



DUBLIN INSTITUTE
of TECHNOLOGY

Institiúid Teicneolaíochta Bhaile Átha Cliath

SwapaH.com Goods Swapping Web Application

Final Year Project Report

DT228 BSc in Computer Science

Ken O'Brien
Jane Ferris

School of Computing
Dublin Institute of Technology

22nd March 2012

Abstract

The use of the internet for the buying and selling of goods and services has in recent years increased significantly. Companies such as eBay and Amazon have thrived. The idea of customer-to-customer transactions is being used more and more, but there still seems to be something missing.

The objective of this project is to design and implement a web application that allows users to trade or swap unwanted items they may own with each other through the means of an online auction or through a straight swap in a fun new swapping game. This website is designed to give the user a more fun and interesting way of getting rid of unwanted items. These unwanted items could be of value to someone else. Instead of the usual auction site where a bid of money is made for the item up for auction, users will bid with other items they own. The seller can accept or reject bids, depending on what they like. By the end of the auction, two users will end up getting rid of something they don't want in exchange for something they do!

The game section of the site is based on the true story of a man who started off with a red paper clip, he swapped that paperclip for something a little better, he kept doing this until he eventually got a house. The site will give users a fun way swapping their items with other users. The users will start off with one of their items up for swap. They can swap that item with another users item. They can then swap their new item for another and so on. When the game ends whatever item you have you get to keep.

The site will be coded using html, PHP, and SQL.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

April 2011

Acknowledgements

I would like to take this opportunity to give a special thanks to my supervisor Ken O'Brien who has given great advice and has helped me keep up to date on this project, even under tough time constraints. Also I would like to thank Susan McKeever who has also given great advice and held a helpful lecture every Monday. Finally I would like to thank the Usability Testers who tried the system and filled out the questionnaire.

Table of Contents

Abstract.....	2
Declaration	3
Acknowledgements	4
Chapter 1 Introduction.....	13
1.1 Introduction	13
1.2 Project Objectives.....	13
1.3 Project Challenges	14
1.4 Structure of this paper	14
Chapter 2 Research.....	16
2.1 introduction	16
2.2 A Website and its Architecture.....	16
2.2.1 What is a Website?.....	16
2.2.2 How does a Website work?	17
2.2.3 The different Technologies involved	17
2.3 Auctions.....	18
2.3.1 What is an Auction	18
2.4 MAC OS X	19
2.4.3 Alternatives to Max OS X	20
2.5 Apache Web Server.....	20
2.5.2 Why I chose Apache web server.....	20
2.5.3 Alternative to Apache Web Server	20
2.6 MySQL Database Management System.....	20
2.6.2 Why choose MySQL	21
2.7 PHP	21
2.8 Android Mobile Devices.....	22
2.8.1 Introduction.....	22
2.8.2 Features of Android	22
2.9 MAMP	23
Chapter 3 Design	24
3.1 Introduction	24
3.2 Design Methodology - Agile Unified Process	24
3.2.2 Why choose Agile Unified Process	26
3.3 System Architecture	27
3.3.2 The Client Layer.....	27
3.3.3 The Application Layer	27
3.3.4 The Database Layer	27
3.3.5 Three tier Architecture Diagram.....	28
3.4 System Functionality	28
3.4.1 Login system	28
3.4.2 Inventory	29
3.4.3 Auctions.....	29
3.4.5 Bids	29
3.4.6 Games	30
3.4.7 Chat	30
3.4.8 Android Application	30
3.5 Initial Use Case diagrams.....	31
3.5.1 Initial Use case diagram for SwappaH.com containing basic functionality	31

3.5.2 Initial Use case diagram for SwappaH Android App containing basic functionality	31
3.6 Initial Database Design	32
3.6.1 Introduction	32
3.6.2 ERD Diagram	32
3.7 User Interface	33
3.7.1 Introduction	33
3.7.2 Low fidelity “paper Prototype”	33
3.7.2 Overview	35
3.7.3 Medium fidelity Prototyping	35
3.8 Other Design Considerations	38
3.8.1 Image storage File system Vs BLOB (Binary Large Object)	38
3.8.2 Use of YouTube	38
3.9 Use case Diagrams	39
3.9.1 SwappaH.com Web Application Use Case Diagram	39
3.9.2 SwapaH Android Application Use Case Diagram	40
3.10 Database Entity Relationship Diagrams (ERD).....	40
3.10.1 ERD Diagram without relations for easier viewing of table	40
3.10.2 Final ERD Diagram including relations.....	41
Chapter 4 Implementation.....	42
4.1 Introduction	42
4.2 Setting up Environment - Local MAMP Server.....	42
4.3 Implementing section by section (using basic functionality)	43
4.3.1 Overview	43
4.3.2 Registration system.....	43
4.3.3 Simple login system	44
4.3.4 More complicated login system	45
4.3.5 What is a php Session?.....	46
4.3.6 Register/Login systems Finished User interface	46
4.4 Implementation of Inventory.....	47
4.4.1 Overview	47
4.4.2 What is an Item	47
4.4.3 Creating a new item.....	47
4.4.4 image storage	48
4.4.5 Viewing particular Item.....	48
4.4.6 Editing particular item.....	49
4.4.7 Use of Fancy Box	49
4.4.8Viewing Inventory	50
4.5 Implementation of Auctions	51
4.5.1 Overview	51
4.5.2 Creating a new Auction.....	51
4.5.3 Auction page	52
4.5.3.1 Countdown Timer	52
4.5.3.2 Auction questions	52
4.5.3.3 Bids.....	53
4.5.3.4 Notification System.....	53
4.5.3.5 Accepting/rejecting offers.....	54
4.5.3.6 Your auctions/ index /search and category auctions pages	54
4.5.3.7 Search results and category search	54
4.6 Implementation of Game Section	55
4.6.1 What is meant by Game.....	55
4.6.2 Problem with Virtual Ownership.....	55

4.6.3 Resolution to problem	55
4.6.4 Joining a game	56
4.6.5 Viewing Your current games.....	56
4.6.6 Viewing a particular game	57
4.6.7 How a swap occurs.....	58
4.6.7.1 Offering a swap.....	58
4.6.7.2 Viewing swap offers.....	58
4.6.7.3 SwapChat	58
4.6.7.4 Swap status	59
4.6.8 Number of swaps	59
4.7 Implementation of “Swaps” section	59
4.7.1 Overview.....	60
4.7.2 Tab Layout	60
4.7.3 Finished Auctions.....	60
4.7.4 Top Bids.....	61
4.7.5 Auctions Won	62
4.7.6 Finished Games	62
4.8 Five star review/account System.....	62
4.8.1 Overview	62
4.8.2 How to create a review	62
4.8.3 Accounts	64
4.9 Implementing GeoCoding	64
4.9.1 What is Geocoding.....	64
4.9.2 Uses of geocoding in this project	64
4.10 Android App	66
4.10.1 Overview	66
4.10.2 The Login Activity.....	66
4.10.3 Image scroll view.....	67
4.11 Deployment	68
4.11.1 Hostgator.com.....	68
4.11.2 FileZilla	68
4.11.3 Exporting/Importing Database	68
Chapter 5 Testing.....	69
5.1 Introduction	69
5.2 Functional testing	69
5.2.1 Links.....	69
5.2.2 Forms	69
5.3 Cookies	71
5.4 Database testing	71
5.5 Usability testing	71
5.5.1 Survey that appeared after the Instructions	72
5.6 Compatibility testing	73
5.6.1 Browser compatibility.....	73
5.6.2 Operating System compatibility	74
5.6.3 Mobile platform compatibility	74
5.7 Security Testing.....	74
5.8 Demonstration	75
Chapter 6 Project Analysis	77
5.1 Introduction	77
5.2 Project Plan.....	77
5.2.1 Original Project Plan	77

5.2.2 Analysis	79
5.3 Tech Evaluation	79
5.3.1 PHP.....	79
5.3.2 Apache Web server	80
5.3.3 MySQL.....	80
5.3.4 TextWrangler.....	80
Chapter 7 Conclusion	81
7.1 Learning Obtained	81
7.1.1 New Programming Languages and technologies learned	81
7.1.2 Time Management.....	81
7.1.3 UI design	81
7.1.4 Website architectures	81
7.2 Key Strengths.....	81
7.2.1 Very unique idea	81
7.2.2 User interface	82
7.3 Key Weaknesses	82
7.3.1 Code layout	82
7.3.2 Constant database connections	82
7.3.3 bad security.....	82
7.4 Future Work	82
References	84

Table of Tables

Table 1: Website that make use of PHP.....15

Table 2: Questions in the Survey.....71

Table of Figures

Figure 1: How server-side scripting is incorporated into websites architecture.....	19
Figure 2: Auction on eBay	20
Figure 3: Mac OSX on the MacBook.....	20
Figure 4 Android Device.....	23
Figure 5: Agile Unified Process (AUP)	27
Figure 6 Three Tier Architecture	29
Figure 7 Use Case Diagram of the Basic functionality for web application	32
Figure 8 Use Case Diagram of the Basic functionality for the Android Application..	32
Figure 9: Database ERD	33
Figure 10 low fidelity index page	34
Figure 11 Low fidelity Auction Page.....	34
Figure 12 low fidelity Top Menu	35
Figure 13 low fidelity Inventory.....	35
Figure 14 low fidelity create new auction page.....	35
Figure 15 low fidelity Your Current Auctions page.....	35
Figure 16 Low Fidelity with colour scheme	36
Figure 17 Medium Fidelity Inventory Page.....	37
Figure 18 Medium Fidelity Create New Auction Page.....	37
Figure 19 Medium Fidelity Choosing item to Put up for auction.....	37
Figure 20 Medium Fidelity View Auction Page.....	37
Figure 21 Medium Fidelity Users Current Auctions Page	38
Figure 22 Medium Fidelity Users Swap Section Tab Page.....	38
Figure 23 Medium Fidelity Join SwappaH game page.....	38
Figure 24 Mediem Fidelity View Game Page.....	38
Figure 25 SwapaH Web Application Use Case Diagram	40
Figure 26 Swapah Android app Use Case Diagram	41
Figure 27 ERD Diagram without relations for easier viewing of table.....	41

Figure 28 ERD Diagram Inc Relations	42
Figure 29 MAMP	43
Figure 30 Registration Form	45
Figure 31 logged out header bar	47
Figure 32 logged in header bar.....	47
Figure 33 Add a new item Form.....	47
Figure 34 View Item Page.....	49
Figure 35 Edit ItemPage.....	50
Figure 36 Using Fancybox to display Youtube Video	50
Figure 37 Individual Item in Inventory.....	51
Figure 38 Users Inventory.....	51
Figure 39 Create New Auction Form.....	52
Figure 40 Pop up inventory for item choice.....	52
Figure 41 Chosen item appear in item area.....	52
Figure 42 Countdown timer.....	53
Figure 43 Auction page	53
Figure 44 Close up of Question section in Auction Page.....	54
Figure 45 The bid/ Notification process.....	54
Figure 46 Individual Auction.....	55
Figure 47 Latest Auctions Index page.....	55
Figure 48 Join game.....	56
Figure 49 A game View in 'Your games' section.....	56
Figure 50 Individual Game Page.....	57
Figure 51 Game Details.....	57
Figure 52 View Swap Offers Button.....	59
Figure 53 Swap Offer.....	59
Figure 54 Swap Chat View.....	59
Figure 55 Swap Status.....	60

Figure 56 User Swaps in certain Game.....	60
Figure 57 Game Swap Cycle.....	60
Figure 58 Tabs inside SWAP section	61
Figure 59 Finished Auction with winning bid.....	62
Figure 60 Finished Auction With no Winning Bid.....	62
Figure 61 Current top bids section.....	62
Figure 62 Leave Review Button.....	62
Figure 63 Five star review system.....	63
Figure 64 Written Review page.....	63
Figure 65 Average Ratings in star form.....	64
Figure 66 Example of Written Review with individual rating.....	64
Figure 67 Distance Away View.....	65
Figure 68 How Android app communicates with Database	66
Figure 69 Android App uploading image.....	67

Chapter 1 Introduction

1.1 Introduction

These days people are much more comfortable using the internet, all ages from the young to the elderly are now using the internet on a regular basis. The Internet has many uses, from information to communication, from ecommerce to online banking and many more. The increased performance of security on the Internet has allowed companies and user to safely and comfortably use to Internet for commercial use, ecommerce in constantly increasing in size. This gives way to develop new innovative ways to allow users and business's to buy and sell goods.

1.2 Project Objectives

The objective of this project is to design and implement a web application that allows users to trade or swap unwanted items they may own with each other through the means of an online auction or through a new swapping game concept.

A secondary objective of this project is to design and create an Android mobile application that allows users to view items within their inventory, and add multiple photo's for each item.

There are four objectives within the scope of this project, they include:

- Research into the different aspects involved in creating a web application which include operating system, Server, Database management system, and coding language.
- Research into existing auction websites and possible swapping websites
- Development and design of database and functionalities
- Research and development of a good and friendly user-interface to would cater for a person with any amount of internet experience

1.3 Project Challenges

This project like any other project like it has a number of challenges associated with it. These challenges must be addressed.

The challenges facing this project include

- As there will be a large use of server side scripting languages, getting to grips and having a good understanding and knowledge of the language will be necessary to create a successful website.
- Successfully designing and developing a database that caters for all the data and functionality the website uses
- The implementation of an ease of use user interface that will be designed to be used for all users technologic knowledge background.
- Researching and implementing the connection between an Android Phone and the created database.

1.4 Structure of this paper

The structure of this paper is as follows.

- **Chapter 2: Technologies Researched**
This chapter discusses the technologies and platforms researched for the purpose of this project. To develop a fully functional website with a lot of data content will require the research of a number of technologies, including operating system choice, servers, database's and programming language choice. This chapter will give a decryption of each of these technologies and the reason why I chose them.
- **Chapter 3: Design**
This chapter discusses the design methodology chosen for this project, It then goes on to describe the basic functionality of the site and the basic database design. It then goes on to describe in detail the design of the user interface and then the rest of the functionality that was gathered from the UI design.
- **Chapter 4: Implementation**
This project discusses the implementation of the project. It starts by describing how the development environment was set up. It then goes on to describe each iteration of implementation in detail.

- **Chapter 5: Testing**

This chapter discusses the system validation. How the implemented project was tested. It then goes on to show links to demonstration of the project forming different tasks.

- **Chapter 6: Project analysis**

This chapter discusses the analysis of the project. It starts by discussing the original project plan and how it differed from the plan that was eventually used. After that it discusses the analysis of the technologies used.

- **Chapter 7 Conclusion**

This chapter discusses the overall conclusion to the project. Starting off with the learning obtained from the project, the key strengths and the key weaknesses of the project and finally any future work.

Chapter 2 Research

2.1 introduction

This chapter gives a detailed description of the background of this project, from website architecture to general auction site layout. Along with the technologies considered for use in this project. After reviewing each technology a choice will be made and the reason behind choosing certain technologies, along with the benefits and drawbacks of developing on and developing for these certain technologies. For this type of project, a website, there are numerous choices in technologies in each aspect of the system.

The technologies employed for this project include:

- MAC OS X Operating system
- Apache server 2.2.21
- MySQL Database Management system
- PHP, Server side scripting Language
- Android device platform for image upload extension:

The structure of this chapter is as follows:

- Section 2.2 - Websites and their Architecture
- Section 2.3 – Auction sites
- Section 2.4 – MAC OS X
- Section 2.5 – will focus on Apache server, which is the core of the website.
- Section 2.6 – will focus on MySQL, which is the Database Management System(DBMS) used.
- Section 2.7 – will focus on the PHP server side Scripting languages incorporated with HTML, which is crucial for the functionality of this project.
- Section 2.8 – will focus on the Android mobile device platform that the item image upload application will run on.
- Section 2.9 – MAMP

2.2 A Website and its Architecture

2.2.1 What is a Website?

A Website is a Collection of Web pages than can be viewed on the World Wide Web. These WebPages are documents that can be views on a web browser and contain mainly digital media, from images to videos to text. These WebPages are grouped together on a server machine. These pages that are grouped and connected together are the website.

2.2.2 How does a Website work?

Generally, All the web pages of a website are contained in a server. A browser anywhere in the world, presuming it has permission, can then access these pages. The browser will send a page request to the server. The server will retrieve the page and perform and server side scripting if needed. Also a server will have a database associated with it. For example if a user has logged in to a restricted website usually there will be a server side script that queries the database to verify the user exists and has the appropriate permission. The main elements in the architecture of a website are, a server containing all the web pages and digital media of the website, A database containing all of the data that builds up the website that the server queries. A users browser that sends request to the server.

During extensive research a number of technologies have been evaluated for use in this

2.2.3 The different Technologies involved

Web Server: A web server are pieces of hardware and software that deliver web pages on request to clients be it a browser or mobile device. The machine itself needs to have a constant internet connection, be fast, have a good deal of storage space and plenty of RAM. There are a number of server applications available such as Apache Web Server and IIS, these are explained future in the chapter.

Database Management System: A database management(1) system is a piece of software that allows user to create and maintain a database. A database is a collection of related data stored in a standardized format. The four main elements of a database are

- Data
- Relationships
- Constraints
- Relationships

The database an associated database management system are collectively known as the database system. The database management system itself allows a user to

- Specify the data types, structures and constraints for the data
- Store the data
- Manipulate a database by performing functions such as querying the database, updating the database, and generating reports from the data stored in the database.\

Server side scripting: While client-side scripting is the nice, eye-catching part of web development, server-side scripting(2) is invisible to the user. Server-side web scripting is mostly about connecting web sites to back end servers such as database's

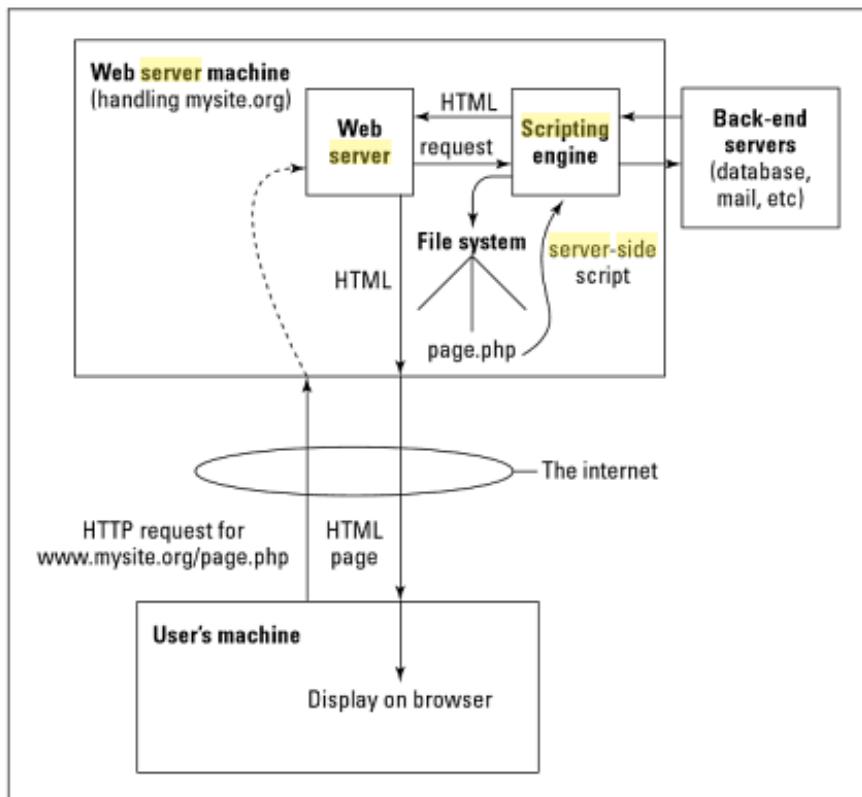


Figure 7: How server-side scripting is incorporated into websites architecture

2.3 Auctions

2.3.1 What is an Auction

An auction is the process of buying and selling good using competing bids. If an item is put for auction it means people can offer bids of money for that item. The bidders compete with each other offering higher bids until eventually there is 1 bidder left with an offer no one wants to match or better. That bidder is then the winner. The winner will pay the owner of the item the amount of that top bid. If no one bids for the item then the item is not sold and still the property of the seller who put it up for auction. Sometimes the seller will put a reserve price on the item. What that means is if there is a winner of an auction but the amount offered is not above the reserve price then the seller can refuse to sell the item to the winning bidder.

2.3.2 What is an Auction site?

An auction site is a website that allows registered users to put a physical item up for a virtual auction, were other registered users can compete to buy item by bidding money for it. For example eBayⁱ. eBay allows users to put their items up for auction, other users can bid money for the item although no actual money is dealt with until the auction has finished. These auctions have a time limit and the Bidders compete until eventually the auction time has ended. The user with the top bid wins the auction and now has the opportunity to buy said item from the original auctioneer user. Payment to the seller is usually done by PayPalⁱⁱ. Once the seller receives the payment they can then go on to post the item to the winner. Sellers pay a listing fee to put an item up for auction. eBay charge different amounts for length of auction, pictures, and content volume.

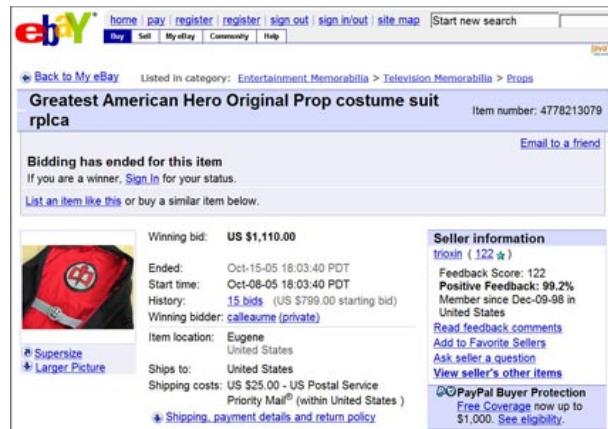


Figure 8: Auction on ebay

2.4 MAC OS X

2.4.1 What is MAC OS X

Mac OS X is a UNIX based operating system that runs on apples computers such as MacBook, MacBook pro. OS X.6 "snow Leopard" was released in 2009. In every MAC OS X computer there is a power Apache Web server built into it.(3)



Figure 9: Mac OS X on the MacBook

2.4.2 Why I chose Mac OS X

There are a few reasons behind choosing Mac OS X to both develop the sight and run the server/database on. One reason as I mentioned before is that Mac OS X has within it a powerful built in Apache web server. This shows that the mac and apache are very compatible and work together with optimum performance. Another reason behind choosing mac os x is my person experience, I have been using mac for over a year now and I have found it to be a far superior operating system to others that I have used, mainly Windows and a small bit of Linux Ubuntu.(3)

ⁱ <http://www.ebay.ie/>

ⁱⁱ <https://www.paypal.com/ie>

2.4.3 Alternatives to Max OS X

Windows Server: is a product of Microsoft. "Windows Server 2008 R2 is a multipurpose server designed to increase the reliability and flexibility of your server or private cloud infrastructure".

2.5 Apache Web Server

2.5.1 What is Apache Web Server

Often referred to as simply *Apache*, a public-domain open source Web server developed by a loosely knit group of programmers. The first version of Apache(4), based on the NCSA httpd Web server, was developed in 1995. Core development of the Apache Web server is performed by a group of about 20 volunteer programmers, called the Apache Group. However, because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons. In many respects, development of Apache is similar to development of the Linux operating system.

2.5.2 Why I chose Apache web server

Mac OS X 10.5 and greater have a powerful Apache web server built into the operating system, Apache is the most widely used web server in the world, because of this I feel that it is the most stable and safe option. Because of the sheer volume of use, most bugs are found and fixed quickly. An estimated 54% of all web sites run Apache. Of course, this does not count any sites that are behind corporate firewalls, and there is no way to measure those numbers, but Apache is clearly the favorite web server in use today

2.5.3 Alternative to Apache Web Server

Internet Information Services: IIS(5) for Windows Server is a flexible, secure and easy to manage Web server for hosting anything on the web. From media streaming to web application hosting, IIS's scalable and open architecture is ready to handle the most demanding tasks.

2.6 MySQL Database Management System

2.6.1 What is a database management system

MySQL is a relational database system(6), it can provide multi-user access to a number of database's. MySQL is a popular choice of database for use in web applications. An example of some popular websites that make use of my SQL include YouTube, Facebook and twitter

2.6.2 Why choose MySQL

MySQL has a great reputation as mentioned above is currently being used by some of the biggest and most popular websites around today. There is already a vast amount of MySQL users out there, which shows that it is a reliable database system. MySQL has some fundamental features to it, these include: (7)

- MySQL is a database server, in other words it is used in client server systems where the bulk of the processing and storage takes place on the server and the client is the personal computer or mobile device.
- MySQL performs multithreading processing, this means that it allows multiple clients to connect to it and run queries simultaneously. This makes it extremely fast and well suited to client-server environments such as web sites and other environments that process numerous transactions for multiple users.
- MySQL features a user permissions system, with which it can control users access to any number of its databases. This system is so sophisticated that few competing relational database systems can match its level of security

2.7 PHP

2.7.1 What is PHP?

PHP(8) is a scripting language, mostly used on the server side, it can be used to generate Hypertext Markup Language (HTML) information dynamically. PHP connects to a web server, usually this is Apache web server or Internet Information Server (IIS). Once it has finished generating proper HTML, it sends its creation back to the web server for delivery to the requesting client. It can be used in other areas such as command line, desktop pc. But it is most commonly used in the web server environment.

2.7.2 Why choose PHP?

- PHP can be integrated with one of many database management tools such as MySQL, SQLight and Oracle. This makes their content as dynamic as possible. In reality what is being produces is a static html file but it is being created in the fly so its seems to be dynamic.
- PHP is easy to understand an learn, anyone with a background in programming languages such as C or C++ can pick it up with ease and will feel comfortable using and understanding PHP. PHP runs on almost every platform including UNIX, Macs and Windows versions.
- Some of the major and most popular websites in the world are powered at some level by PHP, Table 1 includes a short list of popular websites that make use of PHP, it includes their URL and a very brief description of what each site does.

Website Name	Website Description	URL
Facebook	Social Networking	http://www.facebook.com/
Flickr	Photograph sharing	http://www.flickr.com
Wikipedia	Online Collaboration Encyclopedia	http://www.wikipedia.org
YouTube	Video Sharing	http://www.youtube.com
Yahoo	Search Engine	http://www.yahoo.com

Table 1 Website that make use of PHP

2.8 Android Mobile Devices

2.8.1 Introduction

Android is a liuxed based operating system that is located on mobile phone devices. It is owned and run by google. "Android powers millions of phones, tablets, and other devices and brings the power of Google and the web into your hands."



Figure 10 Android Mobile Device

2.8.2 Features of Android

Number of devices: The sheer number of devices in the world at the moment that uses the android platform is enormous with over 300 million Android devices in use by February 2012.

OS versions: The android platform is available in many different versions most phones use from 2.2 all the way up to 4.0.

Different screen sizes: as android can be on a variety of phones, the screen sizes will most likely be different. Therefor any app should look and perform the same on every phone.

Android market: Android uses Google Play to sell/distribute its apps to the public

2.9 MAMP

The abbreviation MAMPⁱⁱⁱ stands for Macintosh, Apache, MySQL and PHP. Each of the technologies researched and discussed in this chapter are can be grouped together in this bundle of software called MAMP. MAMP is used to run dynamic web sites on servers running Mac OS X. All of these software's can be download together from the MAMP website and installed together as a bundle.

ⁱⁱⁱ <http://www.mamp.info/en/index.html>

Chapter 3 Design

3.1 Introduction

This Chapter will discuss the choice of design methodology. Why It was chosen for this methodology, sum advantages and disadvantages behind the choice as well. It will also discuss the general functionality of this website. It will discuss the design of the user interface and how this design enabled the identification of every bit of functionality within the web application. The structure of the following chapter can be seen below.

- **Section 3.2** will discuss the Design methodology chosen for the project, the reason behind the choice.
- **Section 3.3** will focus on the systems Architecture
- **Section 3.4** will focus on the Functionality of the Project
- **Section 3.5/3.6** will show the initial use case diagrams/ER diagrams
- **Section 3.7** will focus on the User interface
- **Section 3.8** will focus on other design considerations
- **Section 3.9/3.10** will focus on the final Use case's/final Er diagrams

3.2 Design Methodology - Agile Unified Process

3.2.1 What is the Agile Unified Process (AUP)?

Agile Unified Process(9) is a simplified version of the Rational Unified Process(RUP). It uses a simple and easy to understand approach to development while still remaining true to the Rational Unified Process. Agile Unified Process is serial in the large and this serial nature is captured in four phases. It's also Iterative in the small, in that its disciplines are performed in an iterative manner. There are seven of these disciplines.

The agile unified process goes through 4 serial phases and uses 7 iterative disciplines. These are all shown and explained below:

- **Inception:** The goal of inception is to identify the initial scope of the project, which means identifying the work that needs to be accomplished to deliver the specified project. Another goal of inception is to identify a potential architecture for the system.

- **Elaboration:** The goal of elaboration is to prove the architecture of the system.
- **Construction:** The goal of the construction phase is to build working software on a regular and incremental basis that would satisfy the highest-priority requirements of the system.
- **Transition:** The goal of the transition phase is to validate and deploy the system into a real product environment.

The disciplines define the activities that are in the focus of development, they are used to build, validate and deliver working software. This can be seen in Figure 6. The different disciplines are as follows:

- **Model:** The goal of this discipline is to understand the problem domain of the project, what that means is to understand the areas of expertise that are relevant to the project. Also to identify a viable solution to address this problem domain by drawing up different models such as use-cases.
- **Implementation:** The goal of implementation is to take these previously created models and transform them into executable code. The code will be put through a basic level of testing.
- **Test:** The goal of the test discipline is to ensure the quality of the system by putting it through objective evaluation. This includes identifying defects, checking to see if the system indeed works as it was designed and supposed to, and verifying that all the requirements of the system are met.
- **Deployment:** The goal of the deployment discipline is to plan for the eventual release of the system to the end-user.
- **Configuration Management:** The goal of this discipline is to manage the updated versions of the system over time, but also manage the actual changes that are made to the system.
- **Project Management:** The goals of this discipline is to direct all the activities of the project, this includes managing risks, tracking progress and making sure the development of the system is on time. It also includes coordinating with people and systems outside the scope of the project to be sure it is delivered on time.
- **Environment:** The goal of this discipline is to ensure the proper process, guidance, and tools such as hardware and software are available.

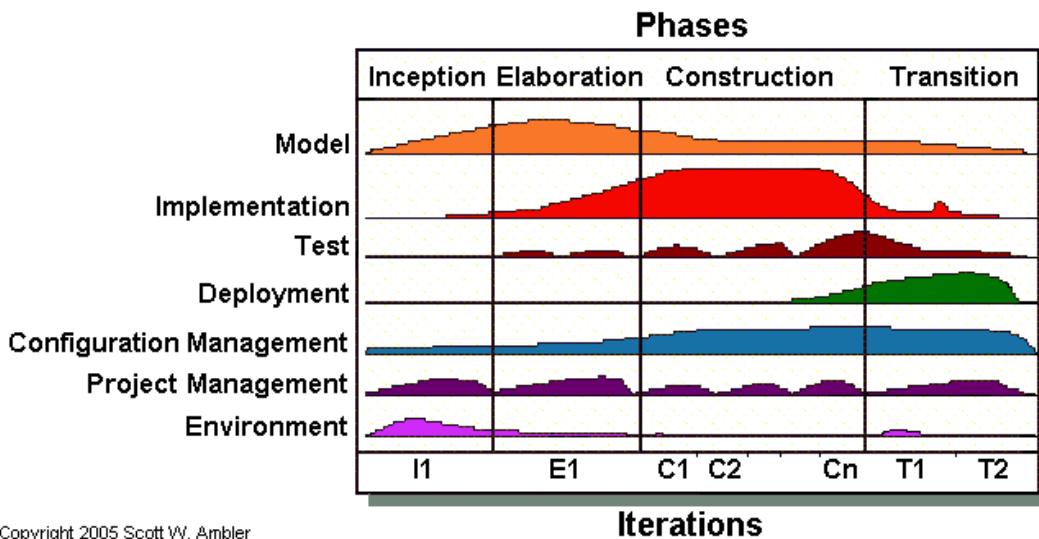


Figure 11: Agile Unified Process (AUP)

3.2.2 Why choose Agile Unified Process

There are a number of reasons for choosing the agile unified process, these include:

Simplicity: The Agile unified process has a philosophy of being simple and easy, everything about it is described in just a handful of pages, not thousand of them.

Rational Unified Process: AUP is basically a simplified version of IBM's Rational Unified process. RUP is a renowned process, AUP is simple and easy. It uses agile techniques while still remaining true to RUP. It uses agile techniques like:

- Agile Model Driven Development: This involves creating models before actually writing any code.
- Test Driven Development: This involves drawing up tests before writing code to perform the test on.
- Agile Change Management: This is a streamlined, flexible approach to requirements change management. Requirements can change frequently, the best way to develop high quality software is to implement and develop the highest priority requirements first.
- Database refactoring: this involves improving the design of the database while not changing the behavior of the database or its data holding ability.

Experience: Having some experience with Rational Unified Process and some of its elements means AUP will not seem to alien.

3.3 System Architecture

This section describes in detail the architecture of the website along with how the website will be designed. It will state the design assumptions along with the design contingencies to reduce the risk of one or more of the assumptions failing. After reading this chapter the reader will have gained knowledge of the different elements of the architecture along with testing and the testing methods applied to the project. They will also learn about the implementation strategy applied.

3.3.1 Proposed Architecture Three tier architecture

This projects architecture will be based on the three-tier architecture model(10). As what the name suggests, this projects architecture will consist of three main sections, the client tier, the middleware or the application tier and the database tier. These tiers communicate with each other and send and receive data from one another. All of these tiers make up the web application, from its user interface to its functionality to storing all the data.

3.3.2 The Client Layer

This is the top level of the application, the one that interacts with the user. In the case of this project this top-level client tier will be the web browser the user is using. This browser would only communicate with the middle tier. It would receive input from the user, it would then go on to send page requests to the middle tier in the form off a http page request. It would then receive the resulting html code back from the web server. All you can see from this tier is the basic html code or css style sheets. If you were to view the source code of a page on a browser you would not see any of the database queries or server side scripting.

3.3.3 The Application Layer

This is the layer that would handle all of the functionality of the website, it controls the execution of all the server side scripts and algorithms. The client layer sends page requests along with some values to the application layer. The application layer receives this request, finds the page then performs and executes the appropriate scripts. It then sends back the page to the client layer (browser).This layer controls all of the business logic, it communicates to the database layer. Sending SQL commands to the database management system and receiving data back.

3.3.4 The Database Layer

This layer is made up by the database servers. It is the layer that stores all of the data associated with the application. The application layer sends SQL commands to the database layer. The database management system executes these commands and retrieves data if required. Sending the retrieved data back to the application layer.

3.3.5 Three tier Architecture Diagram

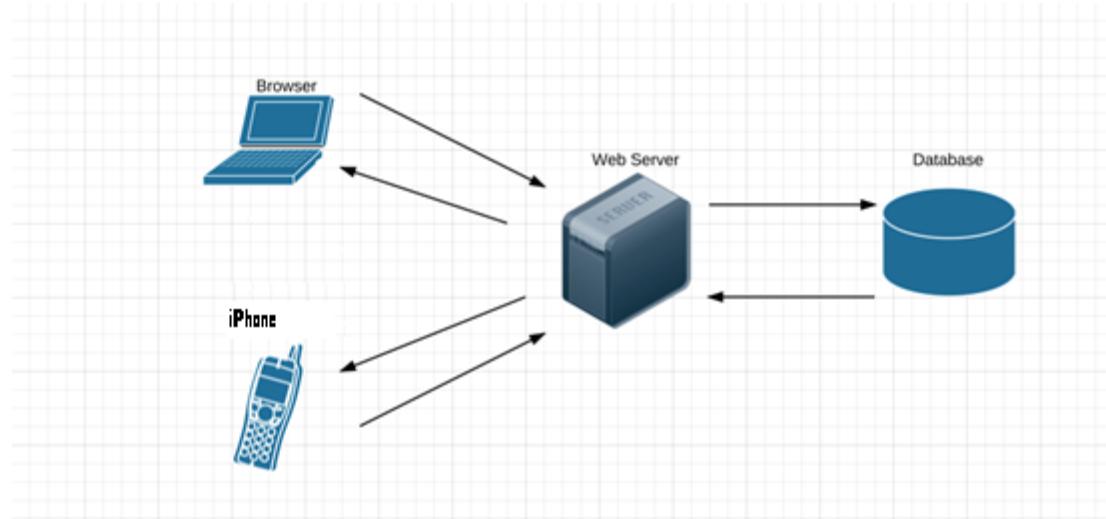


Figure 12 Three Tier Architecture

3.4 System Functionality

This section discusses both the systems basic functionality. As this system is using the Agile Unified process methodology like the Implementation the Identification of the systems functionality will be developed iteratively. It will start with the systems basic fundamental functionality, which is taken from projects description. The 1st iteration functionality's was used to begin design of the User Interface, which in turn enabled most of the other functionalities to be identified. The basic ones are described below

3.4.1 Login system

The login system does exactly what it says on the tin. Its basically a user verification system takes in a user's Username/password combination. It checks these credentials against the systems database to see if the details entered belong to a valid registered user. If so, the user will be logged in. A logged in user can then go on to use all of the available functionality of the web application, such as "create auction" or "join Game". A logged in user can also log out at any time. A non logged in/registered user can still view certain pages of the site. These users can view and search for different auction. The reason behind this is to try and encourage the user to register for the system after seeing something they've like while browsing through the site.

3.4.2 Inventory

The inventory system is the users virtual inventory of items they have added to their profile. Its basically a list of items each with the name and title image of the item. An item can be anything at all that belongs to the user, anything from mobile phones to books to money to vehicles. At any time a user can add an item to their inventory list. Each item has 6 elements belonging to it. They are: Title, Description, Category, Condition, YouTube link and images. The title of an item can be something like "iphone 4 16 GB". Its designed to be a short description of the item and has to have a max size. The description of an item is a more longwinded explanation of what an item is and how it functions. The category and condition are dropdown boxes and will contain options like books, electronics and poor condition, Like new condition. The user will also be able to add images to the item either uploading image files directly from the machine or by using the SwappaH android application which allows user to upload images from their mobile device. When the user wishes to create an auction, join a game or even place a bid. He/she must choose an item from their inventory. The item must also be available and not already up for auction or top bid in another auction.

3.4.3 Auctions

An auction in the web application will be like any other auction but with one main difference of bidding with other items instead of money. When a user wishes to create an auction they would choose an auction length. For example 2 weeks. They will also choose weather or not to allow questions to be asked within their auction. And finally they will choose what item from their inventory the want to put up for auction. When an auction is created it can be found by searching for it or for the user who created it Can be seen in their "auctions" section.any user can then place bids on the auction. The auction will have a countdown clock. Counting down to the end time of the auction. Once the counter has reached zero the auction has ended and no further bids can be placed. The auction is then removed from the public live auction system and placed into the users finished auctions system

3.4.5 Bids

The bid system allows users to offer one of their own items from their inventory for the item of another user. A bid is placed in the auction section of the web application. When a user finds an item that they like up for auction they then have the choice weather to bid on the item or not. If the user wanted to bid they just clicked the "place bid" button and chose the item they wished to bid from their inventory. Once the bid was placed the user who owned the item up for auction would receive a notification say they have received a new bid. That user can then go on to accept the bid or reject it. If the bid was accepted then that accepted item will then become top bid. If rejected then the bidder user must try bid something better. Either way the bidder user will receive notification of wether or not the bid was accepted or rejected

3.4.6 Games

The game section of the web application is a more fun way and quicker way of swapping items. The aim of the game is to perform as many swaps as you can within the game time frame and try gain a better item each time you swap. Each game can consist of any number of players from 10-100,000. Each player in the game will have chosen just 1 of their items to enter the game with. Once the game has begun they can search and swap their item for any other in the game. To do this they must find an item they want. Hit the “offer swap” button and wait for a response. If the other user accepts the swap they can then use the chat feature to chat with each other and decide when and where to meet up and physically swap their items. Once they have physically swapped their items they then go on to use their new item in the game. And try swap that item for something else. The user can continuously do this until the game eventually ends. The users can see their swap history in the finished games section.

3.4.7 Chat

The chat feature in the application is crucial as it provides a way for the users within the system to communicate with each other when they have made a swap. A swap can be made in two ways. Either one user has won an auction or 2 users have decided to swap their items within a swappah game. Either way the users must communicate with each other to decide a location to meet up and swap their items. It's basically a pop up chat box with a send button. The users can type message and send it to each other.

3.4.8 Android Application

The android application is an extension of this project. It's not the main focus of it but it is a very handy and a nice way to make a user's experience simpler. The main function of the android application is to provide a simple and easy way for a registered user of SwappaH.com to upload images of items to the site. The main screens of the application will be a login screen, item list screen, and image view screen. When the app is started the user will be asked to enter in their username and password. If the credentials belong to a registered user they will be brought to the item list screen. This will show a list of all the user's items within their inventory. The user can then choose one of these items. They will then be brought to the image view screen where the user can scroll through all of the 5 images belonging to the item. If any of the 5 photo's do not exist then they will see a default noimage.jpg image. While viewing any of the photos the user has the option to change the image. They would click the change image button and choose an image from the phone's gallery. That new image would be uploaded and either replace the existing image of the item or be placed as one of 5 images that make up the item. By returning back to the log in screen the user has logged out.

3.5 Initial Use Case diagrams

3.5.1 Initial Use case diagram for SwappaH.com containing basic functionality

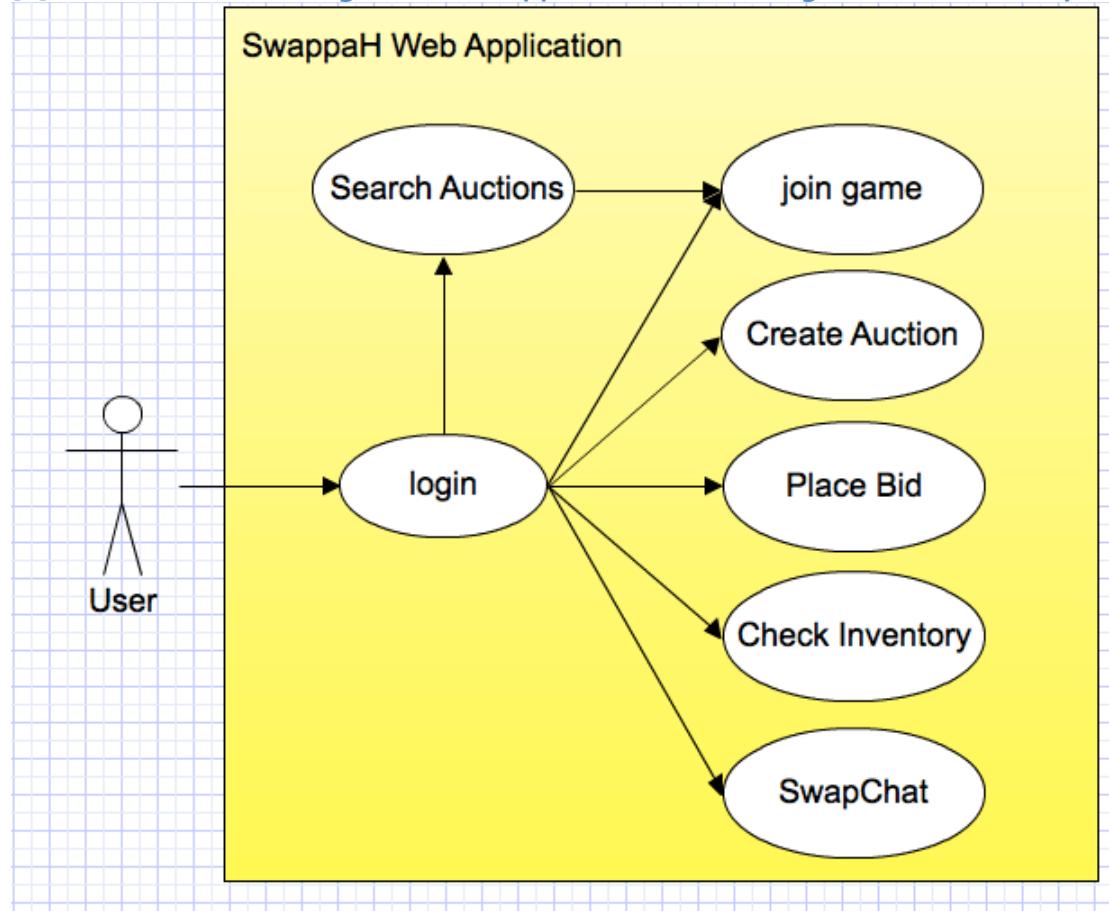


Figure 7 Use Case Diagram of the Basic functionality for web application

3.5.2 Initial Use case diagram for SwappaH Android App containing basic functionality

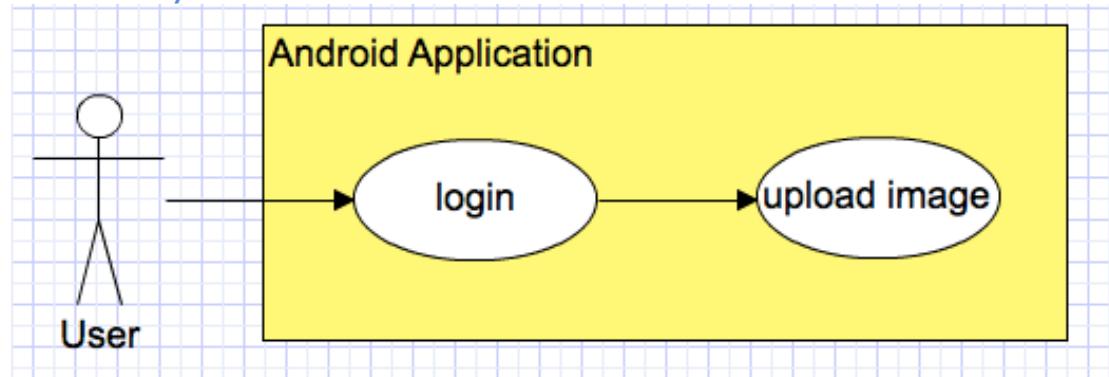


Figure 8 Use Case Diagram of the Basic functionality for the Android Application

These use case diagrams outline the basic functionality of both the web application and the Android Application. As you can see they are quite small and basic as these

use cases do not show all the possible future use cases which were currently unidentified. To identify these the User Interface must first be designed.

Figure 7 and Figure 8 has two main components to it, these are:

- Actors
 - As this is an open website, there could be only one actor using the website at a time. This actor could have two possible states to it. That of a logged in registered user and a non-logged in user. The logged in user would have use of all the functionality of the website. These include creating an auction with one or more of their items, joining a SwappaH game and bidding on items. The non-logged in user would have limited use of the functionality. They would only be able to browse auctions and view these individual auctions. If they wanted to bid on an item or put an item up for auction they must first register to the website. As for the Android App the user **must** be logged in to use any of its functionality. The app is intended to be used by already registered users

3.6 Initial Database Design

3.6.1 Introduction

This design is the initial skeleton of what will become the final complete database design. It was designed using the basic functionality already identified. To design a correct and working database for this system we would first identify all of the functionality and features of the system and design the database around that. First is to create an ERD diagram. This diagram will show all the tables, attributes and relations within the database. Next using this ERD a schema can be generated to create the database within the selected Database Management System.

3.6.2 ERD Diagram

The ERD diagram in figure 10 is just a preliminary diagram. Once all of the functionality of the system is identified the ERD will be re-visited to be expanded further, incorporating the new features.

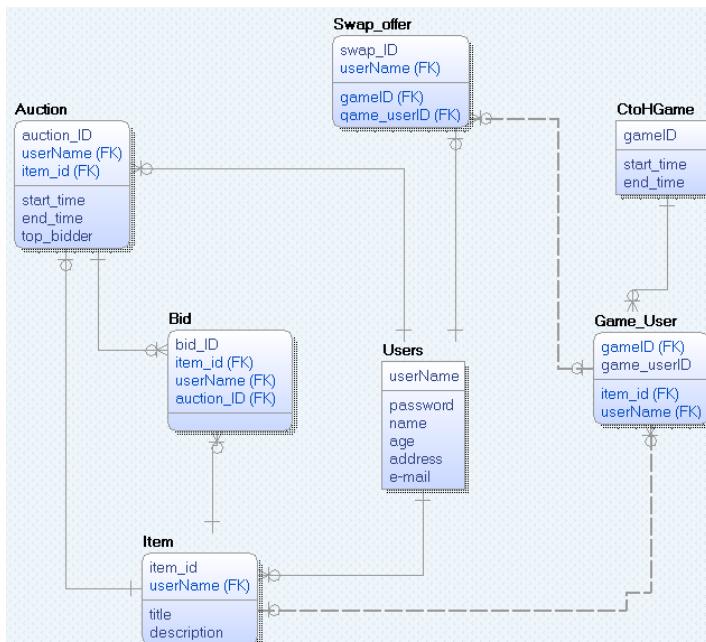


Figure 9: Database ERD

3.7 User Interface

3.7.1 Introduction

Now that I had identified and had a good understanding of the basic high-level functionality it was time to start the design of the user interface. By designing the user interface I was able to identify ever-single bit of functionality that I had not even considered in the broad sense. The design of the user interface was of huge importance, as the intended user of this web application is anyone with any level of computing knowledge. The user interface had to be simple, slick and easy to navigate for anyone. I decided to do 2 stages of design for the user interface stage 1 being a low fidelity paper prototype of the ui. Stage 2 UI design was a more refined medium fidelity prototype. Each of which are explained in detail below.

3.7.2 Low fidelity “paper Prototype”

A low fidelity prototype consists of drawing out the UI design on pieces of paper. Instead of paper the project used a small white board for easy changes and corrections. each eventual screen of the website was drawn out on the white board in a specific order from index -> inventory -> auctions -> games -> swapchat etc. As a whiteboard was being used It was simple to make changes to the design after drawing it. In total there was about 30 different drawings of screens each of which was accepted. Because the project used only 1 small whiteboard, pictures of each drawing were taken with a phone and stored on a machine to work off in the second phase of the UI design. This stage of the design was crucial as it laid out the foundations of the finished user interface

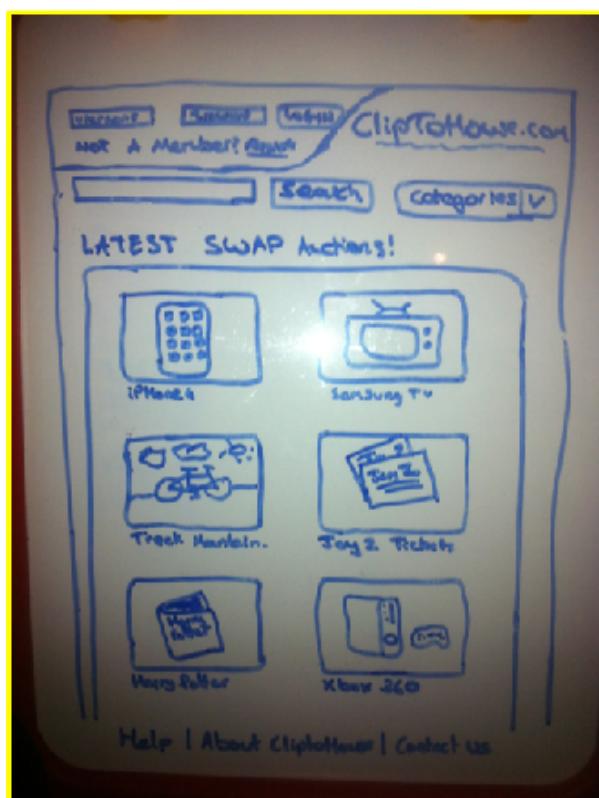


Figure 10 low fidelity index page

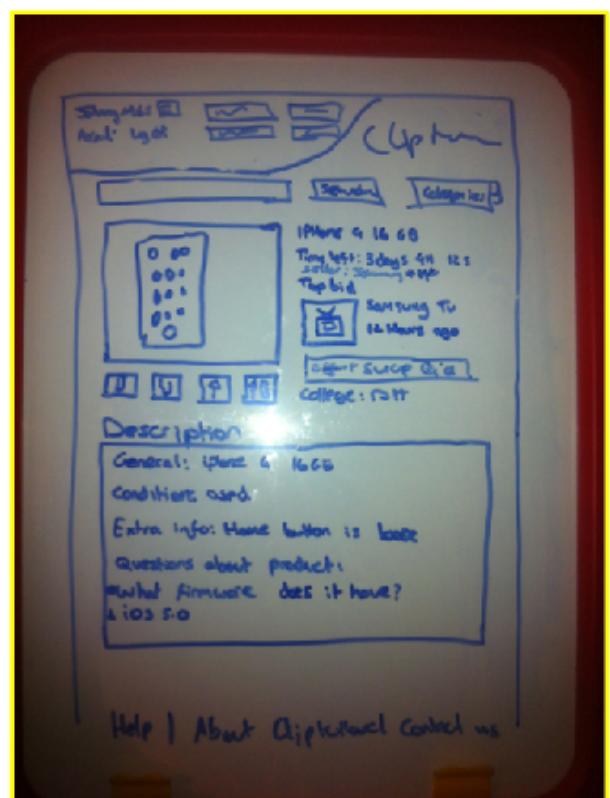


Figure 11 Low fidelity Auction Page

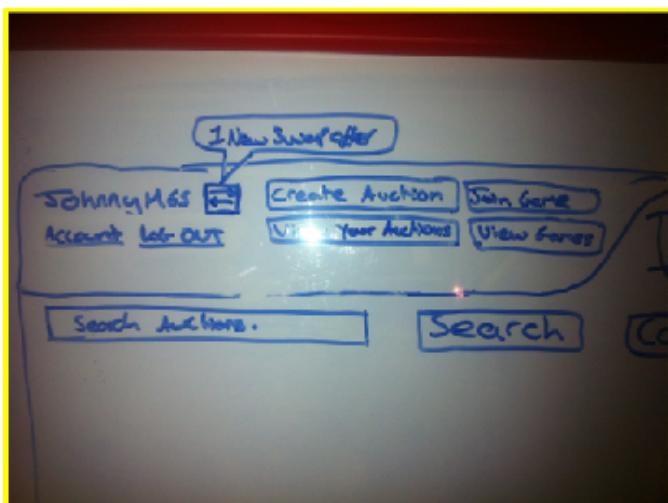


Figure 13 low fidelity Top Menu



Figure 13 low fidelity Inventory

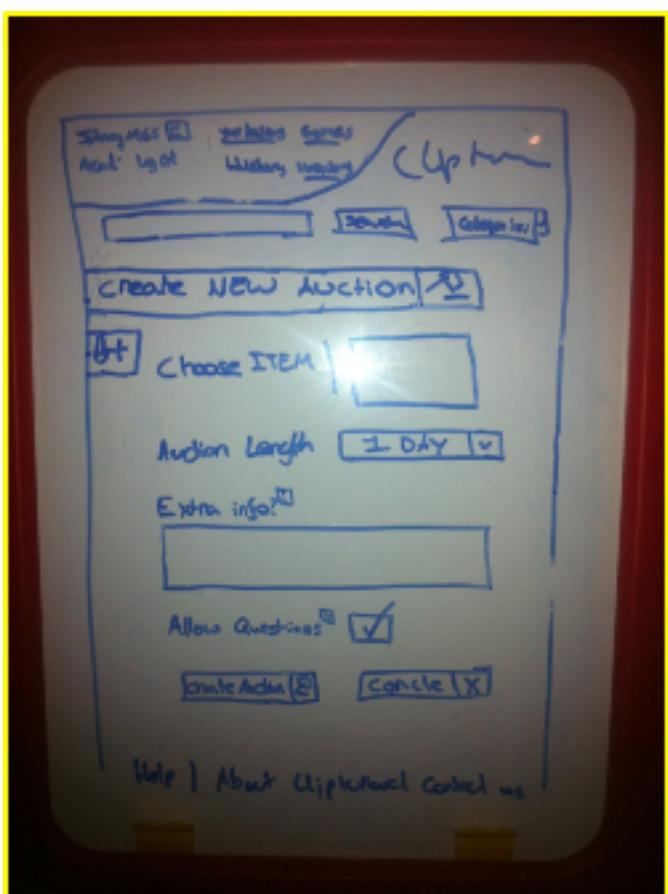


Figure 14 low fidelity create new auction page



Figure 15 low fidelity Your Current Auctions page

3.7.2 Overview

These images are just a few of the many low fidelity paper prototypes that were designed in this fashion for the SwappaH. Each page of the future website was roughly designed. The design for each page showed the page layout along with simple button design and the various links from page to page. These prototypes allowed me to identify many more functionality's that that of the basic six mentioned before. But there were things that these low fidelity prototypes were not able to convey. Such as the colour scheme that the website will follow with the various parts of the sight being certain colours, for example Image borders being yellow. They also could not convey the style of fonts used throughout the site. Fonts are a huge part of the user interface design as they can slick and smooth or bare/tacky. To solve this problem a second series of prototypes were designed. These prototypes were medium fidelity.

3.7.3 Medium fidelity Prototyping

The medium fidelity prototypes consisted of a more refined version of the low fidelity prototypes. It was a lot more close to what the finished look of the user interface would be like. the project used software such as Microsoft paint and Paintbrush for Mac to design the medium fidelity prototypes. Designing a medium fidelity allowed me to take the original paper drawings and play around with colors and shapes to see which complements each other. Colour is a huge design consideration when developing a website and most websites have a colour theme in which they would follow. So this prototype allowed me to identify my colour theme that will run throughout the site. With this prototype I was then able to go on and identify every single piece of functionality that was not already identified by the low fidelity prototypes. The aim was to have the finished site look exactly like this prototype.

I decided to draw 1 more prototype, this low fidelity prototype would show the basic colour scheme the website would follow and would be the basis of the next series of medium fidelity prototypes to come



Figure 16 Low Fidelity with colour scheme

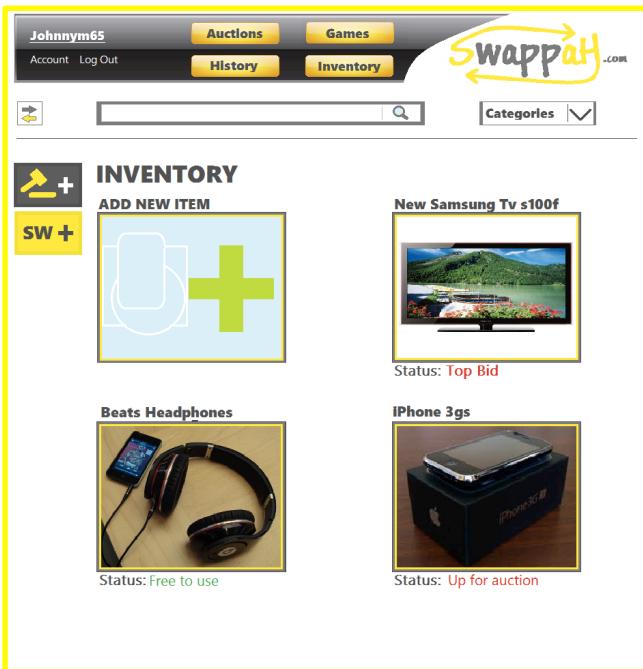


Figure 17 Medium Fidelity Inventory Page

This medium-fidelity wireframe shows the 'Create New Auction' page. It features a title bar with 'Create New Auction' and a swap button. Below it is a 'Choose Item:' field with a placeholder box. Underneath are fields for 'Auction Length:' (set to '3 DAYS') and 'Extra Info:' (with a large text area). A checkbox for 'Allow Questions:' is present. At the bottom are 'Add' and 'Cancel' buttons, along with links for 'Help', 'About SWAPPAH', and 'Contact Us'.

Figure 18 Medium Fidelity Create New Auction Page

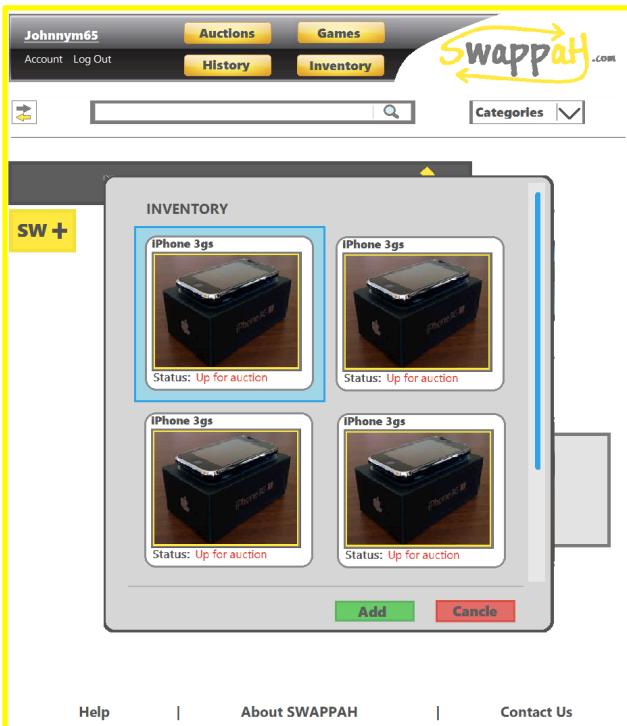


Figure 19 Medium Fidelity Choosing item to Put up for auction

This medium-fidelity wireframe shows a detailed view of an auction for an 'iPhone 4 16GB Black'. The top right shows the item thumbnail, time left (3 Days, 4 Hours, 12 min, 32 Sec), college location (DIT Kevin's Street), and top bid information (Samsung 50" Tv s100f, Bid Accepted 11 Hours Ago, Offer Swap). Below this is a 'Description:' text area. The 'Condition:' is listed as 'USED'. A 'Questions and Answers' section contains a question about accessories and an answer indicating they come with everything. Another question asks about faults, with an answer stating none.

Figure 20 Medium Fidelity View Auction Page

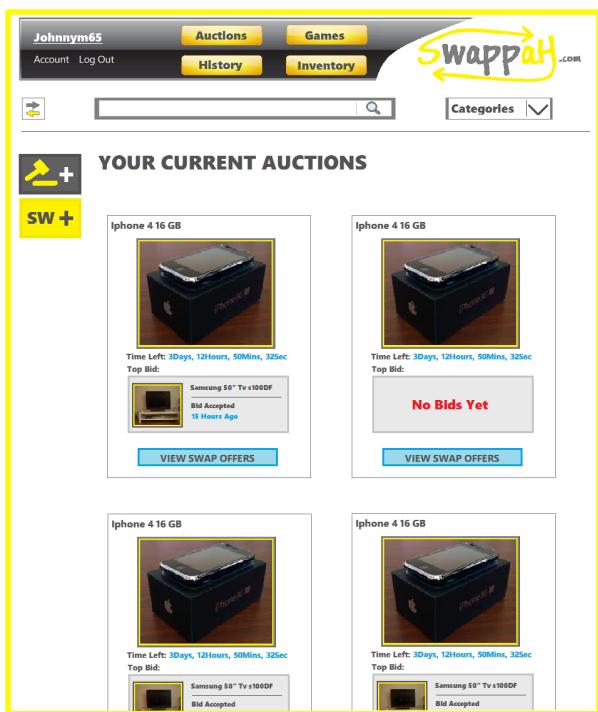


Figure 21 Medium Fidelity Users Current Auctions Page

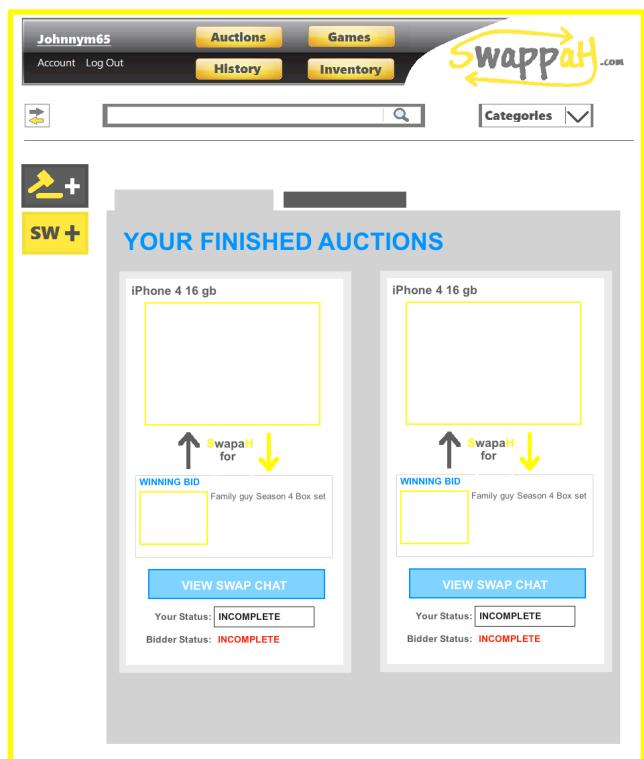


Figure 22 Medium Fidelity Users Swap Section Tab Page

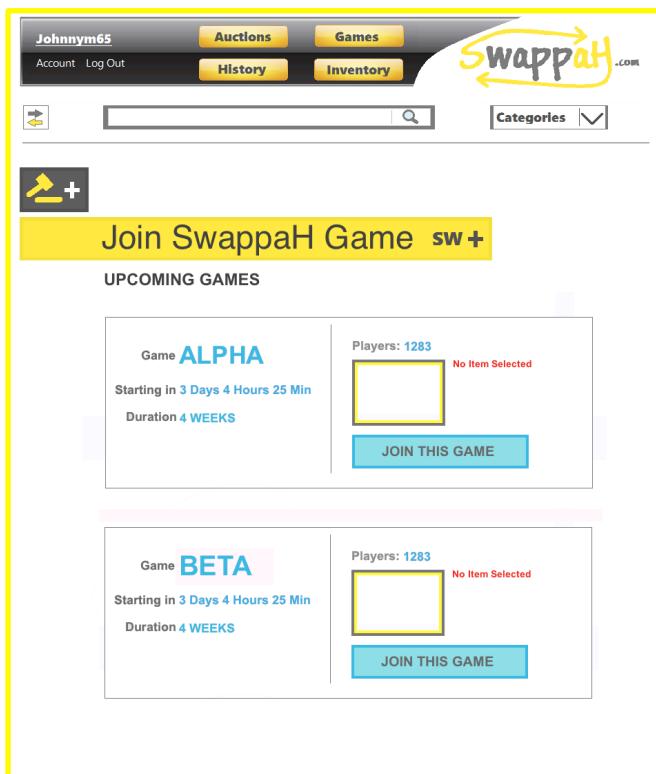


Figure 23 Medium Fidelity Join SwappaH game page

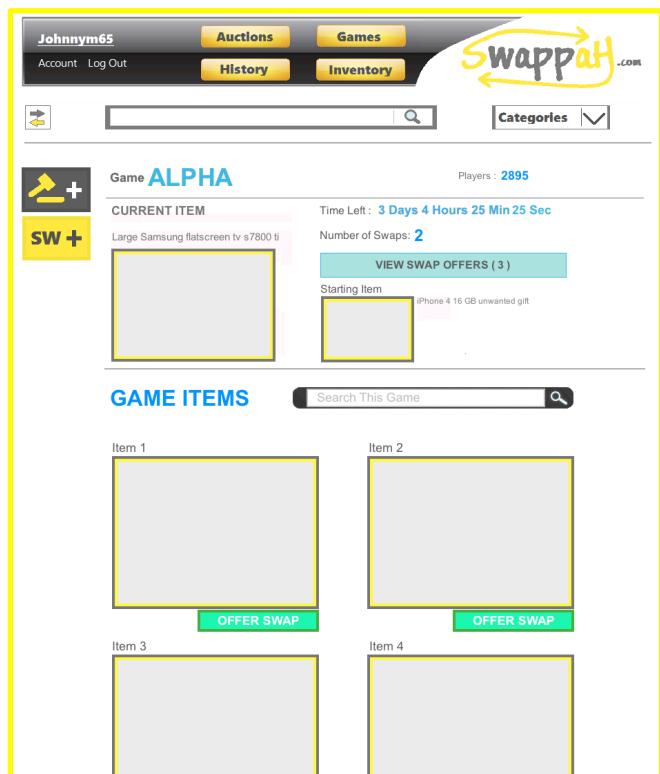


Figure 24 Medium Fidelity View Game Page

3.8 Other Design Considerations

3.8.1 Image storage File system Vs BLOB (Binary Large Object)

I was having problems coming up with a solution as to how I would store all of the item images for every item of every user. The first solution I came up with was to store them directly in the Database using a BLOB (Binary Large Object). There were some issues with using BLOBS that made me decide not to use it. These consisted of

- Size of the Database

Because you were storing the image files directly inside the database the database would grow larger and larger as the amount of users and images grew and grew. Eventually the database could become massive in size which would cause problems such as making difficult to back up the database regularly

- Speed of retrieval

The speed of retrieval using blobs is also slightly slower than that of a file system, on a website like this the speed needs to be quick and smooth as possible as one page could contain many images.

I then decided to use the filesystem itself to store the item images.

3.8.2 Use of YouTube

I designed the incorporation of the use of youtube into the system, the user can enter in a youtube link when they are creating a new item if they wish. The link should correspond to a youtube video that they themselves have uploaded. The video will appear when the item is viewed, as well as inside an auction page for when an item is up for auction, for potential bidders to view the item on video. The fact that the feature will allow a user to show their item on video will have many more advantages than just images as I can show how a certain item is operated. Also it can prove the item works or is in the condition specified.

3.9 Use case Diagrams

These use case diagrams were created using the functionality identified.

3.9.1 SwappaH.com Web Application Use Case Diagram

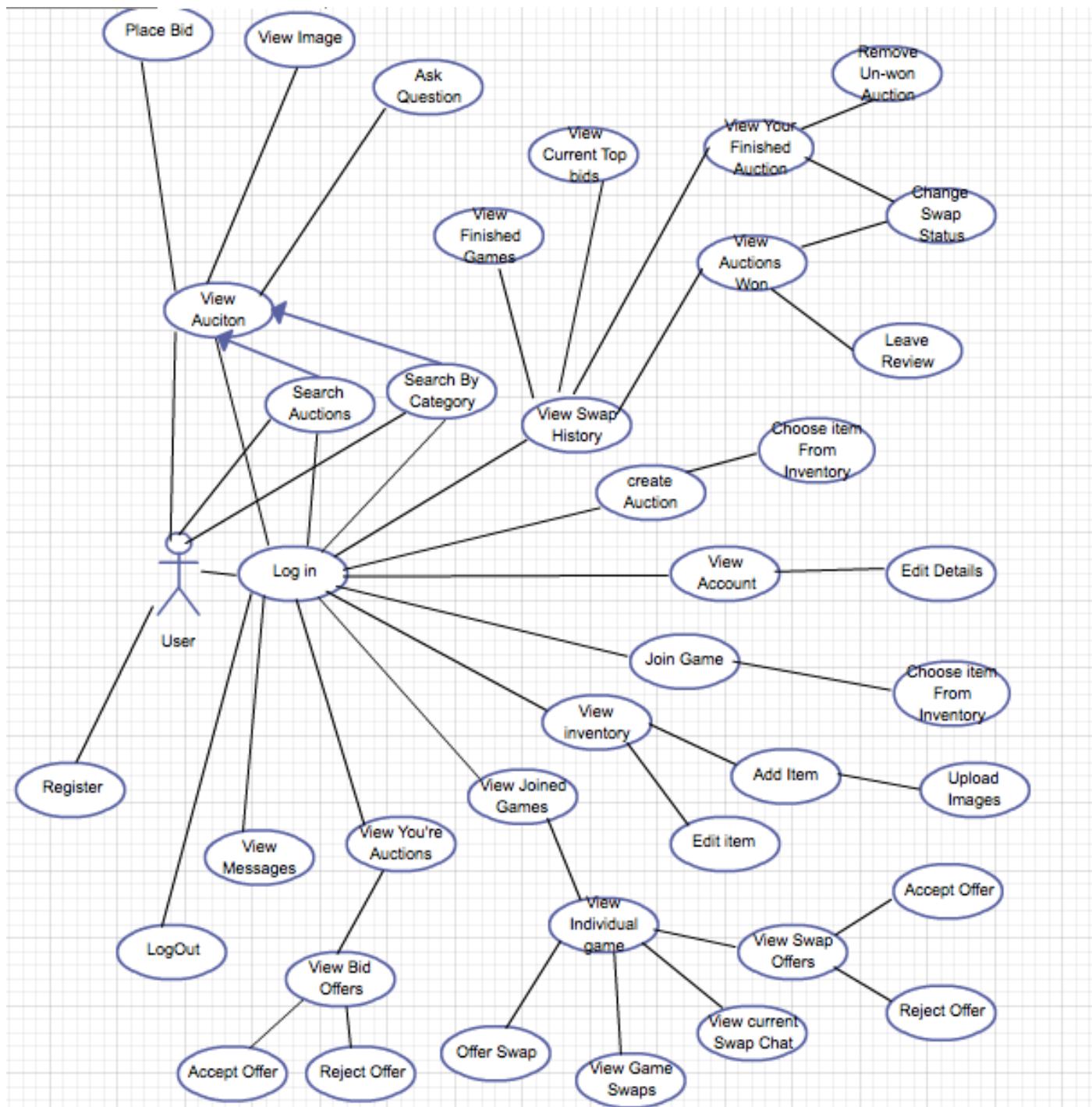


Figure 25 SwappaH Web Application Use Case Diagram

3.9.2 Swapah Android Application Use Case Diagram

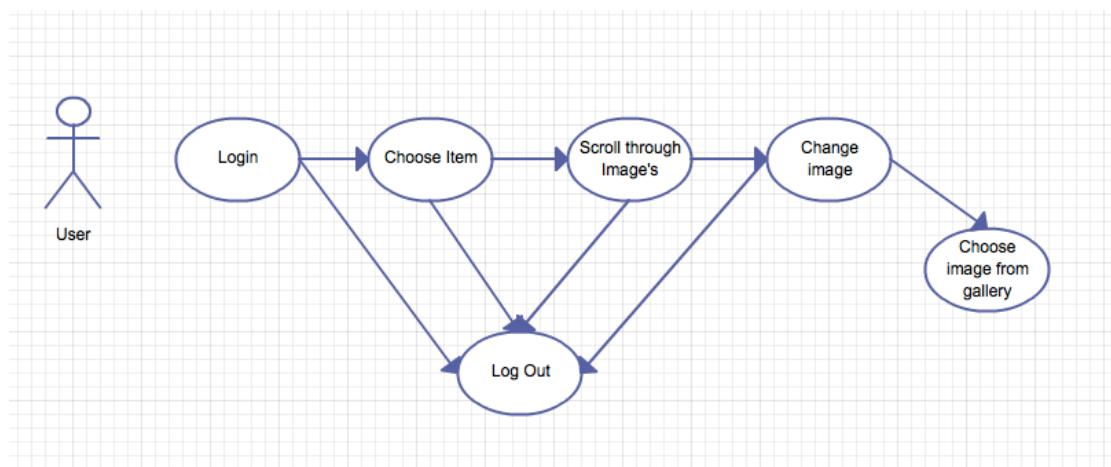


Figure 26 Swapah Android app Use Case Diagram

3.10 Database Entity Relationship Diagrams (ERD)

3.10.1 ERD Diagram without relations for easier viewing of table

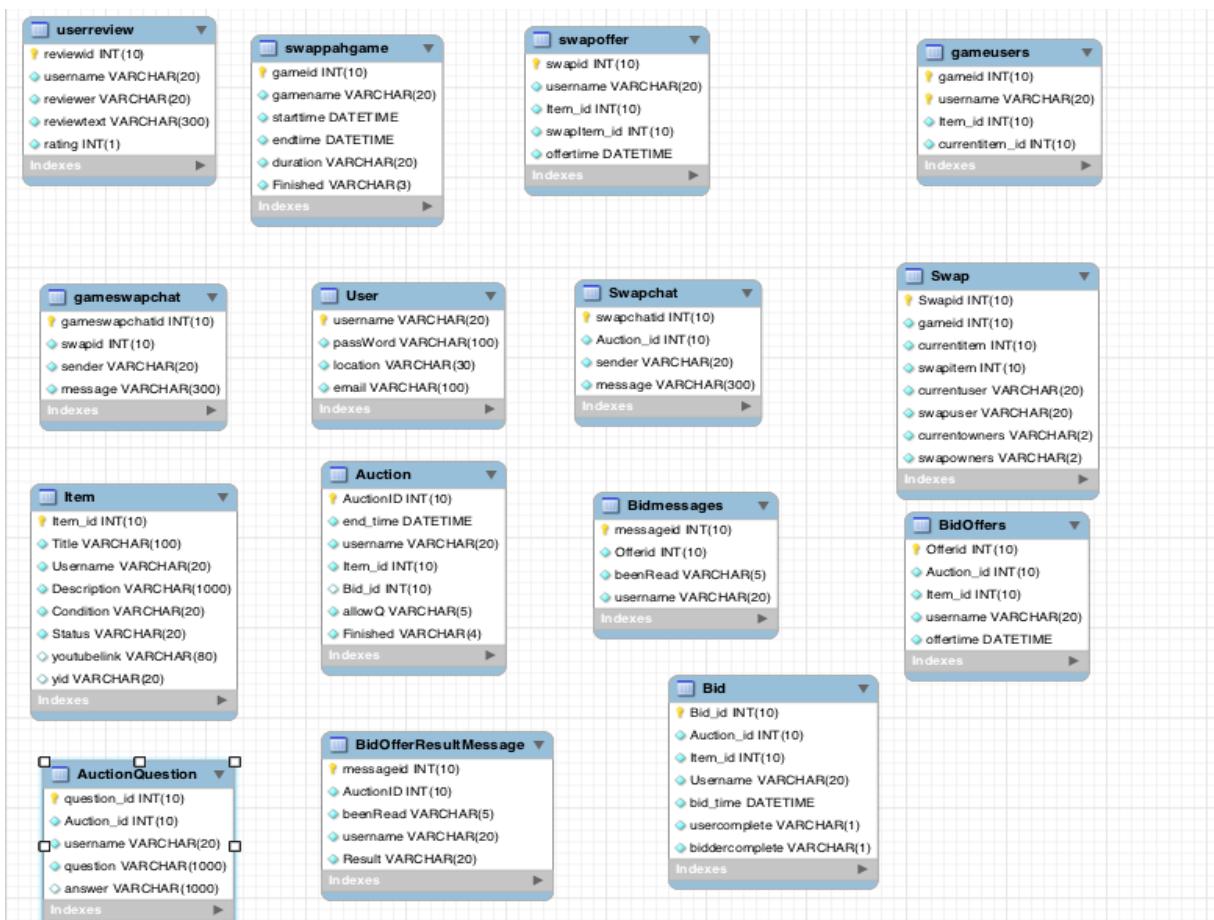


Figure 27 ERD Diagram without relations for easier viewing of table

3.10.2 Final ERD Diagram including relations

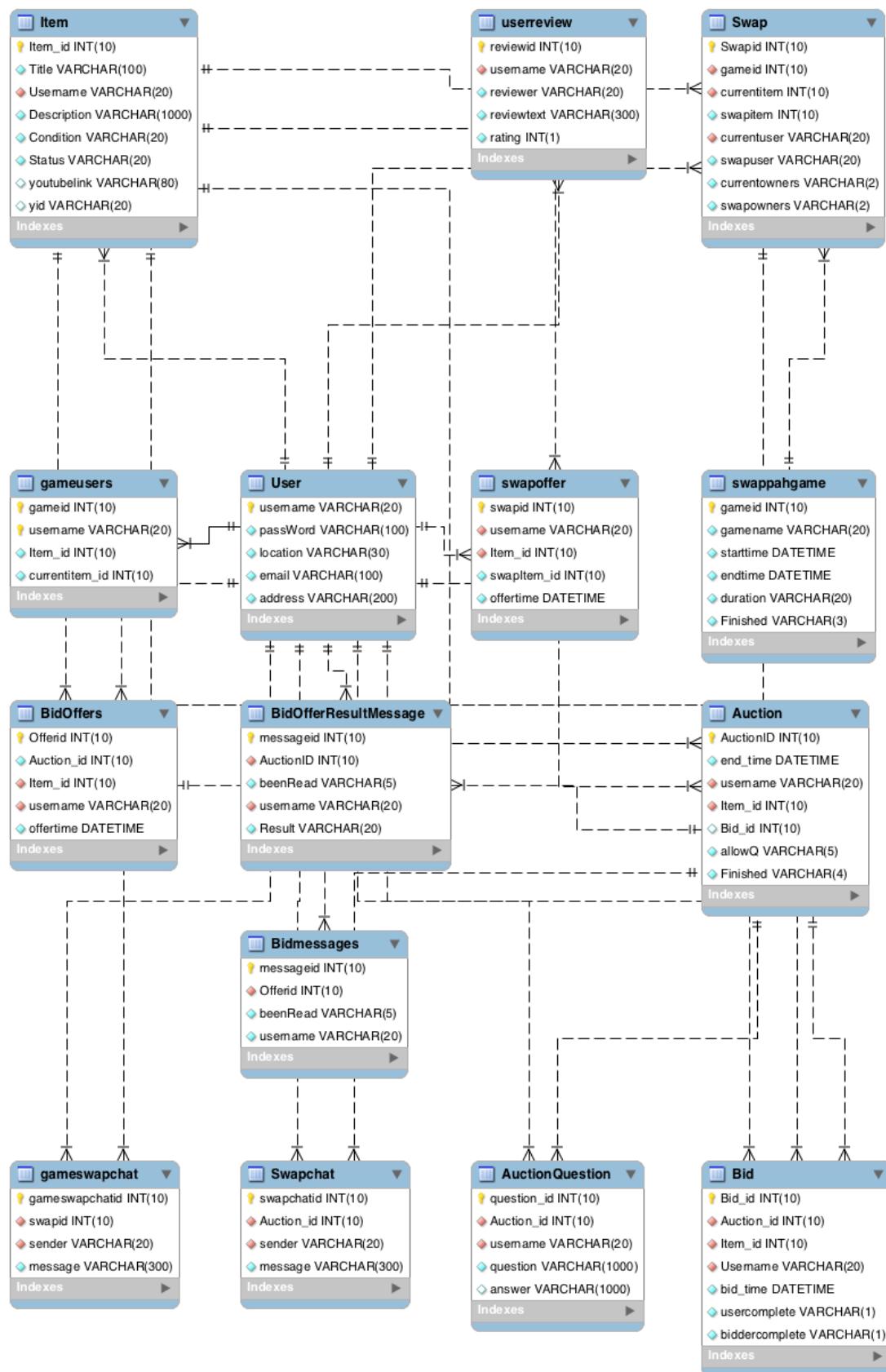


Figure 28 ERD Diagram Inc Relations

Chapter 4 Implementation

4.1 Introduction

This chapter outlines the steps that were taking in the actual implementation of both the web application and the Android application. It begins by describing how the local testing environment was established. It then goes through each individual task that was completed and in the order it was completed. As this project follows the Agile Unified Process methodology the system will be implemented in an iterative fashion, in that it will be broken up into different tasks. For example login system will be the first iteration of the project, then Inventory section second and so on. Each Iteration will add a new feature to the project until its completion. These iterations are described in detail .

4.2 Setting up Environment - Local MAMP Server

MAMP as described in chapter 2 is a software stack used for the local development of web applications. It consists of Mac osx which is the operating system the local server will run off. Apache which is the server it will use. MySQL which is the RDMS(relational Database Management System) and PHP which is the server side scripting language being used. Setting up the local environment was quite simple as the official MAMP website <http://www.mamp.info/> provides all the elements of the stack in 1 download and installment. The stack is installed as one program on the machine called "MAMP". When the program is run it gives the user options such as "start servers" "preferences". I simply had to choose a location on my machine to have the web application located along with what port I wanted the server to use, click the "start servers" button and I was ready to begin. During development I could view the web application in a browser by using the localhost address <http://localhost:8888>, 8888 being the port I chose. It could also be accessed by any device on my Local Area Network by typing in my machines static local ip address <http://192.168.1.100:888> which was handy when I wanted to test the application on other operating system and browsers.

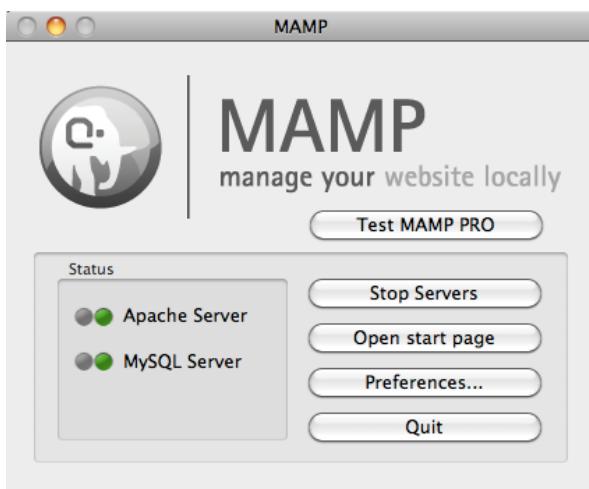


Figure 29 MAMP

4.3 Implementing section by section (using basic functionality)

4.3.1 Overview

The registration system of this application would allow new users of the system to become registered users. A registered user can then log in and make use of all the different functionality the site offers. The registration system will take in a user's details, validate them and store them in the database. Once this is done the user becomes a registered user and can then log in. The idea of the login system of this application will be that it can allow people to validate themselves as registered users of the website. The reason why people would want to be a registered user and log in is because only logged in users can avail 100% of the functionality's within the system. If a user remains logged out they would only have the options to "search Auctions" and "view Auctions". To validate a registered user the system will ask the user to enter their credentials. Which consist of a unique Username and associated password. It will then verify the credentials which will allow/deny access to the rest of the website. The login system will also contain a Register system. Which will allow new users to register to the system.

4.3.2 Registration system

The implementation of the registration system consisted of creating a html form for the user to input all of their details and a php file that will take in these details, verify them and either return to the html form with an error message or input the details into the database. The html form consists of 8 labeled fields. They are Username, Password, Re-enter Password, Address line1, Address line2, Town/City, County, Email Address and a submit button. These details are then passed to the Register_checker.php file, which stores them in local variables. The first thing Register_checker.php will do with these details is check if any of the major details such as username or address line1 are empty or null also if the Email address entered is a valid format e.g. Something@somewhere.com . If they are an error message will be generated and the user will be redirected back to the registration page along with an error message that would state something like "* Error – Address line1 field Blank".

If all the fields are fine then the script will query the database to see whether or not there is already a user with the same Username. To do this it would ask the database, is there a row in the users table where attribute Username is equal to username entered.

```
1. $result = mysql_query("SELECT * FROM User WHERE username = '$username');
```

the system checks how many rows are returned from the query. If it is 0 then the username is free to use. If its 1 then the username is already taken and an error message is generated and sent back to the registration page. When all the details are valid and the username is free to use then the system will input all of the details into the database using an insert query. After I had implemented this I realized I had a problem. The database was storing the users passwords in plaintext, which should really never be done. So I decided to encrypt the users passwords when they register and store the encrypted password inside the database. I used the

encryption algorithm sha1 which turns plaintext strings into long unrecognizable string filled with numbers and HEX. e.g. "password" → "5baa61e4c9b93f3fo68225ob6cf8331b 7ee68fd8". So when the user would login the system would have to encrypt the entered password to see if it matches the stored encrypted password.

```
1. mysql_query("INSERT INTO User
2. VALUES ('$username', '$hashpword', '$location', '$email', '$address');");

```

Registration Form

The form is titled 'Register' in large blue letters at the top. It contains the following fields:

- *Username: Text input field
- *Password: Text input field
- *Re-enter Password: Text input field
- *Address line1: Text input field
- Address line2: Text input field
- *Town/City: Text input field
- *County: A dropdown menu with 'Armagh' selected
- *Country: A dropdown menu set to 'IRELAND'
- *Email Address: Text input field
-

Figure 30 Registration Form

4.3.3 Simple login system

To start off with I wanted to create a very basic skeleton of a login system. To do this I first created a simple html form. Which consisted of two text fields labeled Username and password and a submit button.

```
1. <form action="check_user.php" method="post">
2.   Username: <input type="text" name="uname" /></br>
3.
4.   Password : <input type="password" name="pword" /><br>
5.
6.   <input type="submit" value="Submit" />
7. </form>
```

I then created a PHP file called Check_user.php. The html form would send the inputs strings for username and password to the Check_user.php page in a post request. Check_user.php would then store these strings in local variables called \$username and \$password. Now That I have both username and password stored as local variables in my Check_user.php I could then move onto the task of validating the user. To do this I would have to make a connection to the systems database.

```
1. mysql_connect("$host", "$username", "$password") or die("cannot connect");
2. mysql_select_db("$db_name") or die("cannot select DB");
```

Once the connection is made I can then go on to designing a query that would let the system know wether the credentials the user entered belong to a valid registered user or not. The query I designed asks the database to return any rows from the table users where the column Username is equal to the local variable \$username and password is equal to the local variable \$password. As the system does not allow duplicate Usernames the number of rows returned from this query can only ever be either 1 or 0. If the number is 1 its means the credentials belong to a valid user. If it is 0 it means that there is no record of the user and therefore will not log them in. The finished query can be seen here

```
1. $result=mysql_query("SELECT * FROM Users WHERE userName='$username' and
password='$password'");
```

Check_user.php checks the number of rows returned from the query. If 1 it will print out "Successful Login" if 0 "Login Failed".

4.3.4 More complicated login system

Now that I have a basic login system set up I can now tailor it to what is required for this project. The First thing I had was change the system so it would encrypt the entered password to check with the already encrypted password in the database. I again used the Sha-1 encryption algorithm. In accordance with the user interface I had already designed I wanted the login system to be seen at the top of the screen no matter what page the user was on in the website. To solve this I decided to create a header.php file. This header file will create a header bar that will appear on every screen on the website. This header bar can have two states. It can be in logged out mode or logged in mode. If its in logged out mode it will contain the two input fields for username and password along with a submit button. This will allow a user to roam around the different pages of the site and log in at any time. If its in logged in mode i.e. the user has logged in successfully, then it will contain the username of the user along with various options such as "account" or "Inventory". Basically features that are only available to valid logged in users. Also when the user has successfully logged in, the system will create a new session for that user.

```
1. session_start();
2. $_SESSION['username']=$username;
```

4.3.5 What is a php Session?

A normal HTML website will not pass data from one page to another. In other words, all information is forgotten when a new page is loaded. This makes it quite a problem for tasks like a shopping cart, which requires data(the user's selected product) to be remembered from one page to the next. A PHP session solves this problem by allowing you to store user information on the server for later use (i.e. username, shopping cart items, etc). However, this session information is temporary and is usually deleted very quickly after the user has **left** the website that uses sessions. (<http://www.tizag.com/> 2008).

Its main use in this project is to store user related data such as username. So when the user browses through different pages of the site their Username goes along with them and if needed the content would be dynamically created for that particular user. For example when the user clicks into the "myAuctions.php" page. The script will use their username to gather their particular auctions instead of someone else's.

4.3.6 Register/Login systems Finished User interface

Not logged in Header bar



Figure 31 logged out header bar

Logged in Header Bar

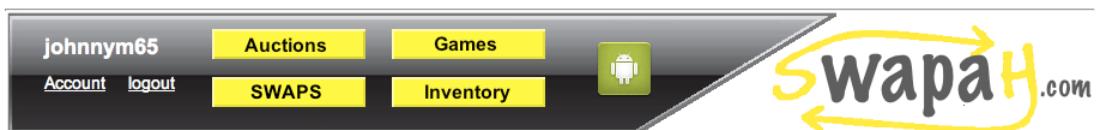


Figure 32 logged in header bar

4.4 Implementation of Inventory

4.4.1 Overview

Now that the first iteration, that of the login/registration system has been complete I can move onto the next feature of the website. This feature is the Inventory system. The inventory system is a place that holds a users items. it is a way for the user to create a new item, view/edit these items and also delete these items. You could think of it to be something like a rucksack that holds different items. These items are used for creating auctions, bidding on auctions and are offered as swaps within the SwappaH games. The items are chosen from the users inventory every time its needed.

4.4.2 What is an Item

An item is a virtual representation of a physical possession belonging to the user. They consist of a title, a description, a condition, a category, a YouTube link which is not mandatory, and a possibly 5 images which again are not mandatory. The first thing I needed to do with the inventory system was to build the newitem.php page.

4.4.3 Creating a new item

Creating a new item started off with the createitem.php page. This page consists of a html form that contained several inputs. These were Title, Description, category, condition, YouTube link, 5 file image upload fields and submit button. The YouTube link and the 5 image inputs were not mandatory fields and could be left blank if desired. When the form was submitted the inputs were then sent to the newitemcheck.php file. This page would take the form inputs and store them in local variables. It would then go on to error check all of these inputs to make sure there was no empty fields or incorrect input. If an error was found the user would be brought back to the newitem.php page along with an error message. It would also check to see wheatear the image uploaded was within the max file size allowed. If the inputs passed all of the checks then the item could then be stored in the database. The item is assigned the user who created it with a the foreign key Username in the Items Table.

The screenshot shows a web-based form titled "Add New Item". The form is divided into several sections:

- Title:** A text input field with the placeholder "*Title: []".
- Description:** A large text area for entering item details.
- Category:** A dropdown menu currently set to "Antiques".
- Condition:** A dropdown menu currently set to "Brand New".
- Youtube Link:** A text input field for the YouTube URL.
- Image Upload Fields:** Five separate file input fields labeled pic1, pic2, pic3, pic4, and pic5, each with a "Choose File" button and the message "No file chosen".
- Submit Button:** A blue "Create Item" button located at the bottom right of the form.

Figure 33 Add a new item Form

4.4.4 image storage

As mentioned in the design section the images will be stored within the filesystem of the application. The big task was to design it so the images could be related to the item itself. The best way I thought of doing this was to simple name the file after the id number that was automatically given to the new item along with the item image number (1-5). For example, if a new item was created and given the ID 285 and two images were uploaded with it. Then these images would be stored in the files system as 2851.jpg and 2852.jpg were the 285 corresponds to the item ID and the 1,2 correspond to the image number(1-5) of that particular item, that way if the system wanted a certain image it just had to search the images for the ones that begin with the particular item id and end in either 1, 2, 3, 4, 5.

1. `move_uploaded_file($_FILES['uploadedfile']['tmp_name'], "image/upload/".$itemID.".1.jpg");`
2. `move_uploaded_file($_FILES['uploadedfile2']['tmp_name'], "image/upload/".$itemID.".2.jpg");`
3. `move_uploaded_file($_FILES['uploadedfile3']['tmp_name'], "image/upload/".$itemID.".3.jpg");`
4. `move_uploaded_file($_FILES['uploadedfile4']['tmp_name'], "image/upload/".$itemID.".4.jpg");`
5. `move_uploaded_file($_FILES['uploadedfile5']['tmp_name'], "image/upload/".$itemID.".5.jpg");`

4.4.5 Viewing particular Item

A view item page is necessary in the inventory/item systems as it allows users to view all of the details of particular items. For instance when a user offers a swap for another users item. The 2nd user will want to view the item in its entirety to see 1. Wether or not they even like the item and 2. Is the item worth the same or more as their own? The viewitem.php page will again take in an ItemID from a get request. It will use this id to query the database and retrieve the details of that particular item. It will also fetch each image of the item from the file system by finding the images named the same as the itemID. Once it has retrieved all the details/images for that item it displays them in the layout designed in the user interface for the user to be clearly able to see all the details of the item.

Family guy Season 4 boxset



Figure 34 View Item Page

4.4.6 Editing particular item

Now that I had implemented creating new items i moved onto how the user could edit these item if they wish. The page that allows this is the Item.php page. This page takes in a itemID from a get request. As in the UI design the edit item page would show all of the details of the item such as title, descriptions, condition ect. The details like description, condition, category, YouTube link are all be displayed within an editable form. This allows the user make changes to these details if they wish. When the user clicks the update button the new details are sent to UpdateItemCheck.php which checks the new details for errors then updates the database with the new information.

5 paintings

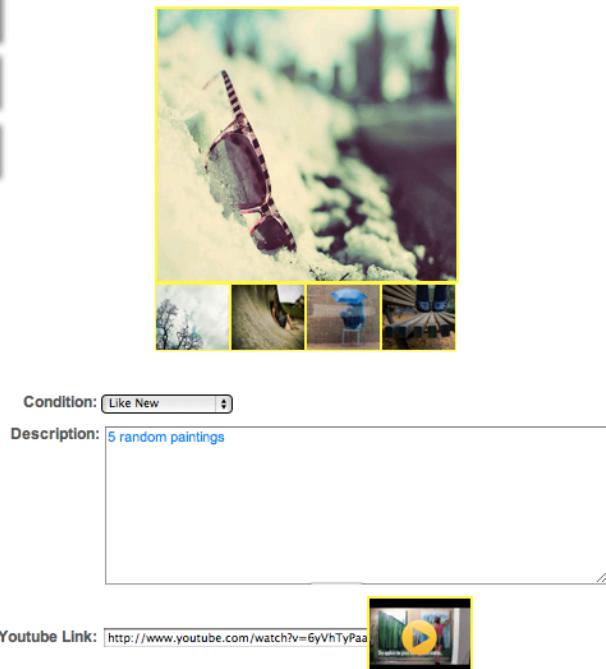


Figure 35 Edit ItemPage

4.4.7 Use of Fancy Box

Fancy box^{iv} is an API that allows piece of html to pop up on screen in a nice elegant pop up box. As you can see from the user interface designs I had always designed the UI to use some sort of pop up box to contain things like forms and images etc. this is exactly what fancybox does. It emphasizes the pop up html while darkening the screen around it. Once the user clicks the x button or outside the box at all then the fancybox disappears. I used fancybox throughout the site for viewing things like images, YouTube videos and also for displaying the users inventory for when it was time for the user to choose an item from it for a particular task such as creating a new auction or placing a bid.

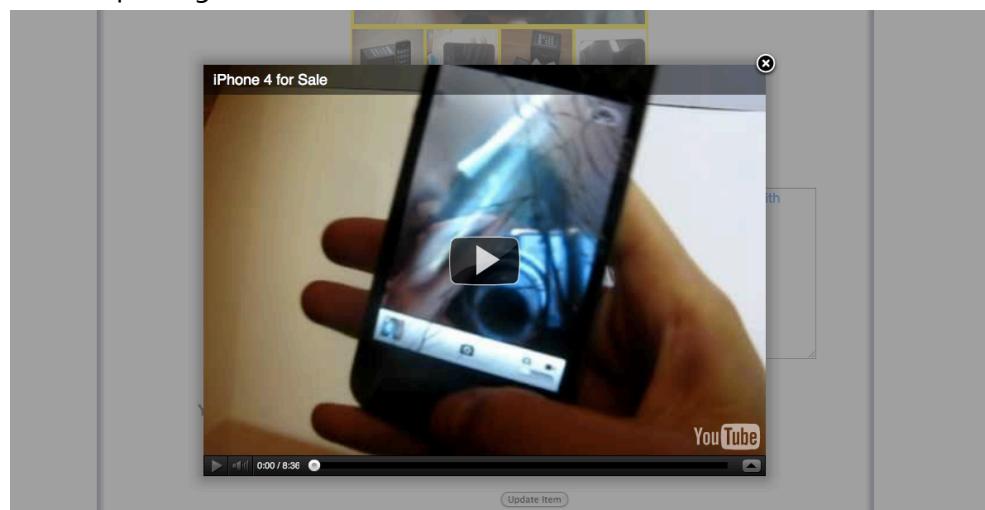


Figure 36 Using Fancybox to display YouTube Video

^{iv} <http://fancybox.net/>

4.4.8Viewing Inventory

The implantation of the Inventory begins with an sql query, this sql query queries the item table in the database to find all of the items that belong to the logged in user. The query then returns every row where the Username attribute of the item is equal to logged in user. This sql query can be seen below.

```
1. $result = mysql_query("SELECT * FROM Item WHERE Username = '$username'");
```

to display the details of every item that the Query above returns I created a while loop that will loop every time there is a row. Seen below

```
1. while($row = mysql_fetch_array($result));
```

Each time this While loop runs a new item is displayed. So if a user had 5 items stored in the database then the Query would return 5 rows which means the while loop would loop 5 times and in turn 5 items would be displayed. The items displayed do not show every detail of the item as there simply would not of been enough room so as designed in the user interface the Title, The First image and the items current status. The status of an image is determined by wether or not the item is involved in anything such as being up for auction, being top bid in an auction or chosen for a game. These are the 3 main details of an item that the user would need to see. If they wanted to see all its details or edit the item they could simply click on it which will take them to the Item.php page.



Figure 37 Individual Item in Inventory

Before any of these items are displayed I have the “new Item” tab which if click

brings the user to the newItem.php page. As in the User interface design the items within the inventory are laid out in a 2 column format. Were Column 0,0 is the “new item” tab. To achieve this layout a new row was started every second time the while loop was called.

Figure 38 Users Inventory

4.5 Implementation of Auctions

4.5.1 Overview

Now that the Inventory/item system has been implemented the next stage of functionality to be made is the Auction system. The auction system would consist of 5 main parts. These are create Auction, View Auction, your Auctions, Search Auctions, and index page of the system that would show the latest auctions to be made. As can be seen in the description of the application and the User interface design the main features of an auction are the item up for auction itself, the countdown timer, the top bid and any auction questions.

4.5.2 Creating a new Auction

The creation of a new auction would begin with another html form in the file NewAuction.php this form is where all the details of the auction will be chosen by the user. There are only 3 different elements to this form. They are the chosen item to be put up for auction, the length of time the auction will last (3 days, 5 days, 1 week, 2 weeks) and a checkbox asking if the user would like to allow questions. When selecting an item to be put up for auction the user must choose one of their items from their inventory. Once the "item square" is clicked a fancy box will appear containing the user's inventory. The user would simply click on the item they wish to put up for auction. Once chosen the "item area" will be filled with the selected item's image along with its title. The user can change this item to another if they wish by clicking it and choosing a different item from their inventory. Next the user chooses the length of the auction and whether or not to allow questions.

These details are then sent via a post request to the NewAuctioncheck.php page which will take in the 3 details ItemID, AuctionLength, AllowQ. The details are then error checked and validated. The next thing it does is calculate the date the Auction should end by taking today's date/time and adding the length of the auction in days. For example today's date/time is 15-03-12 16:25:40, if the user chose a 2 week auction then the end date would be calculated as 29-03-12 16:25:40. The code can be seen here

```
$EndTime= date('oy-m-j H:i:s', strtotime("+.$Auctlength." days"));
```

Once this is done the auction can be inserted into the auctions table of the database using a query. The Auction row will also contain two foreign keys, the username of the user who created it and the itemID of the item up for auction. It will also default the current top bid as 0 as there has not been any bids.

Create New Auction

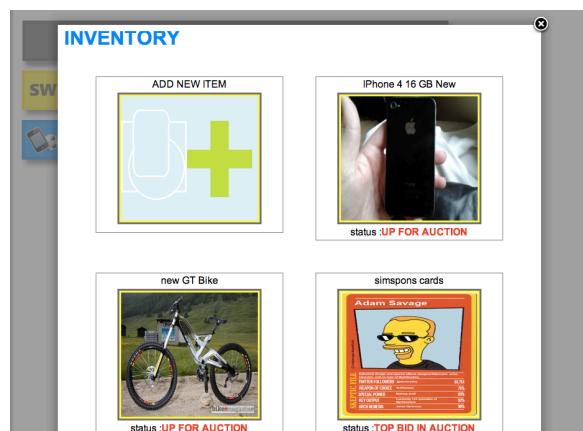
Choose item:

Auction Length: 3 days

Allow Questions?

Create Auction

Figure 39 Create New Auction Form



Choose item:

Collection Of old Vinals

Figure 41 Chosen item appear in item area

Figure 40 Pop up inventory for item choice

4.5.3 Auction page

Once an auction can be made the next step would be to view an individual auction in its entirety. Being able to view the images and all the details about the item up for auction, the countdown timer telling you when the auction will end and the other details like YouTube video or Questions asked. The first thing to do was to get all of the details of the auction, this would involve 2 query's. One to get the auction details itself such as end time and current top bid and the other to get all of the details of the item up for auction. All of these details were stored in local variables inside Auction.php. The first thing on the list to do was to create a countdown timer.

4.5.3.1 Countdown Timer

This involved getting the end time of the auction and the current time. These times were gotten in the format YY-MM-DD HH:MM:SS. They were each broken down into their separate sections Year, Month, Day, Hour, Minute, Seconds. These variables were then passed into a JavaScript function called GetCount(). The function get count would take all these variables in and calculate the difference between them. The output would be something like "3 days, 2 hours, 10 minutes, 4 seconds".

This JavaScript function would be called once a second creating a countdown clock.

A problem then arose. The current time the function was using was taken from the user machine. Therefore the timer would display different time remaining on different machines. Which cannot happen in an auction site. The way around this was to pass the current time off the server machine to the function so that the time remaining will be the same no matter what machine their on. The end result can be seen as

Time Left:
2 days, 19 hours, 12 mins, 30 secs

Figure 42 Countdown timer

Once the countdown timer reaches zero the message "finished" will appear. No more bids are allowed on the auction.

4.5.3.2 Auction questions

If the user had chosen to not allow questions on their auction then the question section of the auction will just display "Questions have been disabled for this auction" but if they are enabled then a user viewing the auction will have the option if they wish to ask the Auctioneer user a question using a textbox within the questions section. The auctioneer user can then answer this question with a new textbox that appears. Each question is inserted into the AuctionQuestions table within the database. Until the auctioneer user answers the question the message "no answer from user yet!" will appear.

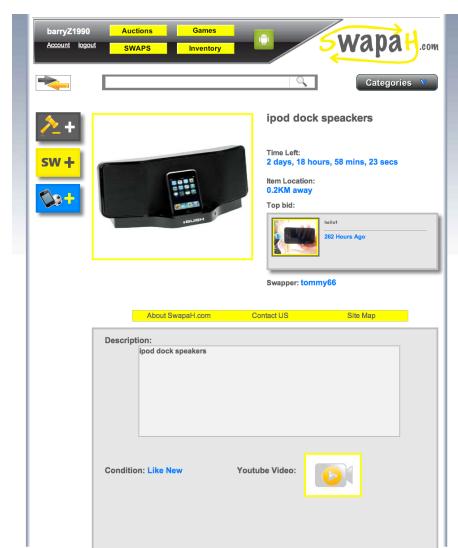


Figure 43 Auction page

Figure 44 Close up of Question section in Auction Page

4.5.3.3 Bids

Now that an auction can be created and viewed the next stage of implementation of auctions was to allow bids. As described before a bid is a user offering to trade one of their items for the item up for auction. A user should be able to bid on an item by getting into the auction section and clicking the "place bid" button. There are a few reasons why a user might not be able to bid on an item. They are

- o The auction has ended
- o The user is not logged in
- o The user is already the current top bid
- o The auction belongs to the user trying to bid

If the user passes these criteria then they will be allowed the bid. The beginning of the bidding process is similar to the creation of a new auction in that the users inventory will appear in a fancy box. The user would then choose an item to bid. The itemID would be sent to the BidChecker.php page. This page would process the item and insert the bid offer in the Bidoffer table in the database. This means a successful bid was placed

4.5.3.4 Notification System

The next stage of implementation was to develop a bid notification system. The idea of these notifications was to let the users know when a bid has been placed on one of their items and visa versa when a bid has been accepted/rejected the bidding user will be notified. A notification in the case of a new bid consists of the bid being offered, the auction its being offered on and the user who offered it. In the case of a bid result message, the notification consists of the auction the user bid on the owner of the auction and the result of their offer, either Accepted/Rejected. The system checks the database for new notifications every time the user reloads a page. If a new notification is found the user is prompted with a "new message" prompt. The message appears up in a fancy box as seen below.

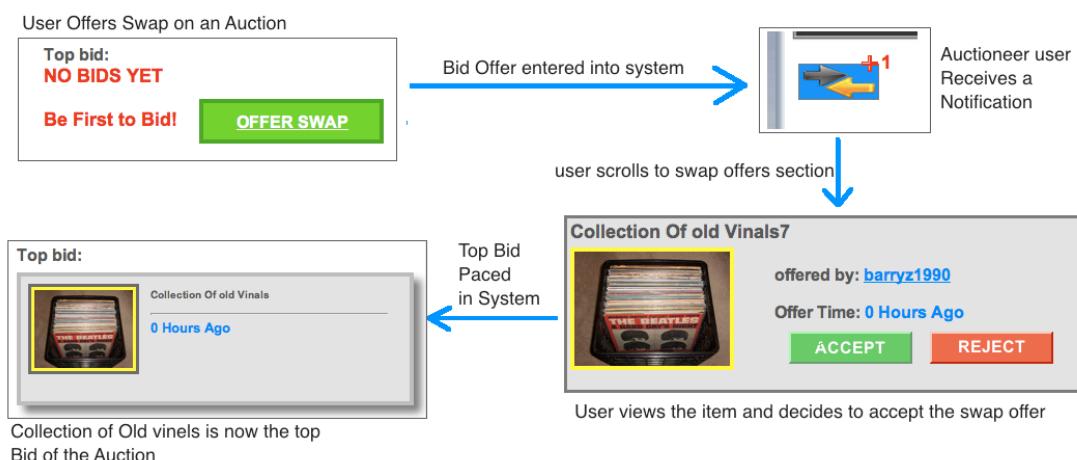


Figure 45 The bid/Notification process

4.5.3.5 Accepting/rejecting offers

After receiving notification the auctioneer user would scroll to the Swap Offers section and view their current swap offers. The layout of the swap offer can be seen above. If the user accepts a swap offer then the offer is placed into the database as the current top bid if rejected it is then the bidder receives notification of rejection.

4.5.3.6 Your auctions/ index /search and category auctions pages

These three pages all each have pretty much the same design both in ui design and code design. They each would search the auctions table within the database using a query. They would all then display the returned auctions in a 2 column table. It would then display these auctions by showing the main details about the auction. These details are the name of item up for auction, the title image of the item, the amount of time left in the auction and the current top bid of the item if any. There are slight differences in each case, which are explained.

Your auctions.php

In the "your auctions.php" page the query would find those auctions within the database that belong to the user. An additional button is added called "viewswapoffers" which will bring the user to the "viewswapoffers.php" page. This page as shown above will display any offers that have been made for

that particular auction for the user to accept/reject.

VIEW SWAP OFFERS

Index.php

The index page of the website is the default page. It is the first page the user will see when they enter "SwappaH.com" into the browser. This page finds the latest auctions that have been created by sorting the results in Ascending order (bottom to top) where the bottom rows are the newest. These auctions contain just the main details of an auction as described above.

```
$result = mysql_query("SELECT * FROM Auction WHERE Finished = 'no' ORDER BY AuctionID DESC");
```

4.5.3.7 Search results and category search

These two pages work in the same way, in that they both take in a string either from what the user has entered in the search bar or the category they have chosen in the categories section. The search results

page again uses a query to search the database for an

iPhone 4 16 GB New



Time Left: 20 days, 0 mins, 49 secs
Location: 5.3KM away

Top bid Collection Of old Vinals
A small thumbnail image of a stack of vinyl records. A yellow rectangular border highlights this image.
Bid Accepted: 10 Hours Ago

Figure 46 Individual Auction

Latest Auctions

iPod dock speakers
A photograph of a black iPod dock speaker system. A yellow rectangular border highlights the device.
Time 2 days, 16 hours, 0 mins, 57 Left: secs
Location: 5.5KM away

iPhone 4 16 GB New
A photograph of a black iPhone 4 held in a person's hand. A yellow rectangular border highlights the phone.
Time 19 days, 22 hours, 57 mins, 20 Left: secs
Location: 5.3KM away

Top bid hello1
A small thumbnail image of a device labeled "hello1". A yellow rectangular border highlights this image.
Bid Accepted: 252 Hours Ago

Top bid Collection Of old Vinals
A small thumbnail image of a stack of vinyl records. A yellow rectangular border highlights this image.
Bid Accepted: 10 Hours Ago

40 Euro Cash
A photograph of several Euro banknotes being held. A yellow rectangular border highlights the cash.
Time 11 days, 22 hours, 58 mins, 36 Left: secs
Location: 5.6KM away

Snow Patrol Tickets O2 Dublin 30-03-2012, unable to attend
A photograph of two concert tickets for Snow Patrol at the O2 Dublin. A yellow rectangular border highlights the tickets.
Time 13 days, 22 hours, 58 mins, 13 Left: secs
Location: 5.3KM away

No top bids yet!!

No top bids yet!!

Figure 47 Latest Auctions Index page

auction that has an item that matches or contains the string searched by the user. it then returns these rows which the search results displays.

The categories page also uses a query to find any auctions whose Categories attribute matched the one chosen by the user. it also displays the returned results

```
$result10 = mysql_query("select * from Item where Title like '%$search%'");
```

4.6 Implementation of Game Section

4.6.1 What is meant by Game

Instead of an auction where a user must wait for an auction to end before they can swap an item, a game is where a user can join up to be part of a collection of users who have the ability to instantly swap items with one another. A game can be made up of 10-100,000 different users and can last for up to a month. The idea is that users would start off the game by choosing one of the items from their inventory. Once the game starts they can search through all the other users items that are in the game. They would then offer to swap their item for another users item. They can offer to swap their item with as many other items as they wish. Once another user has accepted the swap offer the items become virtually swapped, meaning that the item that belonged to user1 now belongs to user2 virtually not physically and visa versa. Those two users are now free to go on and trade their new items for other users items. Once the game has ended all the users must meet and physically trade their items to whoever ended up with it.

4.6.2 Problem with Virtual Ownership

The idea that when 2 users have agreed to swap items and then become virtual owners of each others item was flawed. I discovered that virtual ownership resulted in a chain of swaps, were if even just 1 link was broken the whole chain would collapse. To explain it simply, if one particular user was involved in many many virtual swaps throughout the game and at the end of the game decided that he/she did not actually want to get rid of their item they started the game with. Then the person who ended up with his/her item wont receive anything and in turn wont want to give their item away to another user and so on. Creating an almost cascading effect through the chain breaking it completely.

4.6.3 Resolution to problem

The best way I could think of to resolve this broken chain issue was to have the two users involved in a swap meet up and physically swap their items. Once a swap is accepted the users would be able to chat to one another via the swap chat functionality. They can chat to find a time/place to meet up and swap their items. Once they have physically swapped their items they will then both have to confirm the physical swap by changing the current swap status to "COMPLETE". Until both users swap status's are Complete the users will not be able to offer new swaps with other users or accept/reject offers coming in.

4.6.4 Joining a game

A user can join a game by clicking the “join game” icon. This will then take them to the JoinGame.php page. This page uses a query to find all the games in the database that are currently running. For each of these games it will display all of its details, these are the game Name, time until its starts/how long until its over, duration of the game and Number of current players in the game. It also contains a form and a “join game” button where the user can choose an item from their inventory to start the game off with. This method of choosing an item is the same as when creating a new auction, the item chosen will be shown in the item area. Once the user has chosen an item they can click join game and the details will be sent off to the gamechecker.php page. This page will take in these details and insert them into the GameUsers table in the database. This means that the user has now joined the game and will be eligible to offer swaps and receive swap offers once the game has started.



Figure 48 Join game

4.6.5 Viewing Your current games

Now that a user can join a game they should be able to view all the games they have joined. This is carried out in the “yourgames.php” page where two queries are run on the database to find all the games in the system that the user has joined and that is still running

```
1. $result = mysql_query("SELECT * FROM gameusers WHERE username = '$username'");
2. $result1 = mysql_query("SELECT * FROM swappahgame WHERE gameid = '$gid' AND
   Finished ='n'");
```

The first query finds all the games that the user is currently in, the second query then uses the info from the first query to find out whether each of these games are finished or not. The page then takes the details of the current unfinished games and displays them one after another. There are 4 main details about the game that are displayed, they are the game name, the current

A screenshot of a game view in the "Your games" section. The game is named "Alpha" and is finishing in 14 days. It is a starting item, and there is a "Guilt ticket" associated with it. To the right, a "Current Item" is shown as a 10 euro banknote. The entire screenshot is framed by a yellow border.

Figure 49 A game View in 'Your games' section

time remaining, the starting item of the user and the current item of the user. if the user has not made any swaps yet in that particular game then both starting item and current item will be the same. The user can click on any of these games which will bring them to the game.php page.

4.6.6 Viewing a particular game

This takes places in the game.php page. this page takes in a gameId, it will then run a query to find out whether or not the current logged in user is a player within this game. If not then the game.php will just display the different items within the game and that's it. The user will have no other options.

If the user is a valid player of the game then the page will gather all of the details of the game by running another query using the gameId it will display all of the details of the game such as name, time until finished, current item, starting item and also things such as number of swaps the user has made. When the user has just started a game or has completed a swap they will see a "view swap offers" button. This will bring them to the offers page which will list all of their current swap offers for their current item. If the user is currently in an incomplete swap, in he/she has agreed to swap item with another user but has not yet physically swapped then they will see a "swapChat" button instead. Along with a small form for changing the swap status.

Below all of that there is the list of all of the current user items with in the game. There user can scroll through these items and offer to swap their item for any one of them.

Game: Alpha **Players: 7**

Current Item: Guilt ticket

Time Left: 15 days, 20 hours, 32 mins, 28 secs

Number of Swaps: 1

VIEW SWAP OFFERS (0)

Starting Item: 10 euro cash

Current Swap Status: COMPLETE

Other Swapper: COMPLETE

Your Status: COMPLETE

Game Items

- laptop**: About SwapaH.com
- sky plus remote comnrol**: Contact US Site Map
- classic Alarm clock**: Offer Swap
- Acer Aspire 5920 Used**: Offer Swap

Figure 50 Individual Game Page

Game: Alpha **Players: 7**

Current Item: Guilt ticket

Time Left: 15 days, 20 hours, 27 mins, 40 secs

Number of Swaps: 1

VIEW SWAP OFFERS (0)

Starting Item: 10 euro cash

Current Swap Status: COMPLETE

Other Swapper: COMPLETE

Your Status: COMPLETE

Figure 51 Game Details

4.6.7 How a swap occurs

4.6.7.1 Offering a swap

This is when the user clicks the “Offer Swap” button at the bottom of an item. It will cause the users inventory to pop up in a fancy box. The user would then choose an item to be swapped. These details are then sent to the swapChecker.php, which will insert the swap offer details into the swapOffer table in the database.

```
1. mysql_query("INSERT INTO swapoffer  
2. VALUES ('{$username}', '{$currentitemid}', '{$itemid}', '{$today}')");
```

4.6.7.2 Viewing swap offers

When a swap offer has been made for a users item the system will detect this and change the view swap offers button from “VIEW SWAP OFFERS (0)” were zero is the number of swap offers, to “VIEW SWAP OFFERS (1)” were 1 is the number of

swap offers. The user will be able to see this and click into it. This will bring the

VIEW SWAP OFFERS (1)

Figure 52 View Swap Offers Button

user to the swap offers page were a query is again run to find all of the swap offers for their current game item. The user can then go on to Accept/Reject these offers if the Swap is accepted then the a swap status form and Swap chat button become visible.

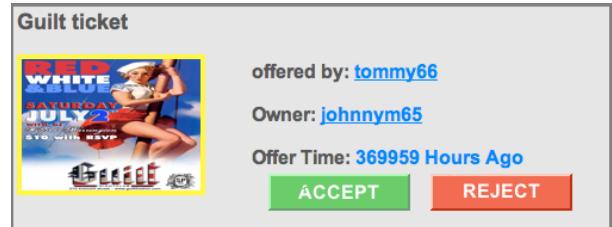


Figure 53 Swap Offer

4.6.7.3 SwapChat

Swap chat is a function of the system that allows two users who have agreed to swap items to chat to one another. this is intended to be used by the users to sort out were they both live and were they could meet to swap the items. When the “VIEW SWAP CHAT” button is clicked a pop out window will appear with the swapchat in view. The two user are linked together in the same chat. They can then type in a message and click send. This system will take the message and store it in the database. The swapchat window automatically refreshes every 10 seconds. If there are any new messages they will appear once the window is auto refreshed.

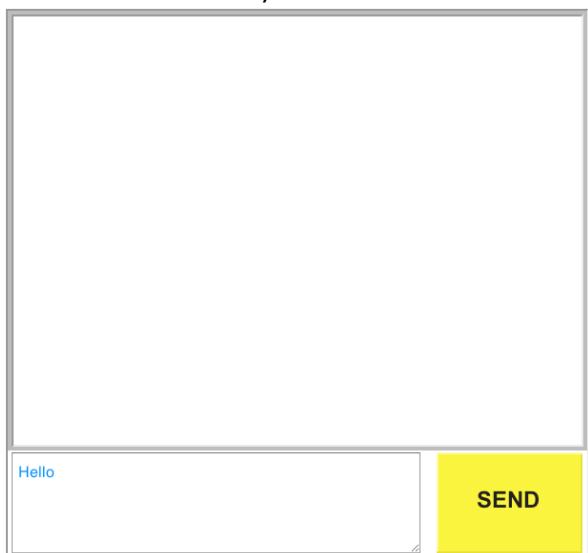


Figure 54 Swap Chat View

4.6.7.4 Swap status

Like the swapchat feature the swap status of a swap is created when a new swap has been created. It displays the current status of a swap, showing both users interpretation of the swap. Either Completed or Incomplete. Once the users have chatted and met up to physically swap their items they can then go on to change their swap status to complete. Only when both users have changed the swap status to complete can the user continue with the game.

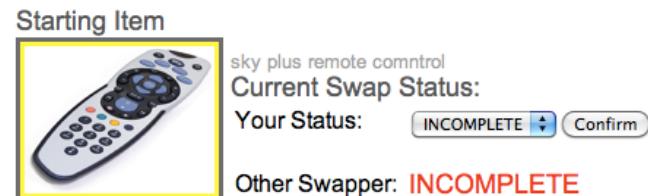


Figure 55 Swap Status

4.6.8 Number of swaps

The “number of swaps” section when clicked will open a fancy box pop up box that will contain details of all the swaps that the current logged in user has been involved in. It will show in order the different swaps that took place with the current logged in user’s item on the left and the items being swapped for on the right. It shows the chain of swaps. It does this by running a query on the swaps table in the database. It finds out all the swap the current user was involved with. It then finds which item belonged to the current item and places it on the left

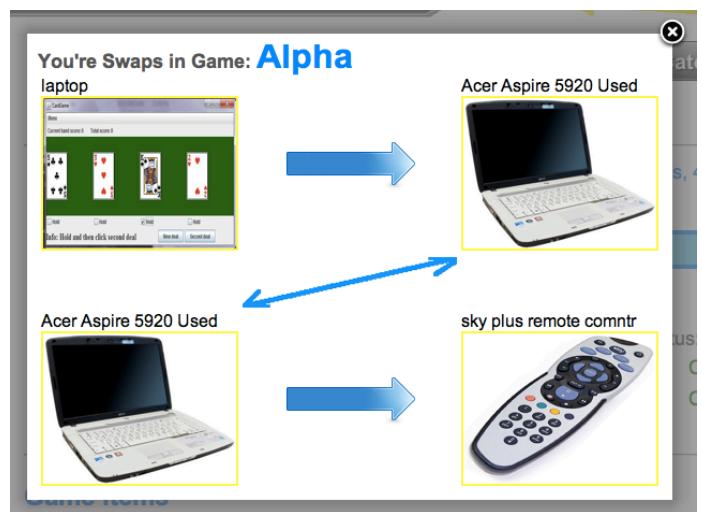


Figure 56 User Swaps in certain Game

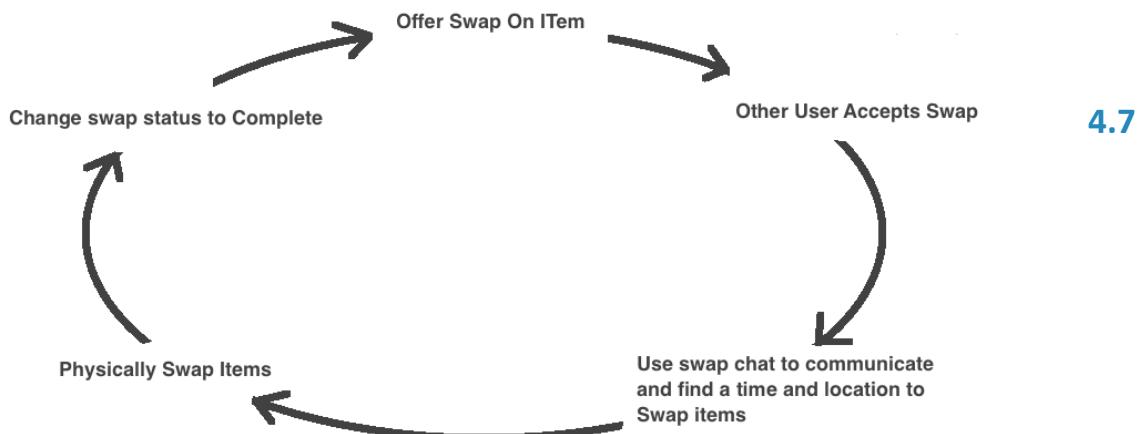


Figure 57 Game Swap Cycle

Implementation of “Swaps” section

4.7.1 Overview

Now that the implementation of inventory, Auctions and Games are complete the next stage is to develop the “Swaps” section. This section is where the user can view any of their past swaps along with current top bids. It does this in 4 sections, they are Finished Auctions, Current Top Bids, Auctions Won and Finished Games.

Finished Auctions, Auctions Won and Finihsed games all contain information of elements were the end time has been reached and no further bids/swaps can be made. This is were for instance a user won an auction can go to organize a meet up to swap their items. This is all carried out in the Tabstest.php Page

4.7.2 Tab Layout

As you can see from the User Interface design the features are laid out in a Tab setup were each of these components reside within a different tab. The user can simple scroll through each of the tabs to view its content. These tabs are created by using CSS and JavaScript. All of the tabs are hidden except for the one that is currently on top. When the user clicks on another tab then that tab will become visible while the others remain hidden. Tabstest.php is quite large as it essentially has 4 pages of content within 1 page. The page loads every piece of data for each tab when it is opened so it has smooth transition when scrolling through the tabs.

YOUR SWAPS

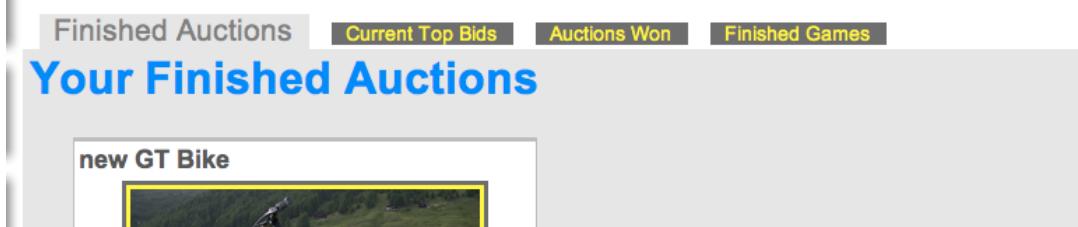


Figure 58 Tabs inside SWAP section

4.7.3 Finished Auctions

The finished Auctions tab as it says holds the information of all the auctions that the user has created and that has ended. The details it holds for each auction are the item up for auction and the wining bid of the auction. It also has features similar to the game swap features like Swap Chat and Swap Status. These allow the Auctioneer user and the winning bid user to communicate with each other. They can go on to decide a time and a place to meet up and swap their items. Once they have physically swapped their items they can change the swap status from INCOMPLETE to COMPLETE. Once both have change the swap status to complete they can then go on to give each other reviews using the review system which will be talked about later in the chapter. The auction are gathered from the database again using sql query finding all of the auctions that belong to the user and that have finished. It then displays them in a 2 column table as set out in the UI design

```
1. $result = mysql_query("SELECT * FROM Auction WHERE username = '$username' AND Finished = 'yes' ORDER BY AuctionID DESC");
```

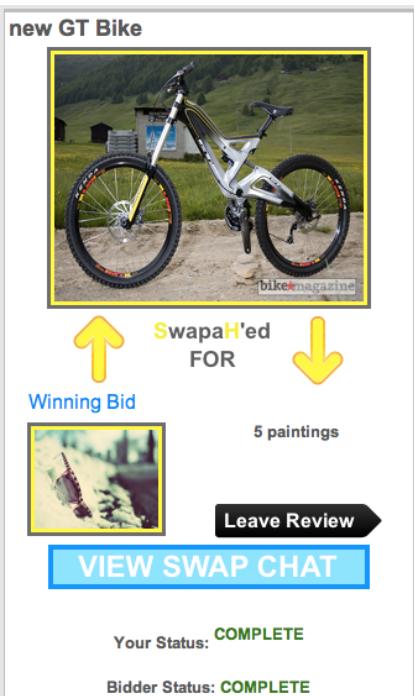


Figure 59 Finished Auction with winning bid

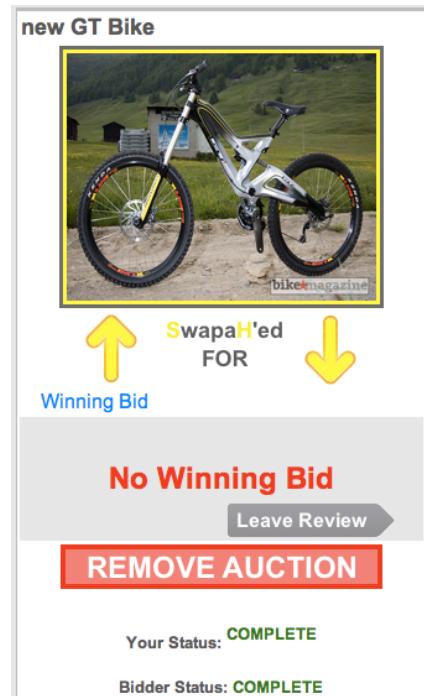


Figure 60 Finished Auction With no Winning Bid

4.7.4 Top Bids

The Top bid section gathers all of the users current top bids in any live auctions. It's a handy way for a user to keep track of how their top bids are doing and to see if they have been outbid in any auctions. It shows the details of the auction the users item is currently top bid in, also the details of the user item itself. Its gets all this info from the Bids table inside of the database. From there it can find out if the user is currently top bid in any live auctions. If so then it retrieves the details from the auction along with the details of the user item and displays them. If the user is outbid in any auction then the top bid will simply disappear from the top bid section of the page.

Figure 61 Current top bids section

4.7.5 Auctions Won

This section shows the user all of the auctions if any they have won. Winning an auction means that the auction has ended while the users bid remains top bid. Once the auction has ended both the auctioneer user and the winning bid user can see it in either the finished auction section or the auctions won section. These two sections are virtually identical. The auctioneer user can communicate with the winning bid owner using the swapchat inside the finished auctions section, while the Auction winner can communicate with the auctioneer user using the swapchat inside the Auctions Won section.

4.7.6 Finished Games

The finished Games tab shows the user the details of all of the games that they were involved in. It shows the item the user started with. The item the user finished with, the number of players, the number of swaps (which can then be expanded to show each individual swap), the final swap features. The final swap features are again the SwapChat and the Swap Status features. One a game ends the game is moved from a users Current games section into the finished game section. The system finds these games by again using two sql queries.

```
1. $result = mysql_query("SELECT * FROM gameusers WHERE username = '$username'");
2. $result1 = mysql_query("SELECT * FROM swappahgame WHERE gameid = '$gameID' AND
   Finished = 'y'");
```

4.8 Five star review/account System

4.8.1 Overview

Now that user can win and finish auction the next stage of development was to create a review system. This review system would allow users to review one another performance between 1 and 5 stars along with a short worded review. But what culminates a user performance? A users performance is determined by the reviewing user. the reviewing user could base it on anything from speed of reply, agreement of a meeting spot, politeness, quality of item swapped etc. The premise of the review is solely up to the reviewer. One a review is made it is then always associated with the reviewed user and can be seen within their account.

4.8.2 How to create a review

Before a review can be made there is a few criteria that must be met, they are that the auction must be finished, the auction must have a winner and the items must have been physically swapped and both swap status's set to COMPLETE. Once these criteria are met a review can be made. The auctioneer user can review the winning bid user and the winning bid user can review the auctioneer user. To review, the user must simply click on the "Leave Review" button that is located on the auction.



Figure 62 Leave Review Button

This will then bring up the 5 star review pop up page inside of a fancy box. What the user will do here click on the star which corresponds the star rating they wish to give the user. for example clicking the fist star mean 1st star rating, clicking the 2nd means a 2 start rating and so on.

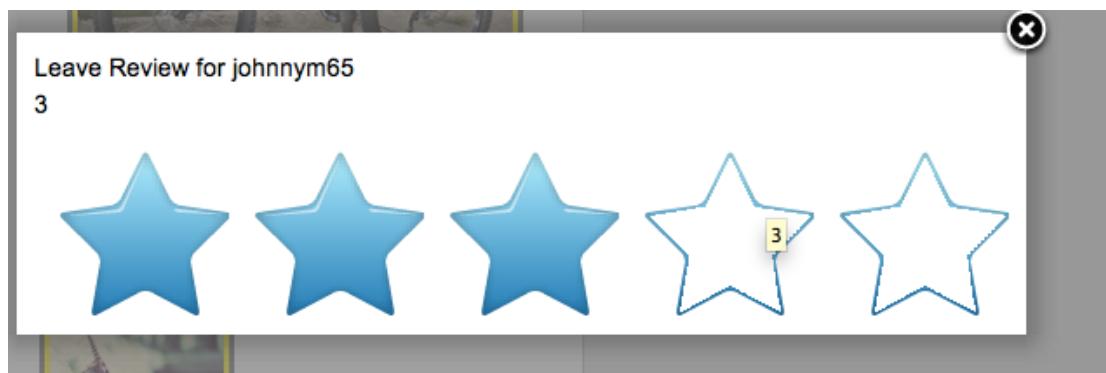


Figure 63 Five star review system

once the user has chosen their star rating they are then brought to the reviewchecker.php page. This page takes in the star number and displays it at the top of the screen. This page tells the users to leave a worded review to go along with there start review. Once the user has written the review they simple click the leave review button. This will send both the start rating and worded review to Submitreview.php which will insert the review into the database.

Review for johnnym65



Write a Review:

Very quick and easy to contact. Item was in great condition when i swapped it.
Definitely Swap with this user again!

LEAVE REVIEW

Figure 64 Written Review page

If a reviewing user was to leave more than one review for the same user the review would overwrite the original review. This is to stop people from attempting to improve their average review through foul play.

4.8.3 Accounts

Each user has an account, which will contain their average star rating along with each individual review, the average start rating will appear at the top of the screen. The average star rating is calculated by adding up every review and dividing by the number of reviews. The average is then rounded off to the nearest half star. For example if a user had 5 reviews, 1 star, 5 star, 1 star, 3 star and a 3 star. Their average review would be 2.6 stars. This would round off to a 2.5 star average review.

Star Rating



Figure 65 Average Ratings in star form

Underneath the average star rating are each individual review. They are made up of the written review and star rating.

simond111 Said:
Very Prompt and easy to contact



Figure 66 Example of Written Review with individual rating

The reason for having an account is for users to view each others history and see what other users opinions about you were like, before offering a swap or accepting a bid.

4.9 Implementing GeoCoding

4.9.1 What is Geocoding

Geocoding the process of getting the geological coordinates of a certain place such as latitude and longitude using something like a house address. This project uses the Google maps API. This allow the system to identify the latitude and longitude of a users home through their street address entered during registration.

4.9.2 Uses of geocoding in this project

The use of geocoding in this project is to find a users geographic coordinates from the address entered during registration. These geological coordinates are then used to find the distance between one user and another. Because this application requires users to physically meet up and swap items they mite prefer to swap with someone who lives within a certain radius from where they themselves live. When a user is searching through auctions or even looking at a certain auction the distance

between where they live and the auctioneer user lives is calculated using the following algorithm

Firstly the users latitude and longitude are found using their home address and the google maps api. This is calculated every time the user logs in

```
1. $geocode=file_get_contents("http://maps.google.com/maps/api/geocode/json?address=". $  
   address."&sensor=false");  
2. $output=json_decode($geocode);  
3. $lat = $output->results[0]->geometry->location->lat;  
4. $long = $output->results[0]->geometry->location->lng;
```

The latitude and longitude are stored in the session memory. Until they log out This code is again used in real time to find out the coordinates of every auctioneer user when the logged in user is searching.

Once both the logged in user and auctioneer user's latitude and longitude have been found it is then possible to use an algorithm to find the distance between them.

```
1. function getDistance(di, alat, along)  
2. {  
3.   var lat1 = latatitude;  
4.   var lat2 = alat;  
5.   var lon1 = longatitude;  
6.   var lon2 = along;  
7.  
8.   var R = 6371; // Radius of the earth in km  
9.   var dLat = (lat2-lat1).toRad(); // Javascript functions in radians  
10.  var dLon = (lon2-lon1).toRad();  
11.  var a = Math.sin(dLat/2) * Math.sin(dLat/2) + Math.cos(lat1.toRad()) * Math.cos(lat2.toRad()) * Mat  
h.sin(dLon/2) * Math.sin(dLon/2);  
12.  var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));  
13.  
14.  
15.  var distance = R * c; // Distance in km  
16.  distance = d.toFixed(1);  
17.  distance=distance+"KM away";  
18.  return distance;  
19. }
```

"This uses the '**haversine**' formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth's surface – giving an 'as-the-crow-flies' distance between the points (ignoring any hills, of course!)".

Haversine formula:
$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}_1).\cos(\text{lat}_2).\sin^2(\Delta\text{long}/2)$$
$$c = 2.\text{atan2}(\sqrt{a}, \sqrt{1-a})$$
$$d = R.c$$

where R is earth's radius (mean radius = 6,371km);
note that angles need to be in radians to pass to trig functions!
(<http://www.movable-type.co.uk/n.d.>)

This algorithm then returns the distance between these two points. The system then takes this distance

Time Left: 1 day, 14 hour
Location: 0.2KM away

Figure 67 Distance Away View

and displays it on the auction itself so user will know how far away the item up for auction is located.

4.10 Android App

4.10.1 Overview

This project includes an Android application to go along with the web application. The reason for having this application is because the website revolves around items, and each of these items can have between 1 and 5 images belonging to it. This could get quite annoying for the user to have to constantly take pictures on a phone or camera, transfer them onto the computer and then upload them directly to the site. The application will allow a user to log in using their Swapah credentials. Once logged in they can then go on choosing an item from their inventory. Once they have chosen an item they can then change/upload a new picture from their android phones gallery for that item. This makes uploading images a lot easier for users as all they have to do is take a photo with their phones camera and upload it via the app.

4.10.2 The Login Activity

The Login screen is made up of a swapah.com logo followed by two textviews for the users credentials. The page simply takes in the users username and password using the textviews and sends them to the next activity called Login.java. The login java activity will then validate these credentials by first hashing the entered password using the SHA-1 encryption algorithm. The android application itself cannot connect to the swapah database. It must connect to a php script on the server and that php script will do the connecting to the database. It sends the credentials to the php script through a request

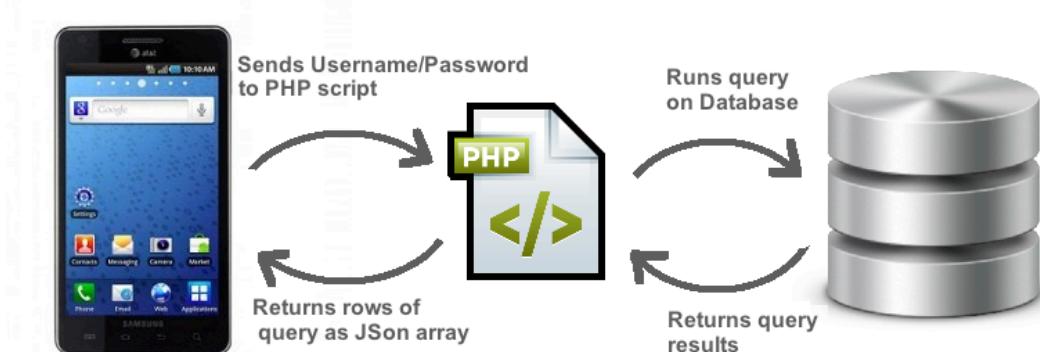


Figure 68 How Android app communicates with Database

The php script then returns the rows from the query to the application as an array. The application can then look at these results and determine whether or not the user is valid or not, if the user is a valid user then they are taken to the inventory list screen. If not they are taken back to the login screen with a message saying "invalid username/password".

In the inventory list screen the app requests another query from the server. It calls a query which returns all of the items belonging to the logged in user. It then displays the items on screen as a listView.

4.10.3 Image scroll view

When the user chooses an item from their inventory list they are brought to the showItem screen. The show item screen consist of the item title, and image of item, a change image button and left/right scroller buttons. The user can scroll through the items 5 images using the scroller buttons. If an image does not exist the default no image.jpg image will appear. The user finds the items image by using the itemID it has stored from the previous query along with the current image the user is on (image 1 - 5) . It inserts these parameters into the URL which will then look like "http://swapah.com/image/upload/" +itemID+photonumber".jpg". when the user wishes to change/upload a new image they click the "change image" button. This will bring them to the phones image gallery. The user then chooses an image from the phones memory to upload to the server. Once chosen the image is sent to the server in a post request and a php script deals with it. It either saves it as a new image or it replaces an existing image.

To log out the user just returns back to the log in screen.

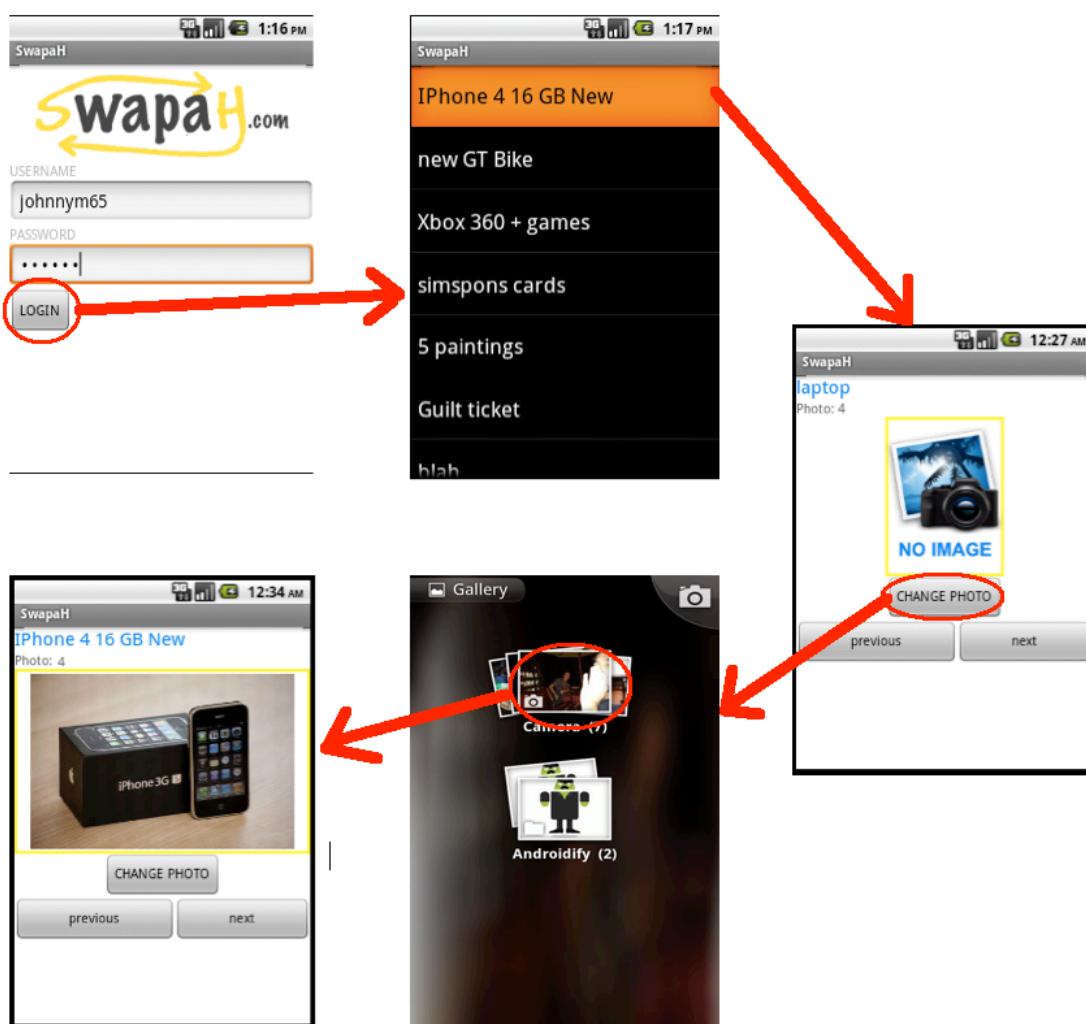


Figure 69 Android App uploading image

4.11 Deployment

Once all of the coding was finished the next stage of implementation was the deployment of the project itself. The deployment involved taking the web application from the local MAMP server in which it was developed on and transfer it to a hosted web server. This would also include Exporting and Importing the database.

4.11.1 Hostgator.com

Many different hosting sites were reviewed but eventually Hostgator.com was chosen to host the web application so it could be viewed from any computer.

4.11.2 FileZilla

FileZilla is an application that allows user to transfer large amount of files from their local servers to an online hosted server. It was a case of a simple connection and drag and drop

4.11.3 Exporting/Importing Database

MySQL allows its users to export a database along with importing a database. To do this the database belonging to this project was exported from the local MAMP server. The database was exported as a .SQL file. This .SQL could then be uploaded and imported into the new database system on the HostGator.com server. The code of the application had to be changed slightly to accommodate the new location database.

Chapter 5 Testing

5.1 Introduction

This chapter will discuss the different types of testing that were undertaken to validate the project. As this project is a web application, testing went a lot further than just the usual Black box, white box testing. This testing went through every element of the site. Each of these tests were performed in a certain environment. What that means is that when the website was being tested it was being run on Google Chrome, on a Mac osx operating system. These tests will be carried out on other browsers/operating system later down the line.

5.2 Functional testing

5.2.1 Links.

This section tested every link in every page of the web application. A link is any item on the webpage that will send the user to another page or open a new page. The main goal of this testing is to check whether or not

- A link will bring you to the page it's expected to take you
- There are any orphan pages, an orphan page is a webpage with absolutely no links within it.
- There are any broken links, Links are deemed broken if the resource that the link was pointing to has been deleted or is permanently unavailable.

Results

File names wrong because of caps. Links didn't work

Swapchat links shouldn't be there

Ask question

5.2.2 Forms

This section will test every form in the web application. A form is a way for a user to enter data/info within a web application. It uses items like textbox's, checkbox's and drop down menus. It will test each form as follows.

- Acceptance of invalid input
- The Default values of the forms
- Sending empty fields

These forms wer

- **Log in form**

when all of the test circumstances were performed on the login form the user was brought to the index page and remained logged out which is correct. When the user entered invalid credentials the same happened. Therefore the form passed

- **Register form**

When incorrect fields were entered for all the field except for town/city and email address the user was given an error with appropriate error message such as "no password entered". But when town/city was left blank the user received no error message. This should not have happened. Also the user has the option to enter info into the "address line 2" field or not. But when the user left it blank, an extra comma was added into the database, which effected the geocoding results. Another fail was that there was no format testing on the "email address field" to make sure the email was in the correct format, "something@domain.com" The Register form did not pass

- **New item form**

When blank fields were sent in the new item form, the user received an appropriate error message. The user left the YouTube link field empty along with the image fields and received no error messages which is correct. The New Item form passed the Tests

- **New Auction Form**

When the user sent empty fields again they received appropriate error messages, the default values for "Auction Length" and "Allow Questions" are "3 Days" and "no". New Auction Form passed the tests

- **Search bar form**

The search bar form passed the tests.

- **Question form**

The user sent a blank field in the "question" field of the question form. The user should of received an error message stating there must be data in the field but the user did not receive the message. The question form failed the tests

- **SwapChat form**

The SwapChat form failed the tests as it did not sanitize inputs correctly, for example if the user inputted "hello", then "hello" would print. But if the user inputted "<H1> hello </H1>". <H1> being the symbol for Heading which changes the size of the text, the text "hello" would print out large instead of the intended small.

5.3 Cookies

This will test whether the cookies working correctly. Cookies are small files sent from the server that are stored on the client's machine, for Swapah cookies are mainly used for session tracking. By enabling/disabling cookies within the browser I can test whether they are working or not.

Result

The cookies were disabled and a user log in was attempted. Because cookies were disabled the user was not able to log in as the user session could not be established. When the cookies were re-enabled the user was indeed able to log into the system. This shows that the cookies are working correctly in establishing a user session.

5.4 Database testing

This will test the web applications database, the consistency of data is extremely important for a web application. This will involve

- Testing each database query and see if it executes correctly, data received is correct and if there are any updates, that they are also correct
- Data integrity

Result

Every single database query in the application was tested to see whether or not they gave the correct results back under certain circumstances. Every query did indeed give the correct results back bar one, The "Search Auction" query that searches the database for items that contain the characters/string entered by the user in the search bar. The Query should return 0,1 or Many rows from the "Item". The Problem is that no matter what the user entered into the search bar even characters or strings such as "i", the number of rows returned was always either 0, 1. Never more than one. Even when it is known that there is more than 1 item in the Item table that contains the letter "I".

The query can be seen below

```
50 $result10 = mysql_query("select * from Item where Title like '%$search%'");
51
52 $numbers = mysql_num_rows($result10);
```

5.5 Usability testing

Usability testing involves getting users to simply use the application and to gauge their successfullness with using the application along with their own opinions of the UI. For Swapah I will have a group of people test the system. I will give them a few tasks to complete. This will show whether or not the design of the user interface is good and easy to use. Along with the tasks I will also give the test users a survey to fill out. This survey will contain questions that will get the users own opinion of the application. The task I will give the user's are as follow's

- Browse to the site <http://swapah.com>
- Register to the site
- Login with new credentials
- Add two items to their inventory
- With item1 create a new auction that will last 2 weeks and allows questions
- Search for the auction “40 euro cash”
- With item2 bid on that particular auction
- Join Swapah game ALPHA using item2
- View game ALPHA
- Wait to receive a swap offer from tester
- look at your current game swap offers for game ALPHA
- Accept swap offer
- View swapchat and say hello to tester and change current swap status to complete
- Log out

5.5.1 Survey that appeared after the Instructions

Q1. Were you able to complete all of the tasks that were laid out for you. If you were unable to complete certain tasks please list them .	Answer:
The next set of questions require an answer between 1 and 5, where 1 is strongly disagree and 5 being agree completely	
Q2. The website was easy to navigate around. The different areas of the website were easy to find.	Answer:
Q3. The look of the site was excellent	Answer:
Q4. The site was fast and efficient	Answer:
Q5. I would definitely make use of this site when it is opened to the public.	Answer:
Please use this section to list any problems that occurred during the test, also with any suggestion you would like to make about Swapah.com	
Problems/Suggestions:	

Table 2 Questions in the Survey

Results of Usability testing

- For q1, every user tested answered "yes" meaning every one of them were able to successfully complete every task that was laid out.
- The next few questions were answered using numbers between 1 and 5, where 1 was 'strongly disagree' and 5 was "Agree completely".
 - Q2 the average result was 3.3
 - Q3 the average result was 4.3
 - Q4 the average result was 3.5
 - Q5 the average result was 4.6
- The last question was reserved for the user to enter in any problems they had or also any suggestions they may have. Some of the results given back were
 - "The login screen is a bit obscured, instead of login the button says submit and there is no indication that it is the login section. The buttons although for creating actions and items on the left while a good choice need actual labels as well to clarify what they do."
 - "second item a broken pic sign came up when I try to bid in auction. Also I didn't seem to get any feedback that I had actually bid in that auction. game offer time was like 37000430"
 - "Register button is hard to see on my screen, the blue link on the dark background makes it so. Login text boxes aren't clear whats for user name and whats for password. Took a few seconds to find create auction.Took a while to notice swap game chat"

5.6 Compatibility testing

This section of testing involved checking whether the web application both looks the same and works correctly on all the different Browsers, operating systems and devices. The application was developed on a Mac OSX operating system along with chrome web browser. So the test will involve the other available systems/browsers. This test in its entirety will test

5.6.1 Browser compatibility

The Browser compatibility test consisted of 4 browsers for the web application tested on. This testing was used just to test the look of the website. All of the other forms of testing were only tested on the application using chrome browser and Mac OSX operating system.

- **Chrome**

The web application was developed using chrome as the constant testing platform. So the website looks and works correctly with this browser. All other browsers will be compared to this

- **Firefox**

Everything worked and looked correct using this browser. The log in form slightly overlapped but is a simple fix

- **Safari**

Everything looked correct in safari but there was one main fault, safari seemed to only render 1 countdown timer when there was more than 1 auction listed.

5.6.2 Operating System compatibility

The web application was used on 3 different operating system. Mac OSX which it was developed on, Windows 7 and Linux Debian. The fact that it was run on different operating systems made no difference to the look/ working of the application

5.6.3 Mobile platform compatibility

The website was run on 2 different mobile browsers. Safari on the iPhone4 and the android web browser. Most of the tests carried out on the site on a normal browser were carried out on the web browsers

- **Safari on Iphone4**

Again the issue with safari re-surfaced. The browser would only render the first of the countdown timers in a list of auctions and not the others. Another problem encountered on safari on iPhone was that it did not recognize the format in which the countdown time was in and therefore printed Days:N/A Hours:N/A Minutes:N/A Seconds:N/A as default.

- **Android web browser on android platform**

The web application worked and looked correct on the android platform. And therefore passed the test

5.7 Security Testing

This section of testing involved verifying the security of the system. Making sure the entire system is safe from potential attacks. The test checked weather

- Passing URL's directly in the address bar would allow access to the site without login authentication
- Changing individual variable values through the address bar for example
- Allowing someone with website security knowledge access to the site to review its security

Results

The result of this test were not good. The website failed as being a secure website. Alto it passed the tests for not allowing access to certain pages when the user was not validated. One the user who had a good knowledge of internet security had a go of it, many security holes were uncovered. Some of these were

- SQL Injection
 - SQL injection is where a user enters SQL statements into text fields on the site. The aim of this is to cause the system to perform operations on the database that were not intended. It is used to cause harm to the site.
- Cross scripting
 - This is similar to SQL injection in that it inputs incorrect info into the text field of websites. In this case the hacker would enter scripts into the text fields for example '<script> window.location = index.php?ownerofcard =me</script>'

5.8 Demonstration

The web application and the android application are now demo-able. For the demonstration I recorded three videos, the following 3 links provide these videos
<http://www.youtube.com/watch?v=3vF4J9yqB6Q>

http://www.youtube.com/watch?v=yw_834VCijM

<http://www.youtube.com/watch?v=6L-a5LPNtXE>

The first video shows a user 'David' register for the site. He then logs in with his new credentials. Once logged in David then adds 2 new items to his inventory, a laptop and tickets to a concert. He then goes on to create a new auction using his laptop item. The demo then goes on to show how David makes a bid on an auction for item '40 euro cash'. David then goes on to join a swapah game using his concert tickets item. Another user 'Simon' sees the concert tickets item within the game and decides to offer David a swap for his item 'DVD box set'. David then accepts the swap offer. David and Simon then chat using the swapchat and find a time/place to meet up and physically swap their items. Once physically swapped the two users change their current swap status to complete, so they can continue with the game and try make another swap. David receives a bid offer on his auction he created using his laptop item. It's a bid offer of 2 gold earrings from Simon. David views then accepts the bid. Simon is now top bid of the auction. The auction ends with Simon

as top bidder. Simon wins the auction. It then shows the two users chatting and changing their swap status's

The second video show a user Darren logging in and adding a new item 'original Gameboy'. He adds a YouTube link to the item and leaves all the image fields blank. He then goes on to view the item to ensure the YouTube link works correctly. Darren takes 5 photos of his Gameboy on his android phone. Darren then uses the swapah app on his android phone to log in, choose 'original Gameboy item', and upload the 5 images to the site. Once complete Darren logs out of the swapah app. Then verifies the images have been uploaded to the site.

The Third video demonstrates how the search bar/ categories system works, it also shows how the auction questions system works.

Chapter 6 Project Analysis

5.1 Introduction

This chapter will begin by discussing the changes that occurred between the original project plan and the project plan that was eventually taken. It will discuss how and why these changes occurred and any measures that could have been taken to avoid them if the project was to be repeated. The chapter will then go on to discuss the analysis of the technologies that were chosen in this project. Whether they were deemed the right choice or wrong choice.

5.2 Project Plan

5.2.1 Original Project Plan

December

1. Install and configure MAMP (MAC, Apache, MySQL, PHP) bundle.
2. Test network design and server connection, try and view a web page that is on the server from inside the Local Area Network.
3. Test network design and Remote Server connectivity, try and view a web page that's on the server from outside the Local Area Network.
4. Expansion on the current use case diagram, identifying every single different function of the system.
5. Re-Visit the database design, incorporating all of the newly identified functionality.
6. Start the development
7. Create some simple PHP scripts that have simple functionality such as HELLO WORLD.
8. Create a successful connection between Server and MySQL database using PHP and SQL code.

January

9. Develop a PHP application that will allow a user to register with the system. Putting their details into the database.

10. Develop a PHP application that allows a user to add/remove items to/from an inventory list that is associated to them.
11. Develop a PHP application that allows a registered user to create a new auction using one of their items. This application will allow users to bid and win these auctioned items
12. Develop PHP application for the game section of the Website.
13. Create a connection between an android phone and the server
14. Transfer data such as strings between android phone and server
15. Transfer an image file from android phone to server which will be stored within the database.

February

16. Design user interface for website around the functionality
17. Create login/logout and item list functionality within android Phone app.
18. Design UI for android app

March

19. Complete documentation, Diagrams and presentation for project. If there are any outstanding issues, resolve them.

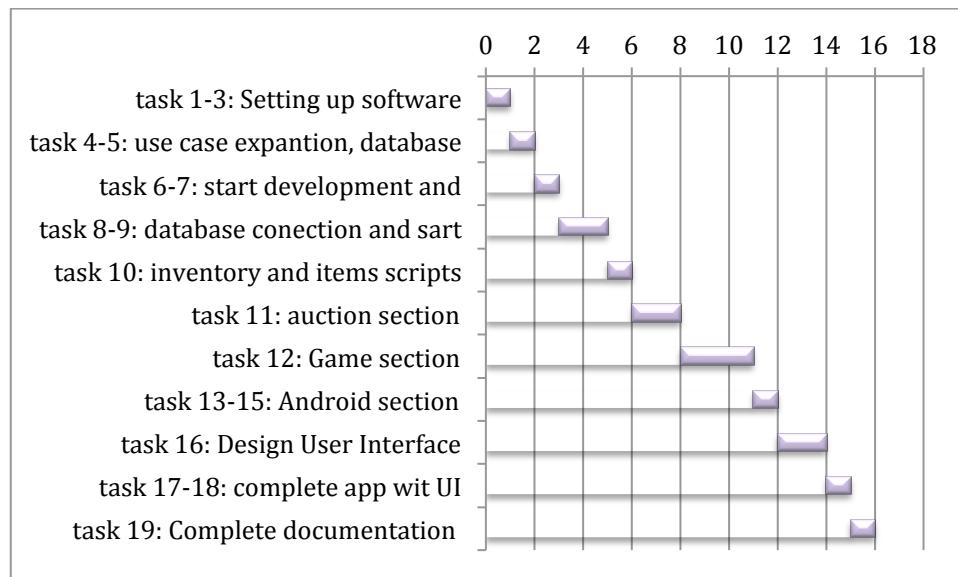


Figure 15 Gantt chart of original project Plan

5.2.2 Analysis

Looking back over the original project plan I can tell that I did not stick to it perfectly as I had planned. For one I did not start any PHP until January. One of the biggest changes that occurred was the fact that I had “design user interface” schedules to be started/finished at the end of February. This was moved to the start of January Straight after my exams and before any coding commenced. Originally I had planned to identify and then code up all of the functionality of the system first and build the User interface on top of that after. But I then came to realize that the best way to order it for this project was to actually completely design the User interface first, then from that identify each and every functionality and then begin the implementation.

In the original project plan I had planned to get most of the coding done and finished in January. This wasn’t the case instead most of the coding was done in February and even spilled over into March. This was mostly due to the fact that coding was pushed back due to the UI development being moved to the start of the plan and also that the UI development received a lot more than just 1 week of time.

Another main change from the original plan was that in the original I had not included time for testing, as testing is a very important part in the success of this project it was a huge chunk left out. The change that was made was that when the coding was completed in March then the rest of the time was split between Testing then documentation.

5.3 Tech Evaluation

5.3.1 PHP

PHP itself was an excellent choice for this project, It is versatile and easy to use. The sheer number of tutorials and libraries that help outweigh any negatives none of which I can think of.

The first thing I had to do with PHP was learn it. Which didn’t take long at all. As stated in a previous chapter, that fact if someone has some knowledge of programming or has some basic programming skills means learning PHP won’t take long. I started by coding simple scripts. Then moved on to more and more advanced scripts. I then eventually started on the project itself. PHP was a perfect fit for the server side scripting language of this web application. I could have used another language like JSP (Java Servlet Pages). But I don’t think the complexity that comes with JSP was necessary for this particular project. PHP was able to handle every bit of functionality identified for this project, nothing had to be changed or redesigned to suit PHP’s abilities.

5.3.2 Apache Web server

Apache web server was again a perfect choice for this project. Apache has a very big reputation for being a great web server and it lived up to its expectations. I didn't have to many problems while in development of the project. When turned on the server was always very quick and very smooth.

One problem that I did notice however was with that the MAMP stack I was using in. If an error occurred in the PHP code the browser would simple return a "SERVER ERROR". It would not actually return the reason for the error or what line of which file was giving the error. For example "missing ';' on line 74 of Auction.php". Other than that Apache worked great with this project.

5.3.3 MySQL

Mysql was an excellent recourse for this project. Having never used it before I was familiar with everything I needed to know very quickly. Having some experience with databases from college work and previous projects I knew a decent amount about databases. Mysql allowed to use my knowledge and create/ manage a database that supported the data for the web application. Another great feature about mysql is that I could just simple export that database from my local machine to a hosted server machine. Very quick/ very easy to use

5.3.4 TextWrangler

Text wrangler is basically notepad for mac, It's a very basic and simple text writing tool. But its also usefull for coding. It uses nice color arrangements for different types of code, PHP, HTML, JAVASCRIPT and SQL for this project. But there were a few problems with this text tool. It did not handle indenting well. If indenting was performed it would forget a lot of the time when that file is reopened. I think a better choice could have been made to use something like eclipse for the coding of this project.

Chapter 7 Conclusion

7.1 Learning Obtained

7.1.1 New Programming Languages and technologies learned

This project allowed me to learn new programming languages, along with learning new ones I also expanded any existing knowledge of other programming languages. These languages are

- PHP
- HTML
- JAVASCRIPT
- SQL
- JAVA

PHP itself was the biggest thing to learn as it was never used for projects in the past. It was learned from scratch starting off with basic hello world scripts to a lot more advanced scripts. As The rest of the languages there was already a basic knowledge for each one. This project allowed the expansion of that knowledge to a more advanced level

7.1.2 Time Management

Learning time management was a huge benefit to this project. Before it there was only small scale assignments were time management was not a major aspect of the project itself. Ive learned how key time management is to a successful project

7.1.3 UI design

This project, because is was so User interface focused showed how important the designing and creation of user interfaces are. It taught that a UI can help identify all of the functionality of an application. It also taught how important ui is for the end user.

7.1.4 Website architectures

This project has also taught how the architecture of a web application works. How each element of this architecture interacts with one another. before this project I again had only the basic knowledge of a web applications architecture from what was learned in college.

7.2 Key Strengths

7.2.1 Very unique idea

One of the core strengths that belongs to this project is that the idea of it is a very unique one. In the research chapter it discusses how after searching a similar website were users can bid on items up for auction with other item, none could be

found. Its sheer uniqueness is its biggest asset. If the Web application was to be continued after this project and released, I believe it could do well commercially

7.2.2 User interface

Another core strength of this project was the user interface design. A lot of time was spent on the designing of the user interface. It involved 3 stages of development, The low fidelity paper prototype stage where each potential page of the site was drawn out and changed if necessary. Next was the Medium fidelity where a more complete UI was designed including colouring. Finally the finished UI was coded using both the low and medium fidelity prototypes. The fact that so much time and effort went into the design of the UI shows that it should be one of the key strengths of this project, one of its core elements. I believe the UI lives up to its expectations and looks/works brilliant.

7.3 Key Weaknesses

7.3.1 Code layout

One main weakness of the project would be the layout of the code. This is mainly due to the fact that the project used a program called "Text Wrangler" which is basically the Macs version of notepad for most of the coding, when it should have made use of a better more powerful tool such as eclipse. Text Wrangler as stated before does not handle indentation well and therefore leaves the layout of the code looking terrible

7.3.2 Constant database connections

Another key weakness would be in the design of the code. As it is now in the project, every file that needs to connect to the database creates its own connection to it. This I believe can slow down the system for users on a large scale. A remedy for this would be to have created a Connection.php file that would connect to the database. This file could just then be included in the other files that need to connect to the Database.

7.3.3 bad security

The Biggest weakness of this project would have to be the Security elements of the website. I found during testing, when the project was given to a user who was very familiar with website security that my website was wide open to hacking, some of the problems mentioned were SQL injection, cross site scripting and the chat application application doesn't properly sanitize inputs.

7.4 Future Work

The future work on this project would focus mainly on getting the Web site up and ready for a real release to the public. This would mean having a fully complete polished website. To start this would involve including an email verification system. An email verification system is a system where as the user is registering to the site, They would receive an email from something like registration@swapah.com,

which would contain a confirmation link. The advantage of this is that it validates the user and the email address. It also stop users from constantly creating new accounts. The next step in future work would be to optimize the site for maximum efficiency as the side would be open to the public which would mean more and more users which means more and more Traffic on the site. Putting it under pressure.

The next step of future work would of course to solve the problem of Security. As said before the project as it stands now is wide open to hacking and if it was released to the public then it would be hacked to pieces. The website would need to be as secure as possible before being releases. I would also like to add an administration section of the website, where administrators would be able to handle user queries, and deal with error or complaints. It would also give them an area to create new games. Choosing its name, start date and end date.

Basically The future work is made up of all the things not yet completed that would enable this website to be successfully released to the general public without any problems or issues.

References

1. W.Amblter, Scott. *The Agile Unified Process*.
<http://www.amblysoft.com/unifiedprocess/agileUP.html#Figure1> (accessed Nov 26, 2011).
2. Apache Software Foundation. <http://www.apache.org/> (accessed Nov 20, 2011).
3. Butcher, Tony. *Sams teach yourself MySQL*. Sams Publishing, 2002.
4. Diffily, Shane. *The Websites Managaers Handbook*. Lulu.com, 2006.
5. Group, Isrd. *Introduction to Database Managment Systems*. Tata McGraw-Hill Education, 2005.
6. <http://www.tizag.com/>. <http://www.tizag.com/>. 2008. (accessed march 14, 2012).
7. Ltd, TestPlant. *TestPlant*. http://www.testplant.com/wp-content/uploads/downloads/2011/06/BB_vs_WB_Testing-1.pdf.
8. MacIntyre, Peter. *PHP: The Good Parts*. P'Reilly Media, Inc, 2010.
9. Micheal Kofler, David Kramer. *The definitive guide to MySQL*. Apress, 2005.
10. Microsoft. *IIS Overview*. <http://www.iis.net/overview> (accessed Nov 28, 2011).
11. Scott Mayers, Mike Lee. *Learn Maco OSX Snow Leopard*. Apress, 2009.
12. Tim Converse, Joyce Park, Clark Morgan. *PHP5 and MySQL bible*. John Wiley & Sons, 2004.