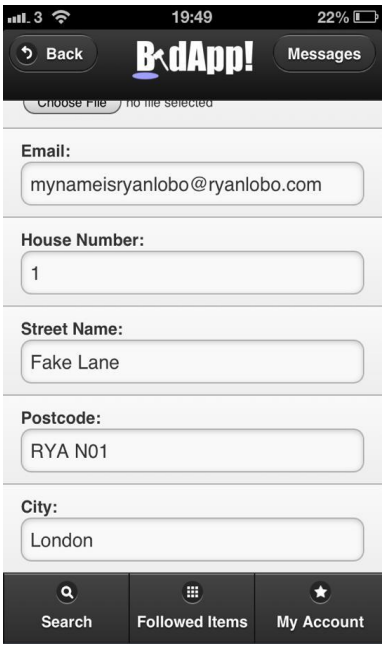


Figure 6.3 – *Evidence of satisfied SYSREQ 1.3*

SYSREQ 1.4 - Users can add a personal photo to their profile ✓

Figure 6.4 shows how users have the option to add a personal profile picture to the account during the initial signup process, and can later modify it through the "My Account" section of the site.

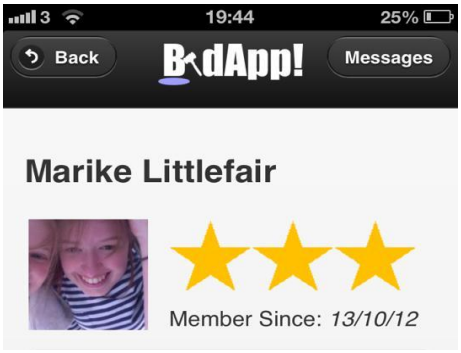


Figure 6.4 – *Evidence of satisfied SYSREQ 1.4*

SYSREQ 1.5 - Users can add banking/financial details to their account

This was not fully explored, as the ability to do financial transactions would not be supported by BidApp.

SYSREQ 1.6 - Users can message other users from their account ✓

Figure 6.5 shows how users can send messages to other users using the Messages area on the site.

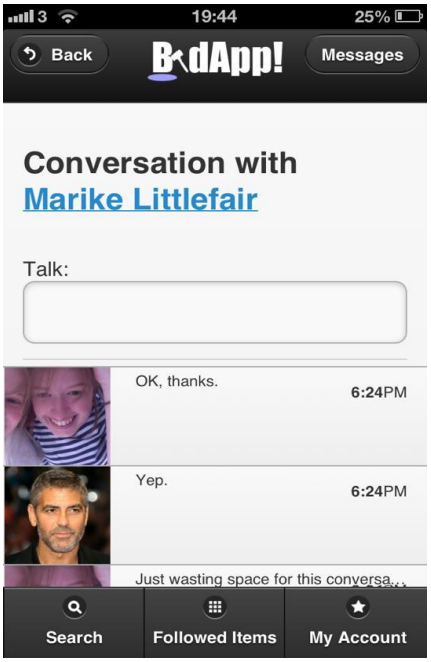


Figure 6.5 – *Evidence of satisfied SYSREQ 1.6*

SYSREQ 1.7 - Users can view messages received from other users ✓

Figure 6.6 shows how users can also receive messages from other users using the Messages area on the site.

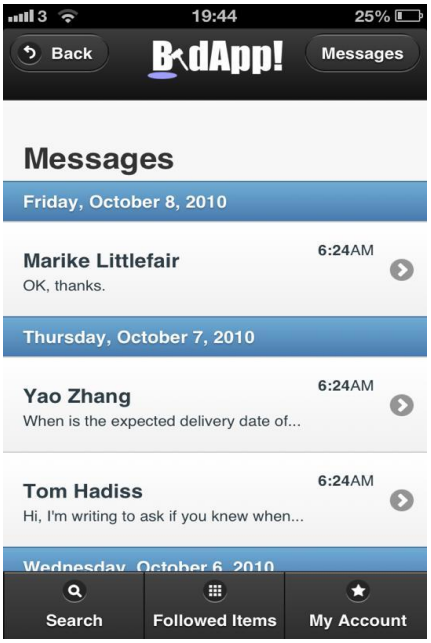


Figure 6.6 – *Evidence of satisfied SYSREQ 1.7*

SYSREQ 1.8 - Users can change all information uploaded to their account ✓

Figure 6.7 shows how users can modify any of their personal details and photos from the "My Account" section of the site.

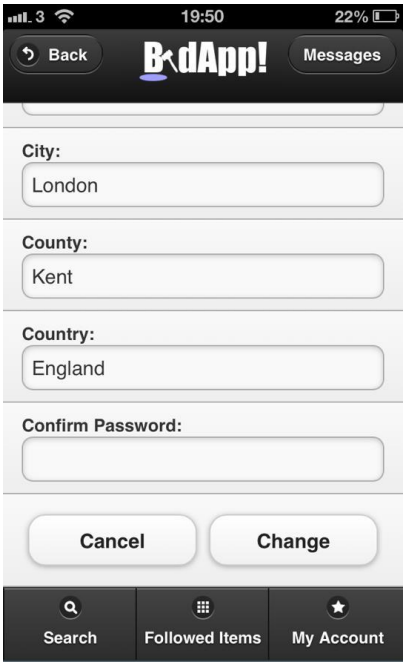


Figure 6.7 – *Evidence of satisfied SYSREQ 1.8*

SYSREQ 1.9 - Users can logout ✓

Figure 6.8 shows how if a user is logged in to BidApp, they may logout at any point from the "My Account" section of the site.

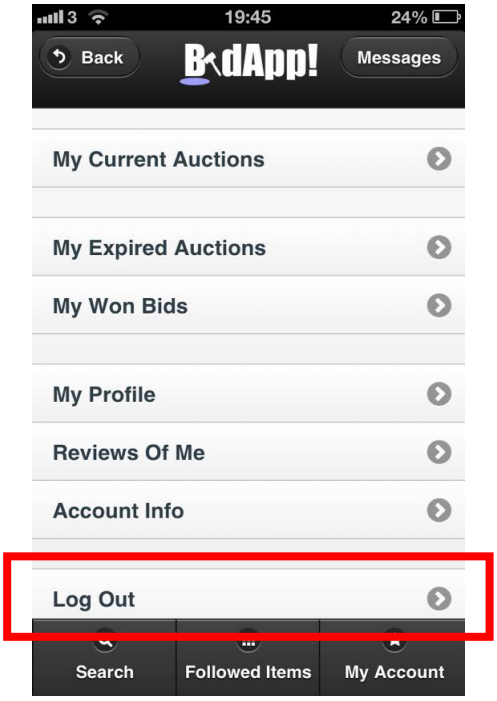


Figure 6.8 – *Evidence of satisfied SYSREQ 1.9*

6.2. SYSREQ 2 - Users can view items on auction

SYSREQ 2.1 - Users can search for items on auction by category ✓

Figure 6.9 shows how upon searching for an item, the user can select a category from a drop down list of options such as "Cars", "Coins etc.

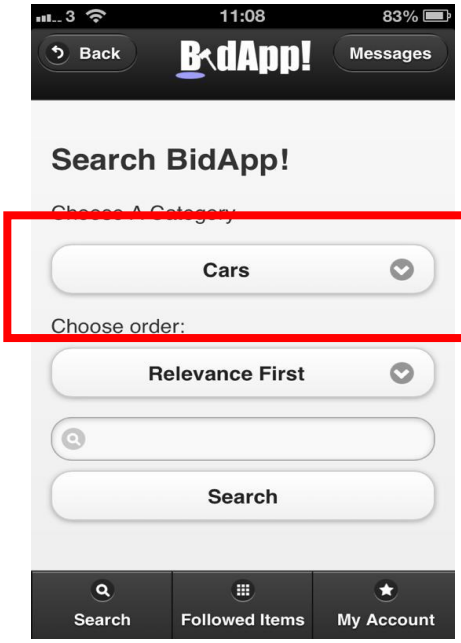


Figure 6.9 – *Evidence of satisfied SYSREQ 2.1*

SYSREQ 2.2 - Users can search for items on auction by keywords ✓

Figure 6.10 shows how users can enter keywords into a search bar to search for items. The search will return items similar to what they have searched for, e.g. searching for "Football", may return "American Football DVD" etc.

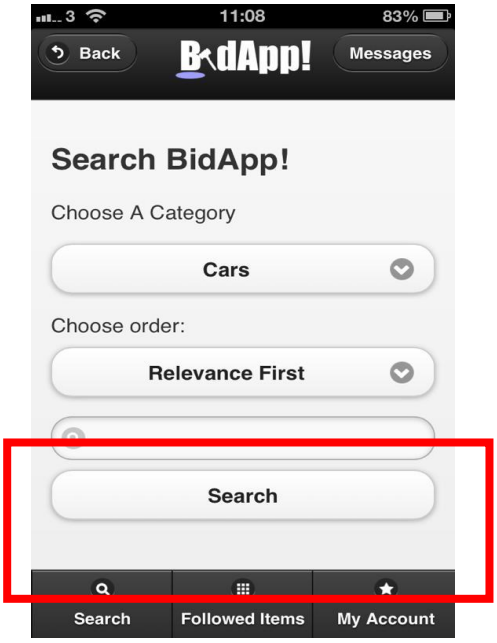


Figure 6.10 – *Evidence of satisfied SYSREQ 2.2*

SYSREQ 2.3 - Users can search a list of items suggested by the system ✓

The system will keep a track of what items the user has searched for, followed or bought. Figure 6.11 shows how using this data, it is able to recommend suitable items to the user which can then be browsed through by the user.

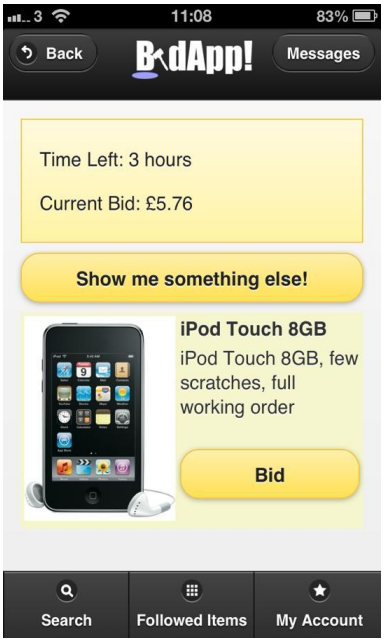


Figure 6.11 – *Evidence of satisfied SYSREQ 2.3*

SYSREQ 2.4 - User is given information regarding item on auction when it is selected from the search ✓

Figure 6.12 shows how when the list of items matching the users search is returned, the user can click on any of these to view additional information. On each item's page, the user can view a description and images of the item, and is also presented with options to:

- *Make a bid on the item (and a maximum bid for the optional process of proxy bidding)*
- *Add the item to the user's followed item list*
- *View the item's seller's profile page*
- *Message the seller about the auction*

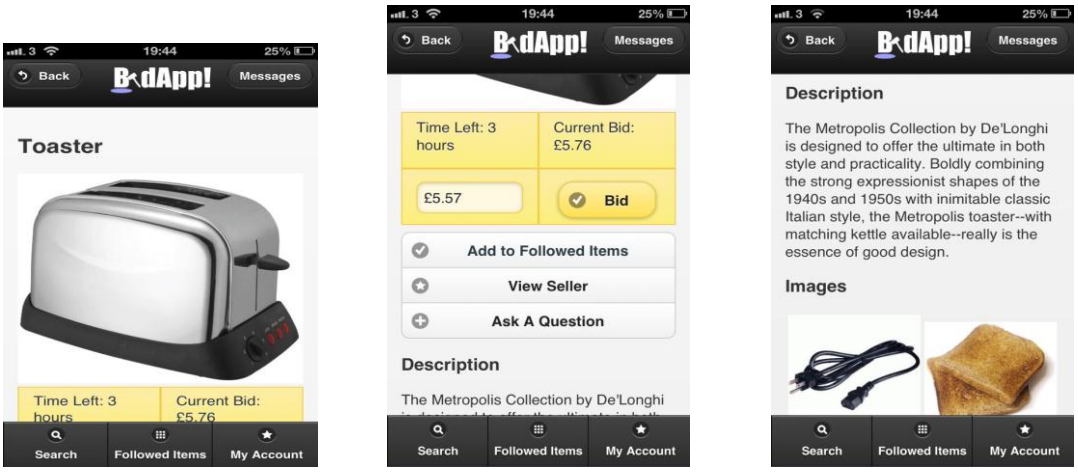


Figure 6.12 – Evidence of satisfied SYSREQ 2.4

6.3. SYSREQ 3 - Users can follow items on auction

SYSREQ 3.1 - Users can add items on auction to their followed items ✓

Figure 6.13 shows how on the page for each item, the user is given an option to follow the item, which will immediately place the item in the user's "Followed Items" list.

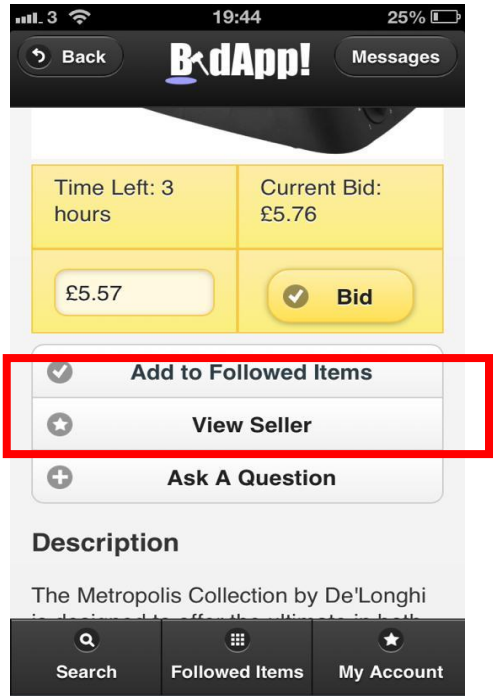


Figure 6.13 – *Evidence of satisfied SYSREQ 3.1*

SYSREQ 3.2 - Users can view items on auction that they are following ✓

Figure 6.14 shows how each user has a "Followed Items" list which simply displays any items they have chosen to follow.

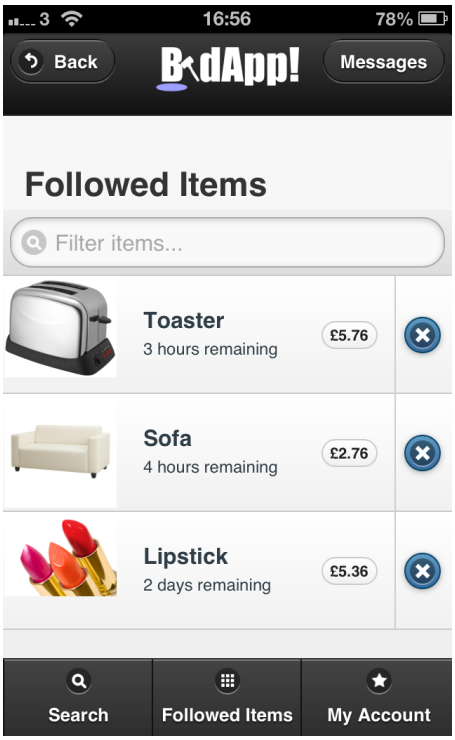


Figure 6.14– *Evidence of satisfied SYSREQ 3.2*

6.4. SYSREQ 4 - Users can bid on items

SYSREQ 4.1 - Users can place a bid on an item, specifying an offer which must be higher than the current highest offer at that time

Figure 6.15 shows how users can bid on any auction currently on auction by selecting Make Bid on the item's page, and entering a value that is higher than the current winning bid.

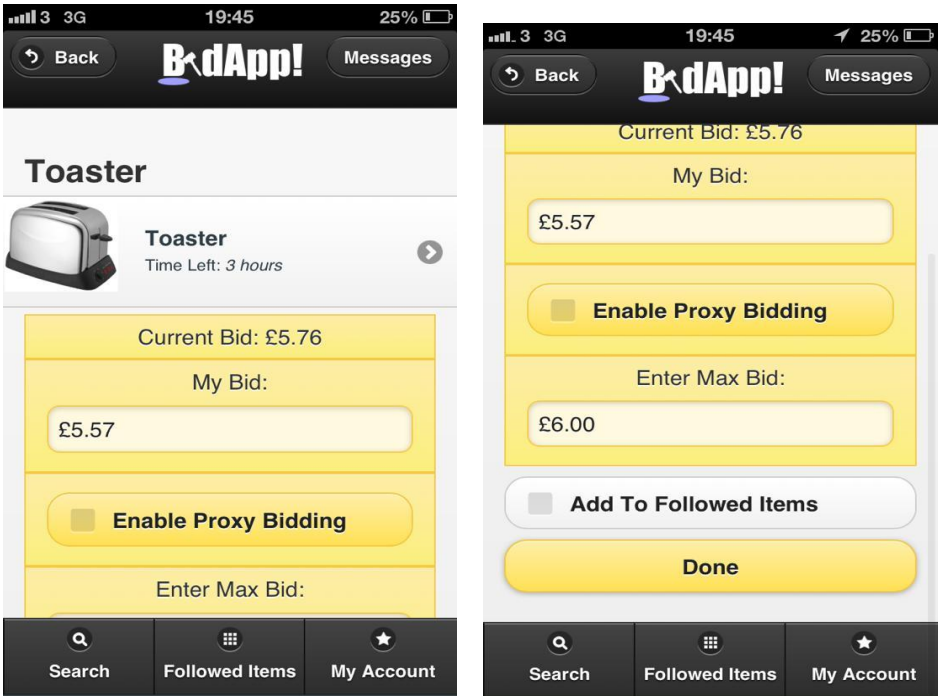


Figure 6.15 – *Evidence of satisfied SYSREQ 4.1*

SYSREQ 4.2 - Users can proxy bid, whereby they submit a maximum bid, and the system will bid for the user whenever they are outbid until this maximum is reached ✓

Figure 6.16 shows how when submitting a bid, users are also given the option to enter a maximum bid as well which must be equal or equivalent to their bid. The system will then continue to bid on the user's behalf until the maximum bid amount is reached.

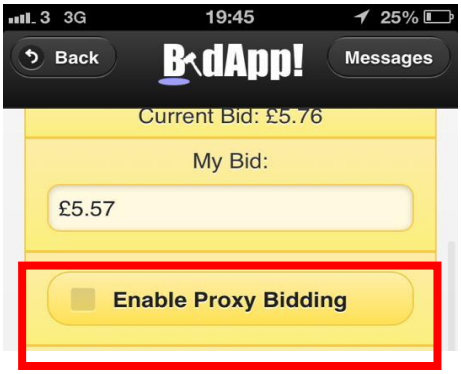


Figure 6.16 – *Evidence of satisfied SYSREQ 4.2*

SYSREQ 4.3 - Users can be alerted as to when they have been outbid for an item ✓

Users will be sent a message by the system when they have been outbid for any item they were previously winning.

SYSREQ 4.4 - Users are alerted that they have won a bid if they hold the maximum bid after the time of the auction's expiry ✓

Users will be sent a message saying that they have won an item if they hold the winning bid at time of auction closure.

6.5. SYSREQ 5 - Users can manage their transactions

SYSREQ 5.1 - Users can buy items they have won in a transaction using the financial details stored with their account

As stated in SYSREQ 1.5, financial transactions would not be supported by BidApp and therefore this was not explored.

SYSREQ 5.2 - Users can view details of all previous transactions

As stated in SYSREQ 1.5, financial transactions would not be supported by BidApp and therefore this was not explored.

6.6. SYSREQ 6 - Users can sell items

SYSREQ 6.1 - Users can place an item on the system which they wish to sell ✓

Figure 6.17 shows how users can place an auction on BidApp by selecting the "My Current Auctions" from the "My Account" section of the site. They can give information about the item such as: Name, Description, Category, Starting Price, Postage and Packaging, and then an expiry date (the time will always be midnight on the specified date). Users must also agree to Terms and Conditions before their auction will go live.

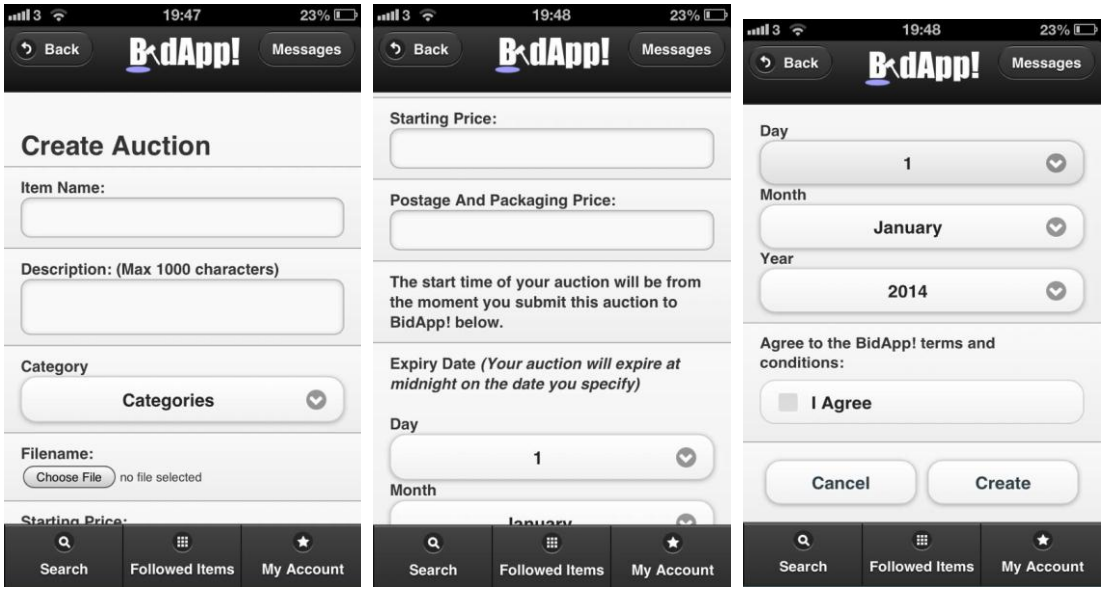


Figure6.17 – *Evidence of satisfied SYSREQ 6.1*

SYSREQ 6.2 - Users can add photos of the item, with a maximum of 3 photos per item

Figure 6.18 shows how users can add one photo initially when first placing an item up for auction. They can then add any others they wish to by modifying the auction. A maximum was eventually decided against as it was deemed the users should be able to add as many as they like

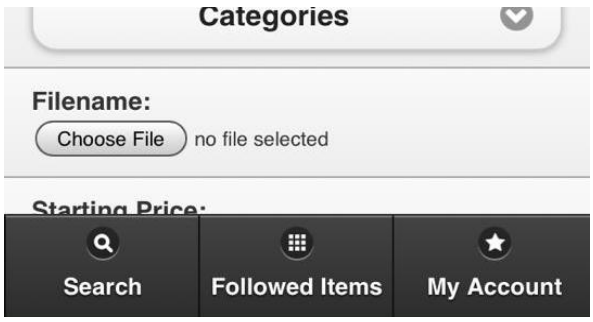


Figure 6.18 – *Evidence of satisfied SYSREQ 6.2*

SYSREQ 6.3 - Users can add a starting bid for the item which the bidding will commence from, otherwise the bidding isgin from a default price ✓

Figure 6.19 shows how users must add a starting price when they go through the process of beginning an auction.

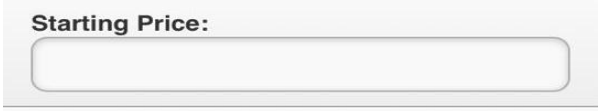


Figure 6.19 – *Evidence of satisfied SYSREQ 6.3*

SYSREQ 6.4 - Users can assign a Postage/Packaging price to an item which is added to the cost of the item when being paid for ✓

[*See SYSREQ 6.1 and Figure 6.20]*

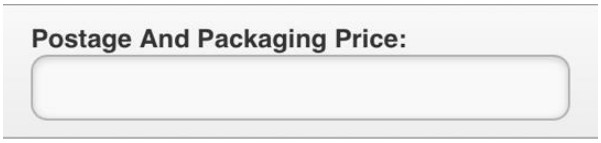


Figure 6.20 – *Evidence of satisfied SYSREQ 6.4*

SYSREQ 6.5 - Users can select the date and time that the bidding for the item ends at ✓

[See SYSREQ 6.1 and Figure 6.21]

Figure 6.21 – *Evidence of satisfied SYSREQ 6.5*

SYSREQ 6.6 – User's auction will expire automatically at specified expiry date and time ✓

[*See SYSREQ 6.1]*

SYSREQ 6.7 – User can view details of their completed auctions ✓

Figure 6.22 shows how on the "My Account" section of the site, there is an area for "My Expired Auctions", which displays all the finished items that user had been selling. When they click on an item on this list it will display the winning bid, the address the product needs sending to and an option to view the buyer of the product.

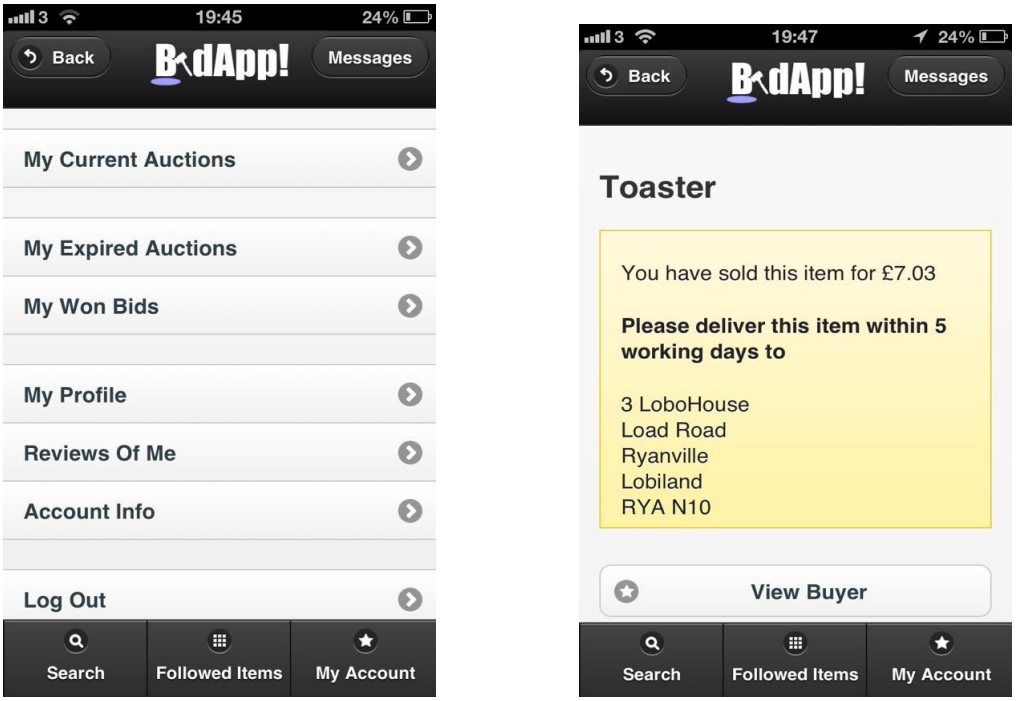


Figure 6.22 – *Evidence of satisfied SYSREQ 6.7*

SYSREQ 6.8 – If their item has been won, the user is provided necessary information to deliver their item ✓

[See SYSREQ 6.7 and Figure 6.23]

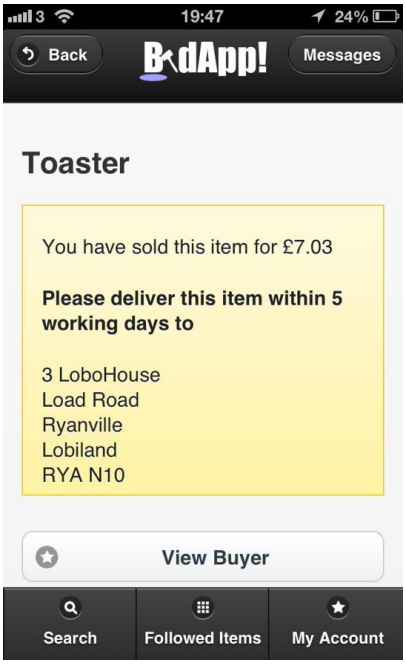


Figure 6.23 – *Evidence of satisfied SYSREQ 6.8*

6.7. SYSREQ 7 - System will implement a feedback system which rates the quality of a user in both buying and selling

SYSREQ 7.1 - Users can give feedback on others once they have purchased from them ✓

Figure 6.24 shows how once the auction has expired, the buyer and seller of an item may provide feedback to one another. They must give a rating out of 5 stars, and can optionally give any comments they feel necessary about their transaction.

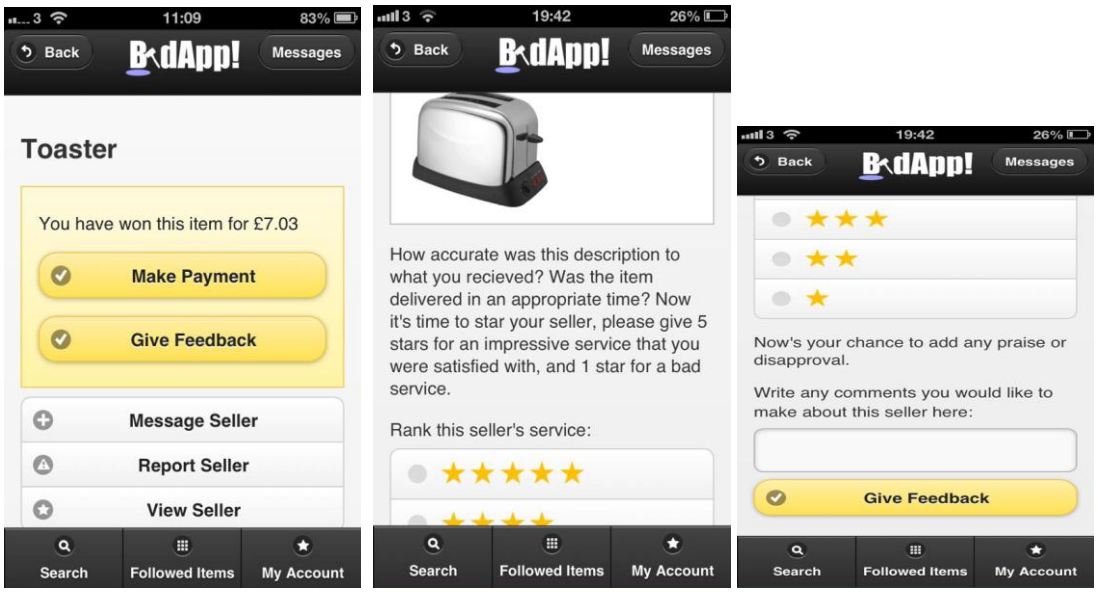


Figure 6.24 – *Evidence of satisfied SYSREQ 7.1*

SYSREQ 7.2 - Users is ranked according to their quality of feedback ✓

Figure 6.25 shows how each user is assigned an average feedback rating which is simply an average of each of their feedback scores. This is represented by a star rating on their profile page.

Figure 6.25 – *Evidence of satisfied SYSREQ 7.2*

SYSREQ 7.3 - Users can view the feedback ranking of all users ✓

Users can see each users star rating on their profile (as described in SYSREQ 7.2), they can also see all the additional comments left by users for each previous transaction.

6.8. SYSREQ 8 - The system should monitor buying and searching patterns of users

SYSREQ 8.1 - The system should monitor what types of items a user most often purchases ✓

The system keeps a track of all items that the user purchases. Using this, it can work out which categories of items they buy the most.

SYSREQ 8.2 - The system should formulate lists of recommended items for the user based on what they buy most often ✓

Figure 6.26 shows how using the functionality described in SYSREQ 8.1 it can work out the categories of their most bought items and can generate recommendations of that category.

Figure 6.26 – *Evidence of satisfied SYSREQ 8.2*

SYSREQ 8.3 - The system should monitor what categories of items a user most often searches for ✓

The system keeps a track of what the user has searched for in the past, and similar to SYSREQ 8.1 this data can be used to determine which category the user most often searches for.

SYSREQ 8.4 - The system should formulate lists of recommended items for the user based on what they search for most often ✓

Figure 6.27 shows how using functionality described in SYSREQ 8.3 it can work out the categories of their most searched-for items and can generate recommendations of that category.

Figure 6.27 – *Evidence of satisfied SYSREQ 8.4*

6.9. SYSREQ 9 - The system is secure in the interest of its users and maintaining integrity

SYSREQ 9.1 – The system will facilitate the encryption of private or confidential data when communicating to other devices ✓

Data stored in the database is encrypted to ensure user confidentiality.

SYSREQ 9.2 - The system will provide security measures to prevent the misuse or hacking of data from within the device ✓

The database is secured to prevent SQL injection attacks etc.

SYSREQ 9.3 - The system will hide information deemed to be unnecessary to display in order to prevent the misuse of confidential information by other users ✓

The system only displays the information of users which need to be seen, namely when providing delivery addresses to sellers of items.

7. Testing

To ensure a quality product, full testing was carried out. These included a variety of tests to encompass all of the facilities and services offered by the application, and guarantee complete user satisfaction.

7.1. JUnit Testing

Firstly, technical tests were carried out using JUnit- this utilized loop invariants and similar to thoroughly test the logic of the program, and catch potential bugs and crashes early. Secondly, functional tests were carried out, which directly related to the original specifications. This tested that the program met the original user needs and demands. For example, we tested components such as “User can login to their personal account”; the result, outcome and any necessary action were then recorded.

The java class source code was tested by using JUnit. Testing cases and testing results are shown in Figure 7.1.

Test class	Test Date	Test successfully?	Successful debugging?
Bids.java	2012-02-03	✓	
Feedback.java	2012-02-03	✓	
FollowItem.java	2012-02-03	✓	
ItemImage.java	2012-02-03	✓	
Listing.java	2012-02-03	✓	
Message.java	2012-02-03	✓	
Report.java	2012-02-03	✓	
Type.java	2012-02-03	✓	
UserSearches.java	2012-02-03	✓	
Userinfo.java	2012-02-03	✓	
BidsBean.java	2012-02-12		✓
FeedbackBean.java	2012-02-12	✓	
FollowItemBean.java	2012-02-12		✓
IOWorkBean.java	2012-02-12		

ItemImageBean.java	2012-02-12	✓	
ListingBean.java	2012-02-12	✓	
LoginBean.java	2012-02-12	✓	
MessageBean.java	2012-03-12	✓	
PageBean.java	2012-03-12	✓	
RegisterBean.java	2012-03-12	✓	
ReportBean.java	2012-03-12		✓
TypeBean.java	2012-03-13	✓	
UserSearchesBean.java	2012-03-13	✓	
UserinfoBean.java	2012-03-14		✓
BidsDAO.java	2012-03-14	✓	
FeedbackDAO.java	2012-03-14		✓
FollowItemDAO.java	2012-03-14		✓
ImageDAO.java	2012-03-14	✓	
Listing_DAO.java	2012-03-17	✓	
MessageDAO.java	2012-02-17		✓
ReportDAO.java	2012-03-17	✓	
TypeDAO.java	2012-03-17	✓	
UserSearchesDAO.java	2012-03-17	✓	
UserinfoDAO.java	2012-03-17		✓

Figure 7.1 –*Results of JUnit Testing*

Finally, user evaluations and reviews were carried out in order to test the functionality of the interface and interactivity of the system. These tested how the users responded, and whether their outcomes could be predicted and whether they were expected or not.

7.2. System Testing Log

Test	Test Description	Expected Result	Actual Result
01	New user can create an account and profile with BidApp. They will become a member, with all of their personal, account and payment details stored.	A new account will be created that stores the users personal, account and payment information, along with preferences, within the MySQL database, for use in the future.	Success
02	Existing users can login to their personal accounts.	Existing users will be able to login to their accounts and see their details, and keep up to date with any outstanding auctions or sales they may have in progress..	Success
03	Existing users can change and edit personal, account and payment details within their profile.	Details such as address can be changed and modified within a user's profile.	Success
04	All users can search for filtered and categorized item listings currently open for bidding.	Both members and non-members can search for items currently on auction on BidApp by searching a categorized database.	Success
05	Users can send and receive messages, and start message threads between other users, to discuss and exchange product and bid information.	Users can directly message other user; the message thread will be available to each user via the Messages page.	Success
06	Users can place bids on selected items of interest.	After browsing items, users can choose a bid value and place a bid on an item. This item will be stored in their 'Current Bids' list.	Success
07	User can follow items of interest.	The item required to be followed will be selected,	Success

		and added to a 'Followed' list for future reference, should the user want to return and check the progress.	
08	Should a user win an auction by placing the highest bid within the specified time, user can specify delivery address and make a payment.	BiddApp will give the user the option to select delivery address and payment method before seller dispatches item.	Success
09	Users can, after interaction with other users, create a feedback report. This feedback will be used within an aggregate function to produce average user ratings, which will be a public entity.	Feedback scores are calculated to represent an average; this number will be available to all users.	Success
10	Users who are unhappy in anyway with the service received or quality of goods purchased can report sellers to admin users.	A message will be send and thread started with an admin user, who will monitor the situation and act appropriately.	Success
11	Admin users can initiate sanctions against users not complying with policy, for example issuing warnings, or suspending accounts for fixed periods or indefinitely.	Warnings will be issued via messaging system, and suspension/deletion of accounts instant.	Success
12	Users can create an item listing for auction. They can specify a starting minimum bid and end date of the auction, and include details of the item for sale.	An auction will be created with the required details, and be available to potential buyers via the search facilities.	Success
13	Sellers can edit auctions, including end date an item details.	Details will be modified, and changes visible to users.	Success
14	Sellers can terminate outstanding auctions, regardless	The auction will no longer be live, and will not appear	Success

	of whether there are current open bids from other users.	throughout the search or other users' followed items listings.	
15	Users can log out of their account after sessions.	Session is suspended and account details not available until logging back in.	Success
16	Recommended items are specific and relevant to users based on preference and past activity.	Items of relevance based on a number of factors are displayed to all registered members.	Success
17	The JSP pages are active and create reliable communication links between the user web interface and backend server.	Any user interaction is processed by the server and responded to appropriately.	Success
18	The MySQL database is updated with new information instantly, such as creation of new accounts will require storing of personal details.	Any updates or new data added (such as new users) will be added to the database instantly.	Success
19	MySQL database executes valid queries and correct aggregate functions, such as user feedback ratings and scores	Correct data such as feedback scores are returned as the result of queries.	Success
20	The remote server hosts the entire application and can be accessed from both mobile and desktop devices.	The full application can be viewed and operated from either mobile or desktop devices.	Success

Figure 7.2 –*System Testing Log*

8. Project Reflections

8.1. Project Management Perspective

Coordinating the group to ensure that the best abilities of each individual could be used to our advantage created many difficult challenges. It was found that multiple members of the group would be familiar with one aspect of the implementation, yet most had no previous experience with other elements such as the JSP code. As a result, the group found success by working through the implementation of the software together more than individually. Using this approach, it was possible to ensure every member of the group could understand the architecture and basic structure of all aspects of the code as opposed to their individual pieces only. With this greater understanding, it was far easier to merge individual implementations as they had been designed with an awareness of other connecting elements of the software, which improved the quality of individual coding even when subjected to new and different coding languages and styles required for the project.

8.2. Problems Encountered

Overall, the group encountered few serious problems, yet were presented with numerous challenges of varying degrees.

The first was to ensure that the implementation was completed by the deadline with all members of the group working in coordination around their respective university schedules. Despite members of the group implementing from home individually, there was a need to ensure that what was implemented would be coordinated between each member. This had led to the organisation of a number of implementation specific group meetings which allowed all members to implement together for four hours a week. As a result, the meetings ensured strong communication between the group and improved individual awareness of the status of the project as a whole. The meetings were also used to discuss code, giving the opportunity for suggested improvements to structure from other members. Moreover, more informed and appropriate implementation decisions could be made as the group was aware of how each other member's code worked.

Furthermore, another challenge faced by the group was the operation of the system's server side on different computers and network configurations such as the university lab computers, where numerous firewalls and security measures needed to be reconfigured to permit the connection of test devices. Initially, it was decided that server implementation would avoid the university connections, but later testing concluded that there was a need to reduce the complex URL needed to access a private computer from a mobile phone. This challenge was overcome using the university's TSG service to gain privileges on the school network when executing the server.

Moreover, the group encountered a challenge towards the end of the implementation with testing a newly implemented feature on a device. The issue came as newer features would make changes to the database, and inaccurate information was as a result added to the database. In order to overcome this issue, update periods for the database were planned, where it was agreed that no members of the group would test features involving the database while changes to the database structure were being made.

One final challenge faced by the team was the testing of the system on touch screen devices. The challenge came when deciding what constituted a touch screen device. It was discovered that numerous smart phones offered partial touch screen, or a far less than typically sized screen. As a result, a cut-off point was needed to be decided as despite the specification to work on all touch screen phones, it was clear that some touch screen devices were not designed to fully facilitate the use of touch enhanced software.

9. References

- Oracle. (2012). *JavaServer Pages Technology.* Available: http://www.oracle.com/technetwork/java/javaee/jsp/index.html. Last accessed 10th December 2012.
- Oracle. (2012). *Java Servlet Technology.* Available: http://www.oracle.com/technetwork/java/index-jsp-135475.html. Last accessed 10th December 2012.
- sql.org. (2012). *Welcome.* Available: http://www.sql.org/. Last accessed 10th December 2012.
- Oracle Corporation. (2009-2011). *JavaTM Platform, Enterprise Edition 6 API Specification.* Available: http://docs.oracle.com/javaee/6/api/. Last accessed 10th December 2012.
- Oracle Corporation. (1993, 2012). *Java™ Platform, Standard Edition 7 API Specification.* Available: http://docs.oracle.com/javase/7/docs/api/index.html. Last accessed 10th December 2012.Oracle Incorporation. (2012). *Netbeans IDE.* Available: http://netbeans.org/. Last accessed 10th December 2012.
- Oracle Incorporation. (2012). *Netbeans IDE.* Available: http://netbeans.org/. Last accessed 10th December 2012.
- Glassfish. (2008-2012). *Glassfish Community.* Available: http://glassfish.java.net/. Last accessed 10th December 2012.
- WiseGeek. (2013). *How Proxy Bidding Works.* Available:
- http://www.wisegeek.com/what-is-proxy-bidding.htm
- Webopedia. (2013). *FQDN.* Available:
- http://www.webopedia.com/TERM/F/FQDN.html
- Microsoft Support. (2013). *Remote Desktop Protocol.* Available:
- http://support.microsoft.com/kb/186607

10. Appendices

10.1 Complete System Requirements Specification

1. Overview

1.1 Purpose

This document specifies requirements for a mobile website that facilitates the auction of items online by its users. The website also aims to assist buyers and sellers by providing additional features that will be of convenience. Such features include a way for users to provide or view the feedback of others using the website so that the reliability of an unknown user may be considered before a purchase. Other features include the ability to easily search for or be updated on items of interest that are on auction.

This document is targeted to a small team of users who will assess the suitability of the software to their needs according to their experience in the auctioning field. This document is also to be viewed by system designers who will assess the feasibility and requirements of implementing the software.

1.2 Definitions, Acronyms and Abbreviations

- Operating System (OS) - Set of computer programs managing hardware and software resources. Examples include Windows, Macintosh OSX and Linux distributions such as Ubuntu [1]
- MySQL - Open source database [2]
- iOS - Operating system developed for Apple mobile systems [3]
- 2G - 2nd generation wireless telephone technology [4]
- 3G - 3rd generation wireless telephone technology [5]
- FAQ - Frequently Asked Question

1.3 References

1. Operating system http://computer.howstuffworks.com/operating-system.html
2. MySQL http://www.mysql.com/why-mysql/
3. http://en.wikipedia.org/wiki/IOS
4. 2G http://cellphones.about.com/od/phoneglossary/g/2g.htm
5. 3G http://voip.about.com/od/mobilevoip/p/3G.htm

2. PRODUCT DESCRIPTION

2.1 Product Concept and Core Services

The purpose of the product is to provide users with a mobile service that aims to improve the process of online auctioning. Included in the product will be web services to promote interaction between users who are buying and selling, and financial services to effectively facilitate the transfer of money after an auction has been won.

The Mobile Auction System is an auction and trading website targeted for mobile devices, which make use of web services that can be accessed by mobile users on all mobile operating systems. The auction is a sale in which a seller presents their products on a public platform to buyers. The selling price in an auction is determined by the bids made by interested buyers. The price they bid is based on their own valuation and buyers' need for the product and the product is sold to the highest bidder at close of auction.

We assume that there are 4 types of users on the system: guests, registered users, administrators and banned users. Guests are limited to basic functionality of the software, whereas registered users are granted privileges intended to facilitate the use of the software by members of the public. Administrators are used to control the public use of the software, and banned users are a side effect of misuse.

Before a user can make use of the full functionality of the site, the user has to register. The user will have to enter some personal information in a form in order to do this. Both guests and users can browse the different auctions, but only registered users can bid on auctions or place their own auctions. The system will enable the user to create a user account and upload a picture to publicly represent their account to other users.

The system will allow the user to place a new item on auction by uploading a picture and description of the item for sale. When a user creates an auction, he or she also has to specify payment methods, transport methods, a minimum price, duration for the auction and other general information about the auction.

The system will allow the users to search through the available products via searching for keywords in the item's name, or via a menu structure which will allow the user to pick a keyword and then select a more specific product through a list of associated words. Users can browse through auctions with categories or tags. The system also provides communication between buyers and sellers so as to exchange information such as prices and contact information to facilitate trading between them. Users can also search for a specific auction.

A user can follow an auction. The followed auctions list is a list of auctions that interest the user but on which he or she may not have bid yet. When the user wants, he or she can bid on the auction. A user can view all the auctions he or she has bid on in the active auctions list. The bidding system itself will support proxy bidding, wherein a user may place a maximum price for an item that they would be willing to pay and the system will proceed to place a bid on the user's behalf. The system will continue to bid on the user's behalf, whenever they are outbid by another user, until their maximum is exceeded or the auction is won. When a user has won an auction, he or she can pay for the item. This will be done through a financial transaction. The buyer can pay for the item by choosing one of the payment options the seller has specified.

When the transaction is done, buyers and sellers can rate the transaction. Each user will have an overall rating which is based on the ratings of their transactions. The user's feedback will then be available on their account, which is publicly viewable. A reputation management system will also be implemented based on the feedback given by users from their experiences of buying and selling using the software. This reputation management system will allow the users to then "filter out" disreputable sellers (i.e. those with low feedback scores) when searching for items to purchase.

The system will also monitor the user's behaviours and patterns for both buying and searching items. The gathered data will be used by the system to describe a user in context, using keywords which will enable the system to recommend items.

Administrators have some extra functionality; they can manage the users and auctions. For example, administrators can also retract a bid of a user, or withdraw them from future use of the software by adding them to a list of banned users.

2.2 Assumptions and Dependencies

The system relies on

- User Hardware: The system will operate on a mobile phone device
- User Operating System: The system must be operating on the any mobile platform
- User Network: The user must have access to the internet via mobile phone connections on 3G, 2G and WiFi networks
- Permanently operating database and server: The system must connect to a database via a server in order to be filled with content

3. USER REQUIREMENTS

User Class #1. Guest

- UREQ 1.1 Browse auctions through searching
- UREQ 1.2 Browse auctions by selecting a category
- UREQ 1.3 Request an account
- UREQ 1.4 Log in
- UREQ 1.5 Get help (FAQ)

User Class #2. Registered users

- UREQ 2.1 Browse auctions through searching
- UREQ 2.2 Browse auctions by selecting a category
- UREQ 2.3 Add an auction to their followed auctions
- UREQ 2.4 View their followed auctions
- UREQ 2.5 Bid on auctions
- UREQ 2.6 View their active auctions
- UREQ 2.7 Make a transaction after winning an auction
- UREQ 2.8 Rate a transaction
- UREQ 2.9 View previous transactions
- UREQ 2.10 Begin a public auction by uploading product information including photos and auction expiry date and time.
- UREQ 2.11 Expire an auction (only for auctioneer)
- UREQ 2.12 View information regarding auctions they are currently hosting
- UREQ 2.13 Access and modify their account information
- UREQ 2.14 Message other users
- UREQ 2.15 View messages from other users
- UREQ 2.16 Report other users

- UREQ 2.17 Logout
- UREQ 2.18 Get help (FAQ)

User Class #3. Administrator

- UREQ 3.1 Browse auctions through searching
- UREQ 3.2 Browse auctions by selecting a category
- UREQ 3.3 Follow an auction
- UREQ 3.4 View their followed auctions
- UREQ 3.5 Message other users
- UREQ 3.6 View messages from other users
- UREQ 3.7 View reports made by registered users
- UREQ 3.8 Retract bids from users
- UREQ 3.9 Ban users accounts
- UREQ 3.10 Delete user accounts
- UREQ 3.11 Access and modify their account information
- UREQ 3.12 Logout
- UREQ 3.13 Get help (FAQ)

4. FUNCTIONAL REQUIREMENTS

4.1 System Requirements

SYSREQ 1 - Users can create their own personal account

- SYSREQ 1.1 - Users can create a username and password
- SYSREQ 1.2 - Users can log in with a username and password
- SYSREQ 1.3 - Users can add their personal details to their account
- SYSREQ 1.4 - Users can add a personal photo to their profile
- SYSREQ 1.5 - Users can add their banking/financial details to their account
- SYSREQ 1.6 - Users can message other users from their account
- SYSREQ 1.7 - Users can view messages received from other users
- SYSREQ 1.8 - Users can change all information uploaded to their account
- SYSREQ 1.9 - Users can logout

SYSREQ 2 - Users can view items on auction

- SYSREQ 2.1 - Users can search for items on auction by category
- SYSREQ 2.2 - Users can search for items on auction by keywords
- SYSREQ 2.3 - Users can search a list of items suggested by the system
- SYSREQ 2.4 - User is given information regarding item on auction when it is selected from the search

SYSREQ 3 - Users can follow items on auction

- SYSREQ 3.1 - Users can add items on auction to their followed items
- SYSREQ 3.2 - Users can view items on auction that they are following

SYSREQ 4 - Users can bid on items

- SYSREQ 4.1 - Users can place a bid on an item, specifying an offer which must be higher than the current highest offer at that time
- SYSREQ 4.2 - Users can proxy bid, whereby they submit a maximum bid, and the system will bid for the user whenever they are outbid until this maximum is reached
- SYSREQ 4.3 - Users can be alerted as to when they have been outbid for an item
- SYSREQ 4.4 - Users are alerted that they have won a bid if they hold the maximum bid after the time of the auction's expiry

SYSREQ 5 - Users can manage their transactions

- SYSREQ 5.1 - Users can buy items they have won in a transaction using the financial details stored with their account
- SYSREQ 5.2 - Users can view details of all previous transactions

SYSREQ 6 - Users can sell items

- SYSREQ 6.1 - Users can place an item on the system which they wish to sell
- SYSREQ 6.2 - Users can add photos of the item, with a maximum of 3 photos per item
- SYSREQ 6.3 - Users can add a starting bid for the item which the bidding will commence from, otherwise the bidding will begin from a default price
- SYSREQ 6.4 - Users can assign a Postage/Packaging price to an item which will be added to the cost of the item when being paid for
- SYSREQ 6.5 - Users can select the date and time that the bidding for the item ends at
- SYSREQ 6.6 – User's auction will expire automatically at specified expiry date and time
- SYSREQ 6.7 – User can view details of their completed auctions
- SYSREQ 6.8 – If their item has been won, the user is provided necessary information to deliver their item

SYSREQ 7 - System will implement a feedback system which rates the quality of a user in both buying and selling

- SYSREQ 7.1 - Users can give feedback on other users once they have purchased from them
- SYSREQ 7.2 - Users will be ranked according to their quality of feedback
- SYSREQ 7.3 - Users can view the feedback ranking of all users

SYSREQ 8 - The system should monitor buying and searching patterns of users

- SYSREQ 8.1 - The system should monitor what types of items a user most often purchases
- SYSREQ 8.2 - The system should formulate lists of recommended items for the user based on what they buy most often
- SYSREQ 8.3 - The system should monitor what types of items a user most often searches for
- SYSREQ 8.4 - The system should formulate lists of recommended items for the user based on what they search for most often

SYSREQ 9 - The system will be secure in the interest of its users and maintaining integrity

- SYSREQ 9.1 – The system will facilitate the encryption of private or confidential data when communicating to other devices
- SYSREQ 9.2 - The system will provide security measures to prevent the misuse or hacking of data from within the device
- SYSREQ 9.3 - The system will hide information deemed to be unnecessary to display in order to prevent the misuse of confidential information by other users

4.2 Hardware Interfaces

- The system requires a server capable of operating permanently unless the service is ever decided to be withdrawn. It is estimated the server must operate on at least a Dual Core 2GHz processor, 1GB of RAM and 100MB HDD for program and database
- The system requires a mobile device using any operating system
- It is estimated the mobile device must have at least a Single Core 1GHz processor, 256MB of RAM and 1MB flash memory for program storage
- The mobile device must be able to connect to an internet browser in order to view the site
- The mobile device must support connections to the internet via traditional mediums such as 2G, 3G and WiFi

4.3 Communications Interfaces

The system will involve the use of confidential data including a user password and financial details. Included in the operation of the system will be the transfer of such private data through mobile networks; an example being 3G. It is therefore necessary that the website will facilitate a secure environment for the storage and transfer of personal or confidential data, and will involve the use of encryption when communications occur. The system will also require communication services such as electronic forms for the input of user data, with the input of some information shown as asterisks where necessary in order to prevent intrusion by other users.

4.4 Software Interfaces

Required to construct the product will be a

- MySQL database
- Java powered client-server networked program with database interaction
- Java libraries for additional support of implementation
- Server facilitating the storage and search of the database
- Client - Mobile device (smartphone, tablet etc) running any operating system

4.5 User Interfaces

The system will make full use of input techniques common to most smartphones and tables. This includes the implementation of a touch screen graphical user interface that will improve usability. Also included in the graphical user interface will be a range of icons and images aimed to facilitate the ease of use and simplicity of the site.

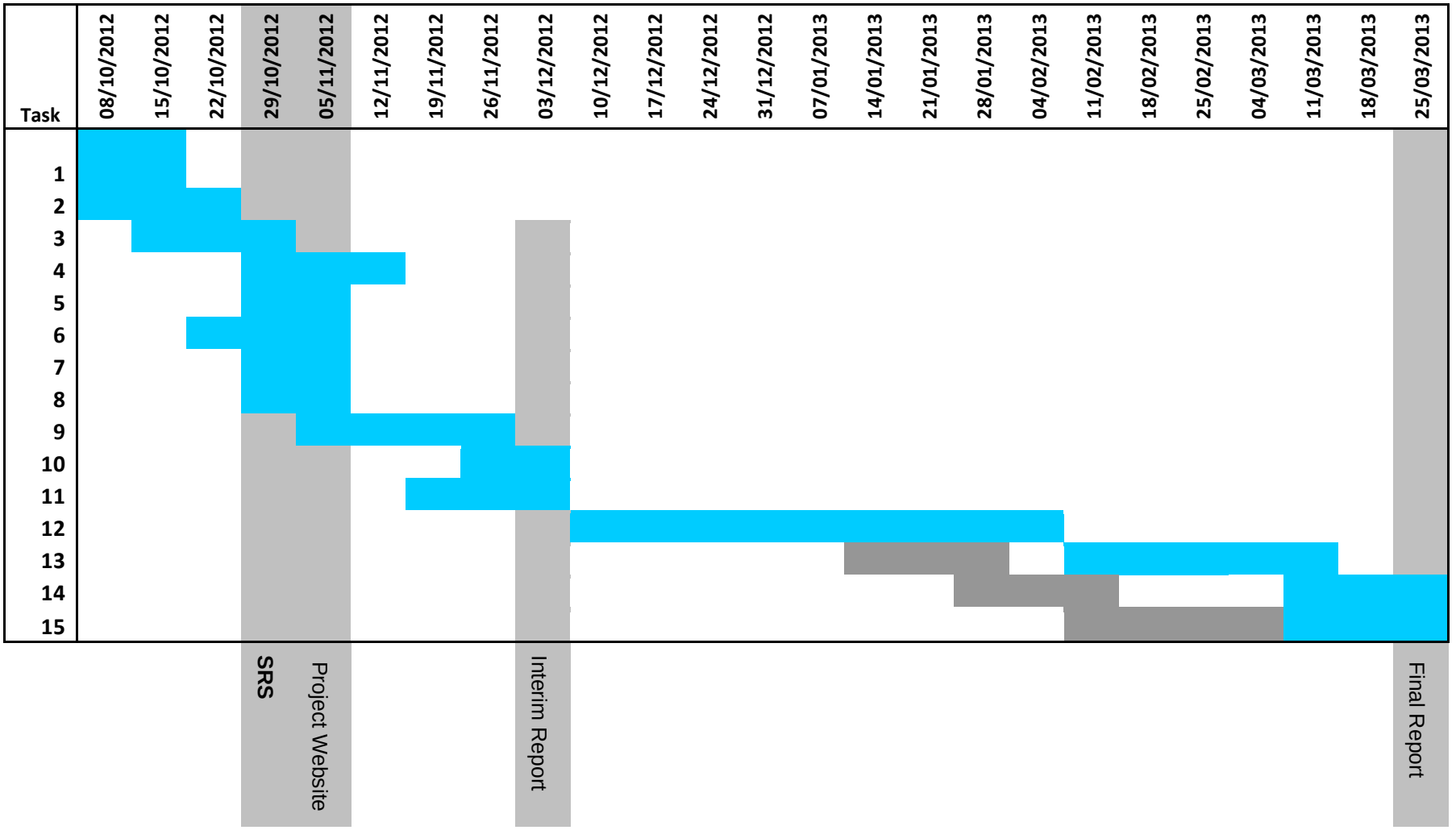
5. NON-FUNCTIONAL REQUIREMENTS

1. The system must be developed in a duration of approximately 6 months.
2. The budget for the development of the software must be small
3. The system must be reliable so that it can run on a 24 hour 7 days a week basis
4. The system must always have a backup copy of data so that it may restart after a potential crash
5. The software must be easy and simple for new users to understand
6. The software must be visually appealing
7. The system must reduce loading times where possible in order to approve usability
8. The software must make use of a variety of input techniques available to mobile devices, with particular attention given to touch screen usage
9. The system must be small in file size in order to reduce data usage of mobile device

10.2. Project Time Plan

The following list is the tasks that are to be completed with dates given for the start and deadline of each task.

Task	Start	Deadline
1 Researching Programming Techniques	09/10/2012	16/10/2012
2 Create Requirements Document	14/10/2012	26/10/2012
3 Create & Update Project Website	19/10/2012	30/10/2012
4 Create Database architecture	30/10/2012	12/11/2012
5 Design User Interfaces	02/11/2012	08/11/2012
6 Vision Scope Document	28/10/2012	05/11/2012
7 Create User stories	02/11/2012	10/11/2012
8 Configure Wiki page	01/11/2012	05/11/2012
9 Create web prototype	08/11/2012	01/12/2012
10 Group Interim Report	28/11/2012	07/12/2012
11 Prototype Development	24/11/2012	07/12/2012
12 Project Code Implementation	10/12/2012	20/01/2013
13 Testing and Refactoring	20/01/2013	01/02/2013
14 Evaluation	01/02/2013	16/02/2013
15 Presentation	16/02/2013	28/03/2013
16 Project Completion		31/05/2013


Task
08/10/2012
15/10/2012
22/10/2012
29/10/2012
05/11/2012
12/11/2012
19/11/2012
26/11/2012
03/12/2012
10/12/2012
17/12/2012
24/12/2012
31/12/2012
07/01/2013
14/01/2013
21/01/2013
28/01/2013
04/02/2013
11/02/2013
18/02/2013
25/02/2013
04/03/2013
11/03/2013
18/03/2013
25/03/2013
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SRS
Project Website
Interim Report
Final Report

10.3. Example Minutes

Minutes from Tuesday 19th February 2013

Chair: Tom Hadkiss

Minutes: Yao Zhang, Tom Hadkiss

Present: Yao Zhang, Tom Hadkiss, Jake Howland, Ryan Lobo, Marike Littlefair, Milena Radenkovic (Supervisor)

- Must identify how trust is defined and calculated in our implementation
- A suggestion to add to the last chapter of the final report in 'future work':
- $R = \sum_{i=1}^{n} o(i) / n$ or $R = \sum_{i=1}^{n} w(i) \times o(i) / n$
- The latter has a weighting so that trusted opinions will have more of an impact on another node's reputation. The overall approach should be that it's very easy to decrease your weighting, but hard to gain it. i.e much harder to loose reputation that to gain it
- **Open day**
- Must be formal
- Must be able to run the product
- Prepare 5/6 sentence summary to introduce product in a clear and attractive manner
- Start the system and explain the scenarios
- Be prepared to answer questions like 'What was challenging? What algorithms?' – they may ask almost anything from the report
- Suggested things to include – leaflets, candy, poster – email Henrik for details on support for some of these
- **Implementation Testing**
- Not all the code must be tested
- More of a focus on bug reporting

POST SUPERVISOR MEETING..

Present: *Yao Zhang, Tom Hadkiss, Jake Howland, Ryan Lobo, Marike Littlefair*

- **Agreed changes to implementation/ plans for future implementation**
- Search – extra filter needed – ie set preferences on ordering of items
- Logged out home page should have info about BidApp
- Logged in home page – scrolling items – should only be only one item on screen at a time to facilitate smaller screens
- My account should have "My current auctions" and "My expired auctions"

ACTIONS

Jake – to be completed by Monday 25th February

Update the database to include a way of telling if the user is banned, and to store the reason why they were banned. Also an update is needed to store if they are an admin, and a way of storing reports too.

Marike – to be completed by Wednesday 27th February

Create formal designs for the user interface to be included in the report

Ryan, Yao – to be completed by Wednesday 27th February

Open an account with TSG/ continue with JSP implementation

Tom – to be completed by Monday 25th February

Complete user interface to report/ give feedback/ make payment. Update the logo using feedback from the group discussion