A project report

On

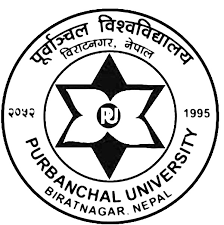
Quid Pro Quo

Submitted in partial fulfillment of the requirement of Project – IV BIT279CO

Of

Bachelor of Information Technology

**Submitted To:**



Purbanchal University

Biratnagar, Nepal

**Submitted By:**

Suyog Adhikari (343762)

Shreesha Shrestha (343760)

Uttaam Kharel (343763)

**KANTIPUR CITY COLLEGE**

Putalisadak, Kathmandu

November 23rd 2019

A project report

On

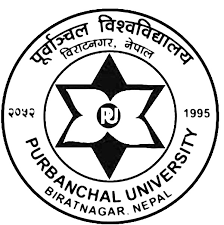
Quid Pro Quo

Submitted in partial fulfillment of the requirement of Project – IV BIT279CO

Of

Bachelor of Information Technology

**Submitted To:**



Purbanchal University

Biratnagar, Nepal

**Submitted By:**

Suyog Adhikari (343762)

Shreesha Shrestha (343760)

Uttam Kharel (343763)

**Project Supervisor**

**Mr. Kushal Niroula**

**KANTIPUR CITY COLLEGE**

Putalisadak, Kathmandu

November 23rd 2019

**TOPIC APPROVAL SHEET**

It is hereby informed that the topic selected by Suyog Adhikari (343762), Uttam Kharel (343763) and Shreesha Shrestha (343760) of BIT IV semester for their semester project has been found suitable and as per the credit assigned by Purbanchal University (PU), Biratnagar, Nepal.

The project Committee has approved the following topic for the above-mentioned students.

Topic Approved: Quid Pro Quo

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Mr. Kushal Niroula |  | Mr. Bikash Neupane |
| Project Supervisor |  | Project Coordinator |

**CERTIFICATE FROM THE SUPERVISOR**

This is to certify that the project entitled “Quid Pro Quo” submitted by Suyog Adhikari (343762), Uttam Kharel (343763) and Shreesha Shrestha (343760) to the department of Information Technology, School of Science and Technology at Kantipur City Collage, Kathmandu, Nepal towards the requirement for BIT279CO of is an original work carried out by them under my supervision and guidance.

Signature:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Kushal Niroula

Department of Information Technology

(Project Supervisor)

Place: Kantipur City Collage, Putalisadak Kathmandu

Date: November 23rd 2019

# **ACKNOWLEDGEMENT**

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. We would like to acknowledge with much appreciation the crucial role of the staff of Kantipur City Collage, who gave us the permission to use all required equipment and the necessary materials to complete the task.

Furthermore, special thanks to our project supervisor, Mr. Kushal Niroula, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report also suggesting us about the task and guiding us during the completion of this project. Finally, many thanks to lab in-charge for providing the facilities of lab during our project. We must appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills, thanks to their comment and advices.

Suyog Adhikari (343762)

Shreesha Shrestha (343760)

Uttam Kharel (343763)

# **EXECUTIVE SUMMARY**

In this technological era, online world has become a basic need. People can fulfill most of their needs via internet with very less effort. Seeing people's dependency on internet reaching this far, project on 'Quid Pro Quo' is developed to Provide goods exchanging platform to the users. Unlike other similar platforms that work as an online store with heavy monetary transactions, Quid Pro Quo brings user a new concept where one can simply exchange products which they already haveand is not needed with a usable product which has a similar price value. This project is an effort to minimize people's concept that money is required to fulfill every need, one desires.

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENT i](#_Toc25383127)

[EXECUTIVE SUMMARY ii](#_Toc25383128)

[LIST OF FIGURES iii](#_Toc25383129)

[LIST OF TABLES iv](#_Toc25383130)

[chapter 1: Introduction 1](#_Toc25383131)

[**1.1** **Project Introduction** 1](#_Toc25383132)

[**1.2 Statement of Problem** 1](#_Toc25383133)

[**1.3** **Project Objective** 1](#_Toc25383134)

[**1.4 Project Features** 1](#_Toc25383135)

[**1.6** **Organization of the Document** 3](#_Toc25383136)

[Chapter 2: Literature Review 4](#_Toc25383137)

[**2.1 Existing System Overview** 4](#_Toc25383138)

[**2.1.1**  **Steam** 4](#_Toc25383139)

[**2.1.1.1**  **Introduction** 4](#_Toc25383140)

[**2.1.1.2 Pros** 4](#_Toc25383141)

[**2.1.1.3 Cons** 4](#_Toc25383142)

[Chapter 3: System Analysis 5](#_Toc25383143)

[**3.1** **System Development Model** 5](#_Toc25383144)

[**3.2** **Requirement Specification** 5](#_Toc25383145)

[**3.2.1** **Functional Requirement** 5](#_Toc25383146)

[**3.2.2** **Non-Functional Requirement** 5](#_Toc25383147)

[**3.3** **Feasibility Study** 6](#_Toc25383148)

[**3.3.1** **Technical Feasibility** 6](#_Toc25383149)

[**3.3.2** **Economic Feasibility** 6](#_Toc25383150)

[**3.3.3** **Resource and time Feasibility** 6](#_Toc25383151)

[**3.4** **Gantt chart** 7](#_Toc25383152)

[Chapter 4: System Design 8](#_Toc25383153)

[**4.1** **Context Level DFD** 8](#_Toc25383154)

[**4.2** **Level 1 DFD** 9](#_Toc25383155)

[**4.3** **ER Diagram** 10](#_Toc25383156)

[**4.4** **Data Dictionary** 10](#_Toc25383157)

[Chapter 5: System Development and Implementation 11](#_Toc25383158)

[**5.1** **Programming Platform and Implementation** 11](#_Toc25383159)

[**5.2** **Testing and Debugging** 11](#_Toc25383160)

[Chapter 6: Conclusion and Future Enhancement 16](#_Toc25383161)

[**6.1** **Conclusion** 16](#_Toc25383162)

[**6.2** **Limitation** 16](#_Toc25383163)

[**6.3** **Future Enhancement** 16](#_Toc25383164)

[REFRENCES 17](#_Toc25383165)

# **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Figure Name** | **Page no.** |
| 1 | Gantt Chart | 7 |
| 2 | Context Level DFD | 8 |
| 3 | Level 1 DFD | 8 |
| 4 | ER Diagram | 9 |

# **LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Table name** | **Page no.** |
| 1.1 | Assignment of Roles | 2 |
| 1.2 | Organization of the Documentation | 3 |
| 4.1 | Data Dictionary | 9 |
| 5.1 | Product Display Page Testing | 10 |
| 5.2 | Login Page Testing | 11 |
| 5.3 | Signup Page Testing | 11 |
| 5.4 | Addition of Product Testing | 12 |
| 5.5 | Trade Offer Testing | 12 |
| 5.6 | Trade Offer Received Testing | 13 |
| 5.7 | Trade Offer Accepted/Declined Testing | 13 |
| 5.8 | Item Deletion Testing | 14 |
| 5.9 | Logout Testing | 14 |

**ABBREVIATION**

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form** |
| QPQ | Quid Pro Quo |
| DFD | Data Flow Diagram |
| ER | Entity Relationship |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| PHP | Hypertext Pre-Processor |

# **chapter 1: Introduction**

## **Project Introduction**

Quid Pro Quo is an online product trading interface which provides a simple and easy to use interface. QPQ being a web-based application any one from anywhere can use this application. This system mostly focuses on exchange between the different, similar and worthy products between two parties. The project is based on the concept that one should not always have a possession of money to own/buy the product they need. This system acts as a medium to make a connection between two willing parties who have items that are not significant to them but matters to others.

## **1.2 Statement of Problem**

In today’s world money has become a prominent part of human life. It works as a fuel to operate in an evolving society. Peoples need are increasing rapidly in the same way. However people even realizing the need of money, spend most of their fortune on temporarily needed products and later those items are just a space occupies in their rooms. Now this is the point our system comes. Users can upload those unused items to our platform. Included with many other similar users products are added there. Now they can exchange between those items with matching price value. This makes less monetary transactions but maximize utilization of unused products overall. The less important product to one may bring big change to other.

## **Project Objective**

Quid Pro Quo is designed to meet the following objectives:

1. To provide traders a platform for exchanging goods.

## **1.4 Project Features**

1. Provides user to user, goods exchange platform.
2. Has account creation and login system for each individual user.
3. Displayed items that can be exchanged with items that other user will upload.

**1.5 Assignment of roles**

Table 1.1 Assignment of Roles

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Project members** | **Roles and Responsibility** |
| 1 | Shreesha Shrestha | Documentation preparation, requirement gathering, interface design and testing |
| 2 | Suyog Adhikari | Front end designs, documentation, database and validation |
| 3 | Uttam Kharel | Validation, documentation and diagrams, resource gathering, literature review |

## **Organization of the Document**

Table 1.2 Organization of the Document

|  |  |  |
| --- | --- | --- |
| Chapter | Heading | Contents |
| Chapter 1 | Introductions | * 1. Project Introduction   2. Project Objective   3. Project Features   4. Assignment of roles   5. Organization of the document |
| Chapter 2 | Literature Review | 2.1 Overview of existing system |
| Chapter 3 | System Analysis | 3.1 System Development Model  3.2 Requirement Specification  3.3 Feasibility Study |
| Chapter 4 | System Design | 4.1 Context level DFD  4.2 Level 1 DFD  4.3 ER Diagram  4.4 Data Dictionary |
| Chapter 5 | System development and implementation | 5.1 Programming platform and implementation  5.2 Testing and debugging  5.3 Implementation and Result Analysis |
| Chapter 6 | Conclusion and Future Enhancement | 6.1 Conclusion  6.2 Limitation  6.3 Future Enhancements |

# **Chapter 2: Literature Review**

## **2.1 Existing System Overview**

## **2.1.1 Steam**

## **2.1.1.1 Introduction**

Steam is a video game digital distribution platform developed by Valve Corporation. It was lunched to provide automatic update for games but its services expanded as to provide a tradable feature to virtual gaming items.

## **2.1.1.2 Pros**

Out of all features provided by the system, item trading feature has also provided users. Steam provides users or especially gamers a simple yet efficient trading platform to trade virtual items. User has the option to select the virtual items from their inventory and also choose those times from friend’s inventory which they would like to receive. User then sends trade offer after making those selections. The friend then will receive the notification regarding the trade request and has the option to accept, decline or make a counter offer. The trade offers are sent as a URL using the web browser.

## **2.1.1.3 Cons**

Steam being a gaming platform, it is only focused on issued virtual faming items. These kinds of trade are entertained only for specific purpose and are not useful in day to day activities. For any new users to be able to make any trade transactions from transactions from the steam market one must have a paid transaction within 30 days of trade and not older than a year. All the items are not instantly tradeable i.e. one must possess that items in their inventory for certain amount of time.

# **Chapter 3: System Analysis**

## **System Development Model**

For this project we have used waterfall model of software development because all the requirements of the project were gathered beforehand of designing the system. Then the system is designed according to the requirements. Afterwards the implementation and testing are done after that.

## **Requirement Specification**

### **Functional Requirement**

* The user should be able to login to the website.
* The user should be able to add the desired tradable item.
* The user should be able to confirm trade.
* The user should be able to edit or delete the items they have added.

### **Non-Functional Requirement**

* Smooth transitions and proper layout must be implemented in order to make it simple yet attractive.
* The system should have good security measures for protecting credentials of registered users.
* The system should provide valid authentication.
* The system should not be heavy on browser.
* Usability: This software is usable to the traders to make the trading system convenient.
* Reliability: Since the system is password protected the system is reliable to store data. The database is maintained with XAMPP so the data is secure for further use.
* Performance: This system takes minimal amount of time to response given piece of task.
* Backup: Data and information are backed up in the hard drive of the computer. XAMPP is used to store data on the computer.

## **Feasibility Study**

### **Technical Feasibility**

Quid Pro Quo is an online trading website so it is a complete web-based application. The main technologies associated with this project are HTML, CSS, JavaScript, PHP, MySQL, and commonly used web browsers. Most of these tools are freely available and technical skills required are manageable. While in development phase website will be hosted locally and tested and later it will be hosted in a free and paid web hosting service gradually. Bandwidth required in this application is high, since it has many multimedia aspects like images, videos and audios. Thus, it’s clear that it is technically feasible.

### **Economic Feasibility**

Being a web-based application; Quid Pro Quo will have some hosting cost. Since the system consists of many multimedia contents and data transfer, bandwidth required for the operation will be high. This system will follow freeware software standards. Bugs fixes and maintaining cost will have high cost too. It will also allow for advertisement to gain revenue. Also, users in another hand using this platform should not in any other way pay any amount of money during their trade. Hence, it is clear that system is economically feasible.

### **Resource and time Feasibility**

Resources required for Quid Pro Quo project includes: programming device (laptop), programming tools (freely available), and programming individuals. Thus, it has the required resource and time feasibility.

## **Gantt chart**

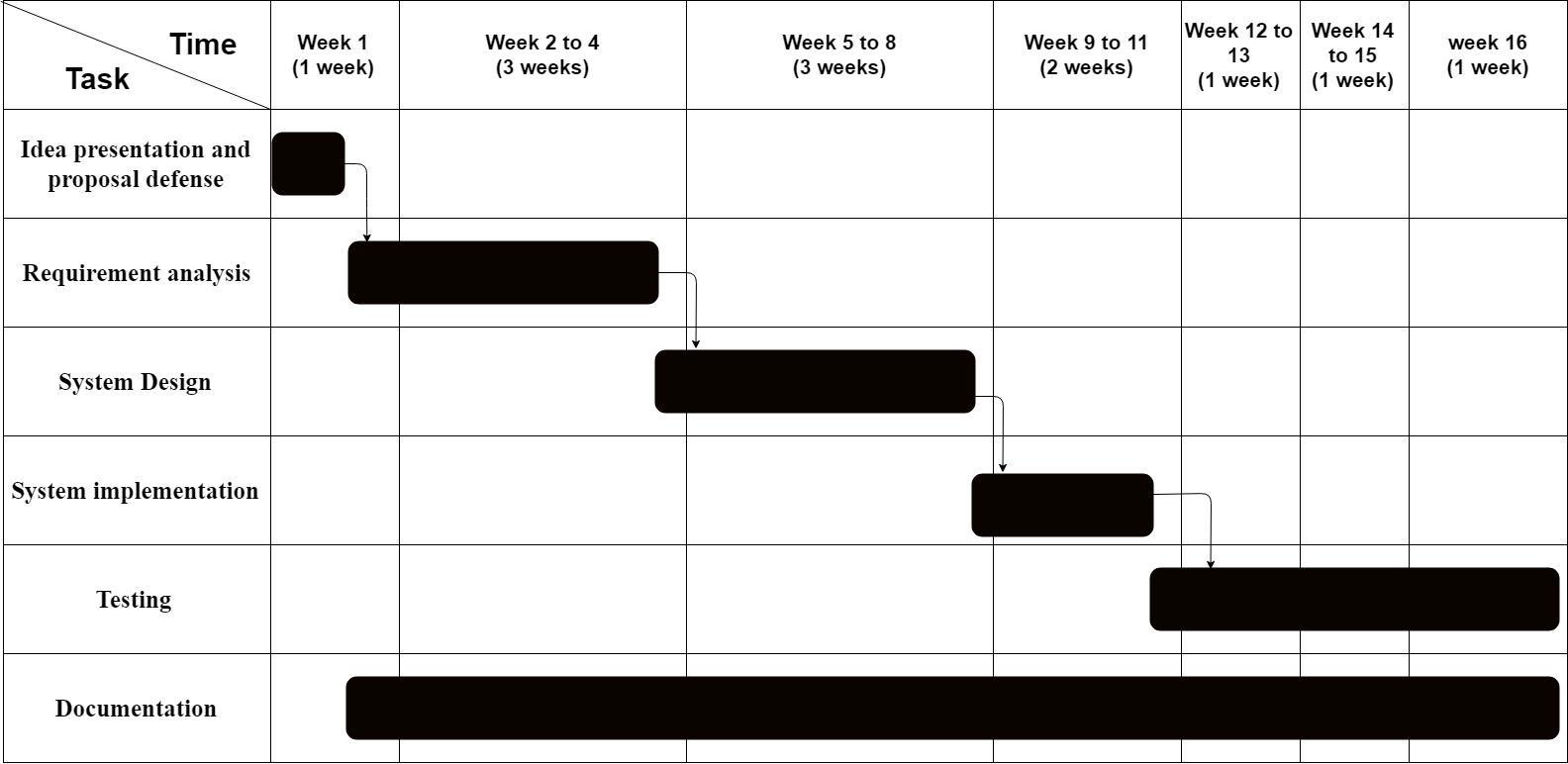


Figure 1. Gantt chart

# **Chapter 4: System Design**

## **Context Level DFD**

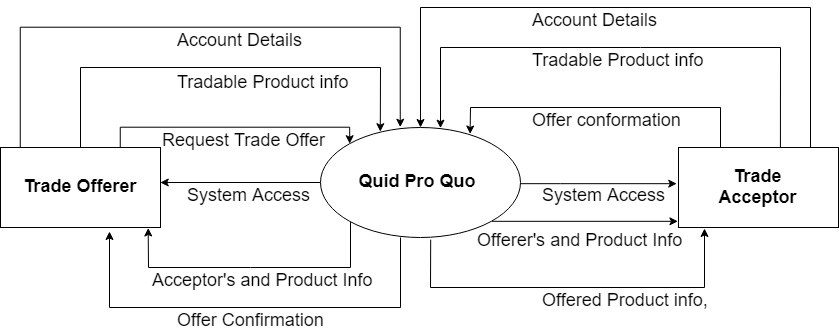


Figure 2. Context Level DFD

## **Level 1 DFD**

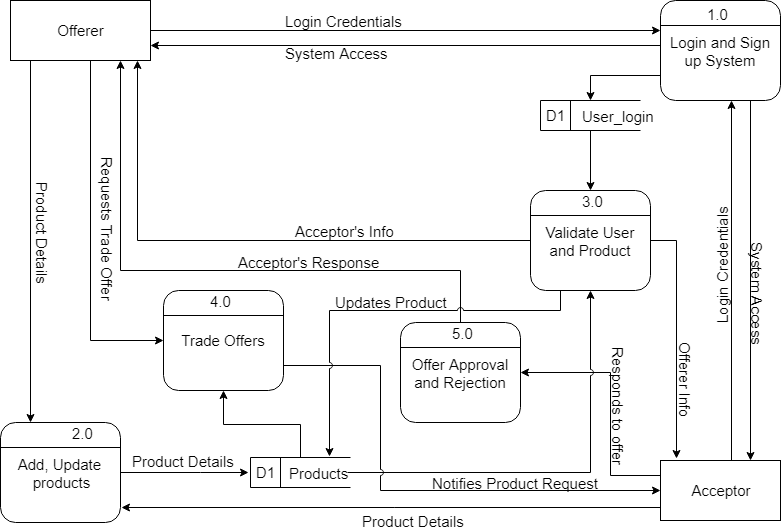


Figure 3. Level 1 DFD

## **ER Diagram**

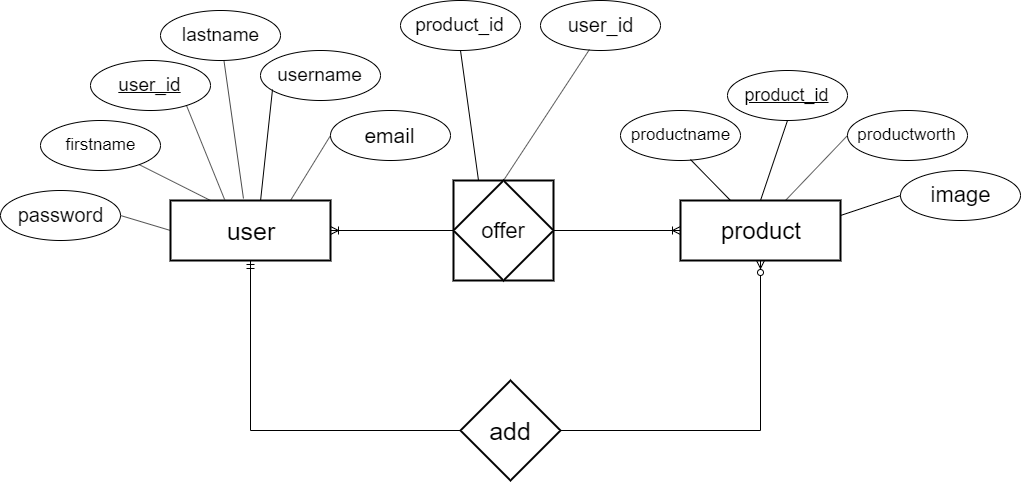


Figure 4. ER Diagram

## **Data Dictionary**

Table 4.1 Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.N** | **Table name** | **Attribute** | **Type** | **Null** |
| 1 | user | username  user\_id  email  password  firstname  lastname | varchar(50)  bigint  varchar(50)  varchar(50)  varchar(20)  varchar(20) | notnull  PK notnull  notnull  notnull  notnull  notnull |
| 2 | product | product\_id  productname  productworth  image | bigint  varchar(50)  bigint  varchar(100) | PK notnull  notnull  notnull  notnull |
| 3 | offer | product\_id  user\_id | bigint  bigint | FK notnull  FK notnull |

# **Chapter 5: System Development and Implementation**

## **Programming Platform and Implementation**

a. Front-End WEB DEVELOPMENT

* Mark-up Language-HTML
* Style Sheet-CSS/ Bootstrap classes
* Dynamic Programming Language-JavaScript
* Font awesome icons/ Fonts

b. Back-end web development

* Scripting language –PHP
* Database-MySQL

c. Development tools: XAMPP, Visual Studio Code

d. Interface: Web application

## **Testing and Debugging**

Table 5.1 Product Display Testing

|  |  |
| --- | --- |
| Test case | Product display page |
| Test objective | To check whether the products are displayed |
| Test data | Running the program |
| Expected result | Products should be display properly |
| Test result | Main screen appears correctly |
| Conclusion | Expected result matches actual result |

Table 5.2 Login Page Testing

|  |  |
| --- | --- |
| Test case | Login |
| Test objective | To check whether login is successful |
| Test data | Entering username/email and password |
| Expected result | User should be logged into the system |
| Test result | User is logged in to the system |
| Conclusion | Expected result matches actual result |

|  |  |
| --- | --- |
| Test case | Signup |
| Test objective | To check whether signup is successful |
| Test data | Entering First Name, Last Name, Username, Email, Password, Confirm Password, Phone Number |
| Expected result | User should be logged into the system |
| Test result | User is logged in to the system |
| Conclusion | Expected result matches actual result |

Table 5.3 Signup Page Testing

Table 5.4 Addition of Product Testing

|  |  |
| --- | --- |
| Test case | Adding Product |
| Test objective | To check whether signup is successful |
| Test data | Display product addition form |
| Expected result | User should be able to add new products |
| Test result | User is able to add new products |
| Conclusion | Expected result matches actual result |

Table 5.5 Trade Offer Testing

|  |  |
| --- | --- |
| Test case | Offer trade request |
| Test objective | To check whether user is able to send trade request to another user or not |
| Test data | Offered a sample product from user1 to user2 |
| Expected result | User1 should be notified about the trade offer success |
| Test result | User1 is notified about the trade offer success |
| Conclusion | Expected result matches actual result |

Table 5.6 Trade Offer Received Testing

|  |  |
| --- | --- |
| Test case | Trade offer received |
| Test objective | To receive trade offer sent from user1 to user2 |
| Test data | Sample product received from user1 |
| Expected result | Sample product should be received to user2 |
| Test result | Offered sample product is shown in user2’s offer request |
| Conclusion | Expected result matches actual result |

Table 5.7 Trade Offer Accepted/Declined Testing

|  |  |
| --- | --- |
| Test case | Trade offer accepted |
| Test objective | To check whether the offered product can be accepted or declined |
| Test data | User2 accepts and declines trade offer simultaneously |
| Expected result | The trade offer should be acceptable or declinable |
| Test result | Trade offer can be acceptable or declinable |
| Conclusion | Expected result matches actual result |

Table 5.9 Logout Testing

|  |  |
| --- | --- |
| Test case | Logout from the system |
| Test objective | Trader can logout from the system |
| Test data | - |
| Expected result | User should be able to logout from the system |
| Test result | User is able to logout |
| Conclusion | Expected result matches actual result |

# **Chapter 6: Conclusion and Future Enhancement**

## **Conclusion**

Understanding the market in the process of requirement gathering, there are interfaces that are based on heavy monetary transaction. Considering that Quid Pro Quo is developed to provide goods exchanging platform to the users where one can simply exchange products which they already have and is not needed with a usable product which has a similar price value. This system avoids the monetary transaction and provides a trading interface.

## **Limitation**

* User can get confused with his own product.
* User do not have option to choose multiple products to make a trade offer.
* Products are not categorized.
* Scams cannot be prevented.

## **Future Enhancement**

* Separate products in accordance to the category.
* Communication medium between traders.
* User can view the worn quality of a product.
* Complete database entries and records management.

# **REFRENCES**

* Bayross, I. (2010). *Web enabled commercial applications development using ... Html, Dhtml, Javascript, Perl Cgi* (4th ed.). New Delhi: BPB publication.
* HTML Tutorial. (n.d.). Retrieved from https://www.tutorialspoint.com/html/index.htm.
* CSS Tutorial. (n.d.). Retrieved from https://www.tutorialspoint.com/css/index.htm.
* (n.d.). Retrieved from https://www.w3schools.com/sql/.
* (n.d.). Retrieved from https://origin.fontawesome.com/icons?d=gal