

PURCHASE ORDER SYSTEM

by

TEJASWI BATTULA

B.Tech., Jawaharlal Nehru Technological University, 2009

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2011

Approved by:

Major Professor
Dr. Mitchell Neilsen

Abstract

Maintaining paper bills is always a tedious job and there is always a chance of missing the purchase orders or even the payment dates. But, using an online purchase order system makes it easier to enter and maintain the correct information. The main objective of this application is to keep track of all the purchase orders made by the faculty members or staff members of the department for their students or for their research work.

All the purchase orders can be made just by one click by using the online purchase order system. The user needs to register with the website letting the admin know if he is an authorized user of the department. Once the admin gives the required permission the user can then create a new purchase order from the desired vendor registered with the department. He can also make the payment by providing the funding source through which the order will be paid. Then the orders are finalized and provided with an invoice in a pdf format which can also be printed.

Additionally, there exists an Admin user who can manage/ view users and edit their PO's according to the information provided by the user. The admin also has the permissions to add new vendors and manage them.

This website is developed using the PHP Scripting Language which is one of the major technology used now-a-days to build various websites along with HTML, jQuery, CSS web technologies for a better design of the website. Major emphasis of this application is to build user interactive techniques for simplifying user needs and to provide specific products required by the user.

Table of Contents

List of Figures	v
Acknowledgements	vi
1. INTRODUCTION	1
1.1 Motivation.....	1
1.2 Objective	1
1.3 Scope.....	2
1.4 Software Requirements	2
1.5 Hardware Requirements	2
2. SYSTEM REQUIREMENT ANALYSIS	3
2.1 Requirements Gathering	3
2.2 Economic Feasibility	3
2.3 Technical Feasibility	3
2.4 Behavioral Feasibility	3
3. TECHNICAL DETAILS	4
3.1 PHP (Hypertext Preprocessor).....	4
3.2 MySQL Database Server	4
4. SYSTEM ANALYSIS.....	6
4.1 Use Case Diagrams	6
4.2 ER Diagram	9
5. IMPLEMENTATION.....	12
5.1 Database Design and Implementation	12
5.2 User Interface Design and implementation	13
5.3 Technical Discussion	31
6. TESTING.....	32
6.1 White Box Testing.....	32
6.2 Unit Testing	32
6.3 Non Functional Testing	33
6.4 Performance Testing.....	33
7. RESULTS & CHALLENGES.....	36

7.1 Challenges.....	36
8. CONCLUSIONS	37
8.1 Limitations	37
8.2 Future Work.....	37
9. REFERENCES	38

List of Figures

Figure 1: Use Case Diagram for User.....	7
Figure 2: Use Case Diagram for Admin	8
Figure 3: Entity Relationship Diagram	9
Figure 4: Database Implementation	12

Acknowledgements

I would like to thank my major professor Dr. Mitchell Neilsen for his constant guidance, encouragement and support throughout the project.

I would also like to thank Dr. Daniel Andresen and Dr. GurdipSingh for serving on my committee and for their valuable cooperation during the project.

Finally, I wish to thank my family and friends for all their support and encouragement.

1. INTRODUCTION

Purchase Order System provides an effective, interactive and secure solution for managing the purchase orders made in a particular department or organization. Now the department need not keep track of all the paper bills. This system saves valuable time and money by keeping track of all the purchase orders made by a particular Faculty Member or Staff Member. The website provides a user-friendly interface to encourage the customers to use the website. Thanks to Web 2.0 for providing all the incentives and solve this application successfully.

Web services are used to communicate between purchase order system and the vendors. Security is also implemented in the web services so that the data is transmitted securely.

1.1 Motivation

To implement the online purchase order system lot of technologies must be used. In order to differentiate this system from others in the market, the website must be built effectively. Designing such kind of system gives more scope to learn various new technologies and new concepts. While developing such systems the software development cycle can also be considered and put into practice. By developing such kind of websites, it helps to learn how to use internet technologies along with the traditional ideas to choose the right option. This application is developed using PHP, a server-side programming language along with HTML.

1.2 Objective

- The main objective of the project is to build an effective and secure web site.
- Design patterns must be used to simplify the maintenance of the application.
- To efficiently manage the customers, vendors and company purchase order database records.
- Facilitate to track purchase and sales order number and payment details through a user-friendly interface.
- Efficient navigability throughout the application so that they can reach a destination page without much traversal within the website.
- Password protected utility with advance data backup and restore policy.

- Providing a good user interface to the customer to encourage using the web site.

1.3 Scope

Currently the PO's can be viewed in PDF format. But new features can be embedded such as Vendor Reporting, Quickly viewing user's purchase orders and so on. The vendor reporting helps in keeping track of all the vendor information and also to maintain a track of all the purchases made from a single vendor. For now this system has a simple form of managing and viewing the PO's, while verification of Purchase orders can be made on the user side rather than on the admin side which would add to the efficiency of the system.

1.4 Software Requirements

This project can be supported on any operating system. But requires PHP Server installed on the system to deploy. MySQL database server is used to store and maintain the records and data.

1.5 Hardware Requirements

A processor of Pentium IV and above with a minimum 250 MB RAM and space requirements of about 100 MB are required.

2. SYSTEM REQUIREMENT ANALYSIS

2.1 Requirements Gathering

The important and crucial part of the application development is the requirement gathering phase. Information was collected from various existing purchase order systems to learn the advantages and disadvantages of those sites and to come up with the features that were developed in this application to handle some of the issues that came up during the research of such online websites. Thus Web 2.0 can provide the necessary framework and object oriented design in solving the problem efficiently.

A feasible solution includes multiple factors for an organization and here I present that this application is feasible economically, technically and behaviorally.

2.2 Economic Feasibility

This application is economically feasible because the only requirement for using this application is a minimum modern day computer with an internet connection at a good speed with web browser such as IE5 or higher, Mozilla Firefox, Chrome or Safari. To deploy this application all we need is a computer with above mentioned hardware and software specifications.

2.3 Technical Feasibility

To deploy this application the technical details would be

- Operating system [Windows 2000 and later or Linux]
- PHP installed on the server.
- MySQL Database Server.

Also internet browser along with internet connection would add to the technical details for using this application.

2.4 Behavioral Feasibility

Most of the features that are developed in this application are self-explanatory hence no special technical expertise is required in order to use this application. Users are directed through the usage of tool-tips and user friendly messages.

3. TECHNICAL DETAILS

Modern technology enables us to solve many problems in the present scenario; this application depends on PHP and MySQL server.

3.1 PHP (Hypertext Preprocessor)

PHP stands for **PHP: Hypertext Preprocessor**. Hypertext refers to files linked together using hyperlinks, such as HTML (HyperText Markup Language) files. Preprocessing is executing instructions that modify the output.

PHP is probably the most popular web scripting language used today. PHP is popular because it's easy to learn and use - generally PHP makes life more exciting. PHP is a powerful tool for making dynamic and interactive Web pages.

PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. PHP also has a command-line interface capability and can be used in standalone graphical applications. PHP can be deployed on most web servers and as a standalone interpreter, on almost every operating system and platform free of charge. PHP is a competitor to the present Microsoft Active Server Pages [ASP] Server side scripting languages.

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on web sites or elsewhere. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

3.2 MySQL Database Server

MySQL is a relational database management system that runs as a server providing multi-user access to a number of databases. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in

a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, MyBB, phpBB, Drupal and other software built on the LAMP [Linux, Apache, MySQL, PHP] software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google and Facebook.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

4. SYSTEM ANALYSIS

Based on the above research carried out in analyzing this application, various entities have been identified and the relationships between these entities have been demonstrated using an

E-R Diagram, Use-Case diagrams, state transition diagrams and data flow diagrams.

4.1 Use Case Diagrams

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence. UML diagram contains graphical elements (symbols) - UML nodes connected with edges (also known as paths or flows) - that represent elements in the UML model of the designed system. The UML model of the system might also contain other documentation such as use cases written as templated texts. The Use Case Diagram describes the system functionality from an external observer's view. It shows who can do what. The following are the use case diagrams for my application.

The user can perform various activities like Register or setup an account with the website, login, update the profile, browse the list of vendors, create new purchase orders, edit, lock, finalize PO's, purchase orders history and logout.

In this application two major actors are identified user and administrator.

User is the one who tends to use the application to make a purchase order. The user can register into the system once he accepts the confirmation code sent to him through the email. Secondly he can login into the system, create a purchase order of his own choice from a desired vendor. Along with the above functionalities he can manage the PO by editing, locking, adding items, making a payment, view in pdf format and logout. The below use case diagram shows the various functionalities that can be performed by the user.

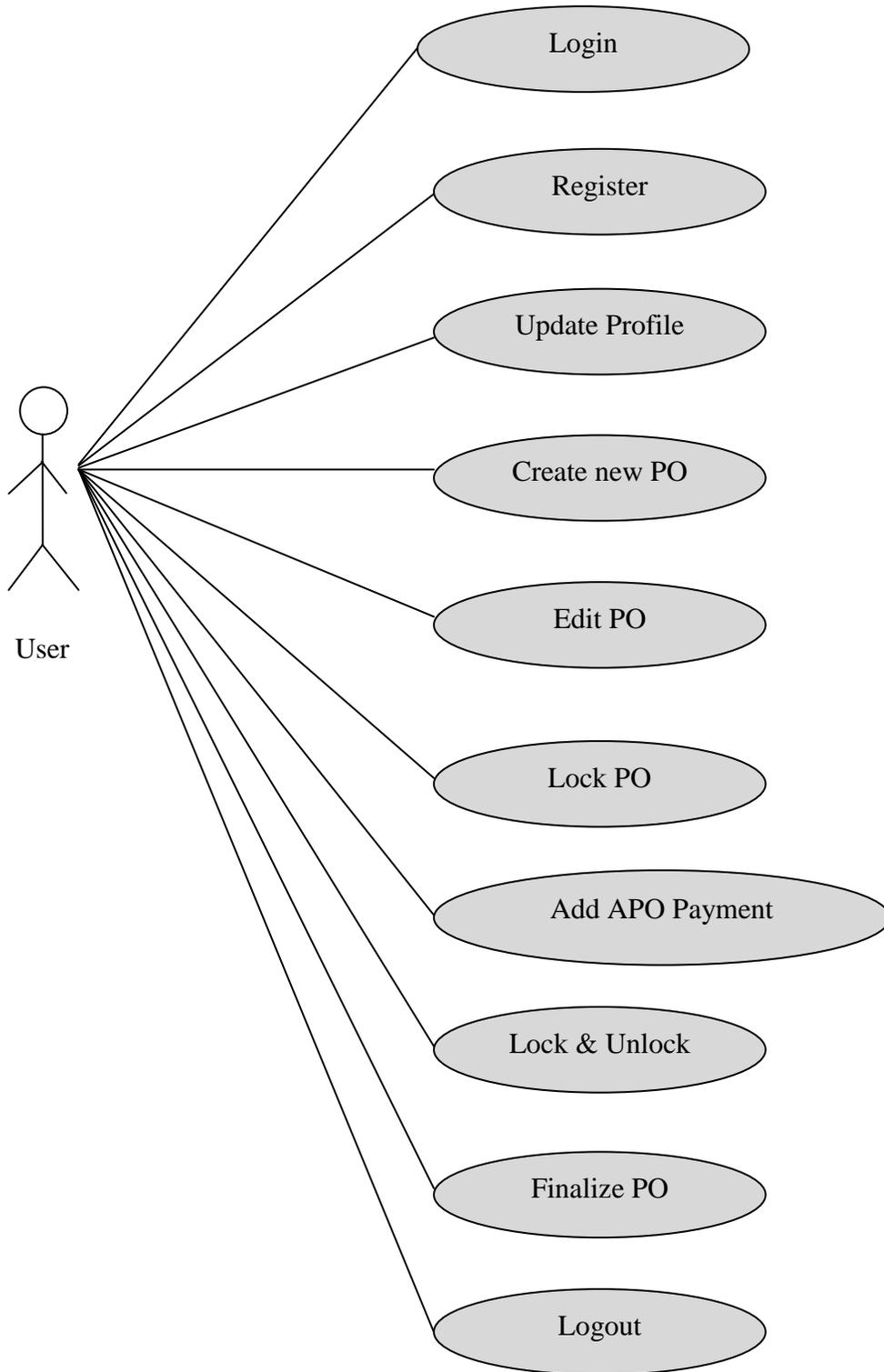


Figure 1: Use Case Diagram for User

Along with the above user functionalities the admin can perform various other actions. The admin has full access all through the website. He can manage all users, vendors in the database. He has the permission to add new vendor to the database. Various other actions that can be performed by admin are creating new PO, view all PO's of the present fiscal year and processing customer payments.

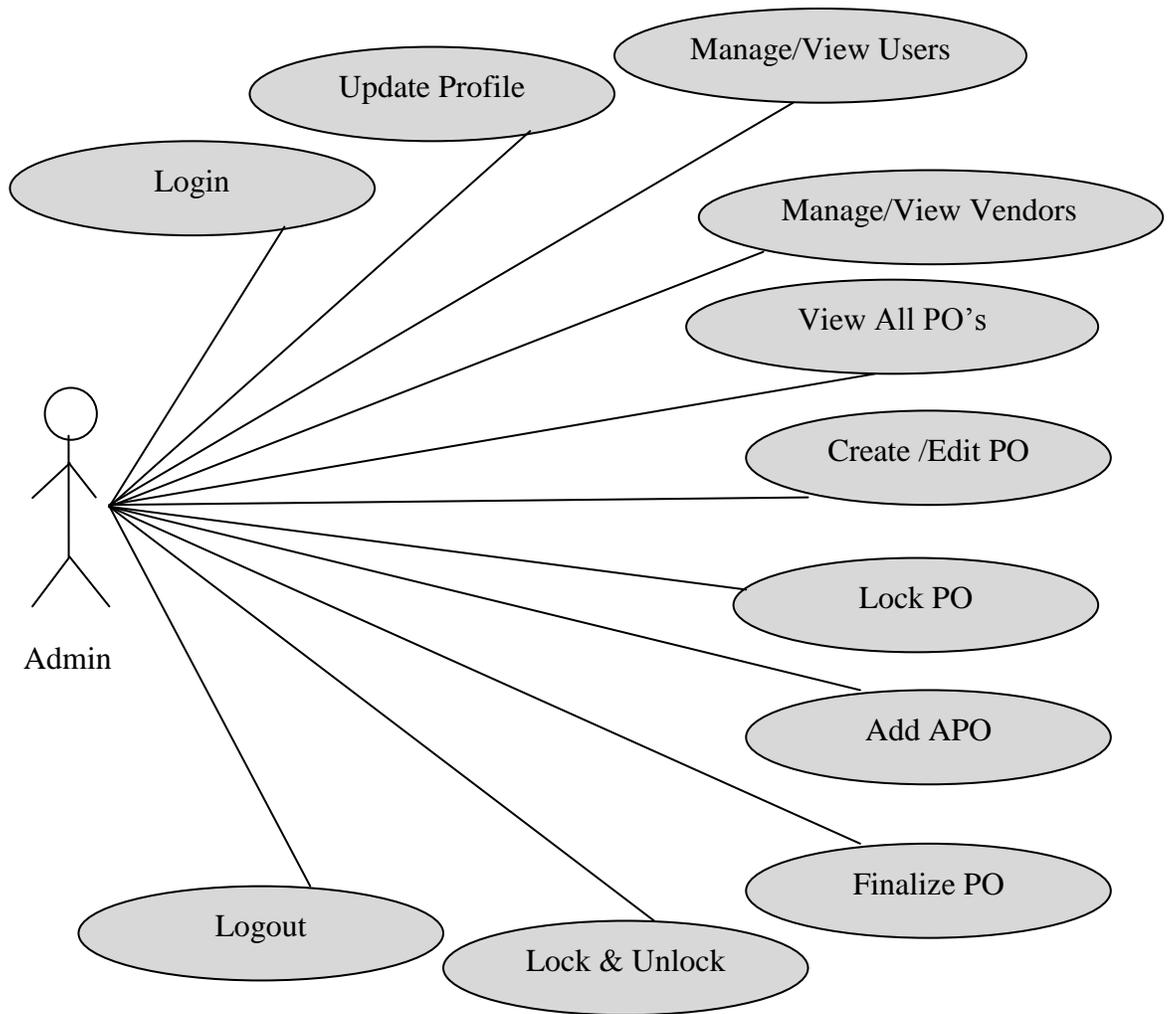


Figure 2: Use Case Diagram for Admin

4.2 ER Diagram

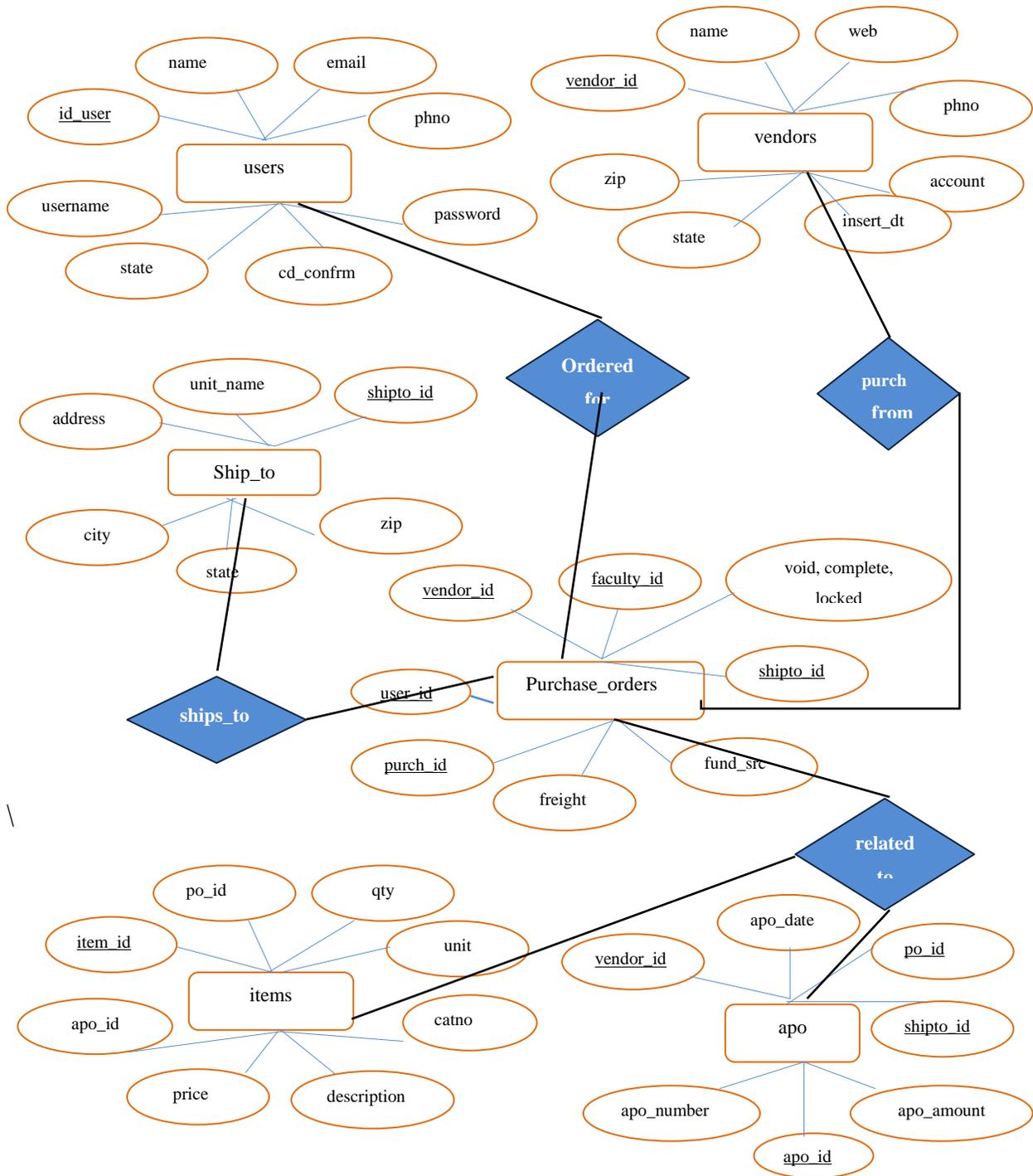


Figure 3: Entity Relationship Diagram

Understanding the Diagram

The major entities of this application are users, vendors, purchase_orders, items, apo and ship_to.

Users Table: This table holds the information of all the users who are registered with the website using the registration page in the application. All user information such as registered username, password along with Address information is stored in this table and every registered user is given a unique id called as id_user which will be used throughout the application to recognize the logged in user and to track his activity in the application. All the usernames and email id's in the table are unique.

Vendors Table: This table holds the information about the vendors that are registered by the administrator of the application. The vendor information such as name, address, phone number, email and all required information. Every Vendor is given a unique id called as Vendor_Id which is used throughout the application to identify the particular vendor.

Purchase_Orders Table: This table is used to store the information about all the purchase orders made by the users in the database. Each purchase order has its unique identifier with other attributes such as its state contract, price, freight and status of the product which means if the product is locked, finalized or voided by the user. This table has reference to the vendor_id, user-id, shipto_id which help to recognize by whom the order is made, from whom it is made and where it should be shipped. Various relations like order from, purchase form, ship to depicts the relation how the purchase order table is related to other table and gets the data from other tables in the database.

Items Table: This table has all the information about the items related to a particular purchase order such as price, quantity, category number, description, and related apo_id which determines the payment for the items. Each item is uniquely identified with its respective id.

Apo Table: This table holds the information about the payment details made for a particular purchase order with respect to its vendor_id. This table stores information such as date when the payment was made, payment method, apo_number and apo_amount.

Ship_to Table: This table holds different types of shipping options that are available to the user. Each method is uniquely identified and has attributes such as country where they can ship to, its pricing details.

Entities and their attributes along with relationships among them were carefully designed such that there is no scope for any redundancy in the data. Tables were normalized such that no duplicate data exists in the database thus making database retrievals, inserts and updates much simpler.

5. IMPLEMENTATION

5.1 Database Design and Implementation

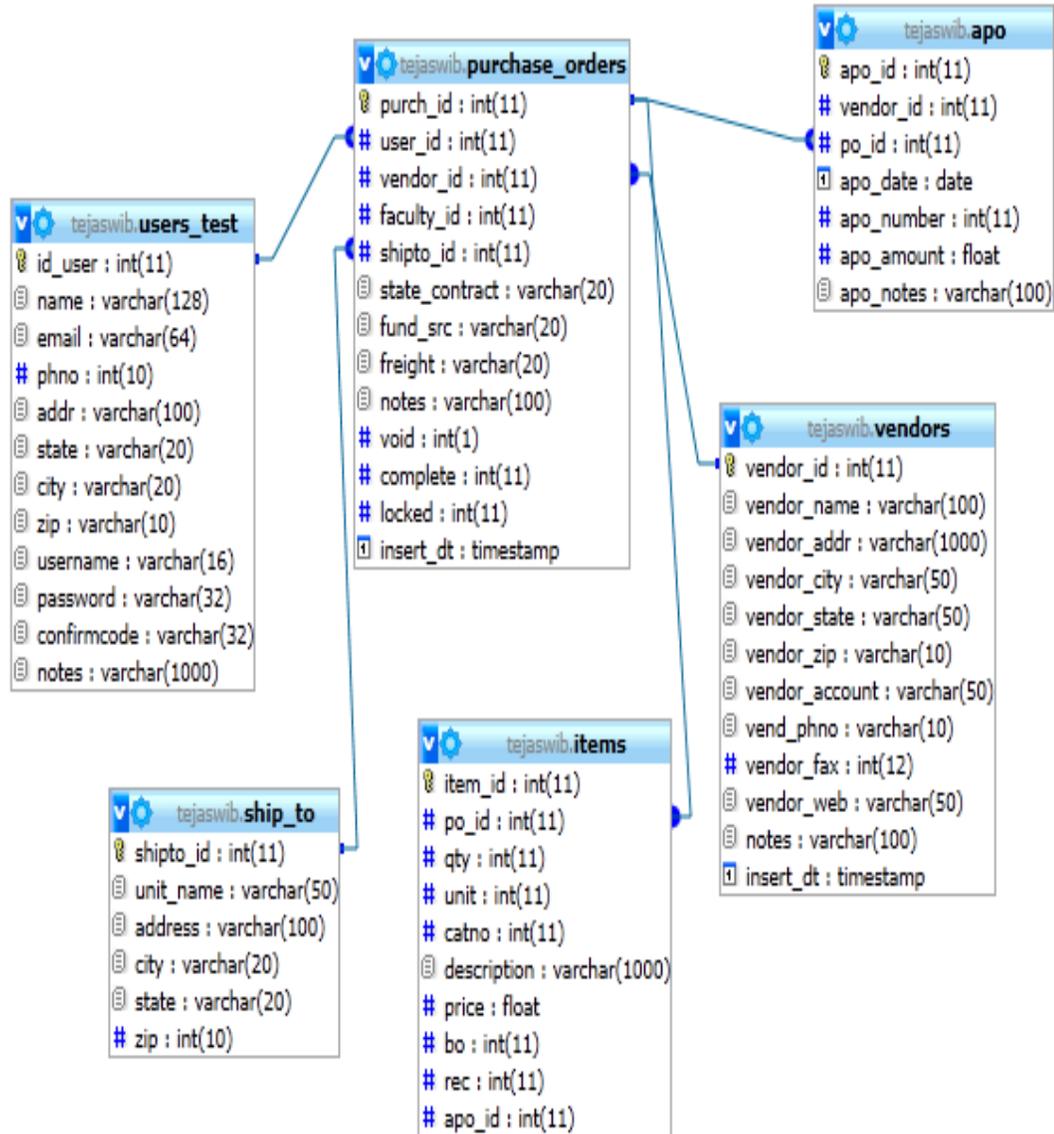


Figure 4: Database Implementation

Above diagram represents the schema of the application along with design of database along with foreign key and primary key relations.

5.2 User Interface Design and implementation

User interface has been designed using PHP, HTML, CSS, jQuery. Firstly the screen shots of the User view are shown below and then the Administrator view are depicted below.

The screenshot shows a web browser window with the title "Register Page" and the URL "http://people.cis.ksu.edu/~tejaswib/source/register.php". The page content includes a shopping cart icon, a "Purchase Order System" header, a "Register Here" section, and a registration form with the following fields and labels:

- * required fields
- Your Full Name*:
- Email Address*:
- Username*:
- Password*:

Below the form are links for "Show" and "Generate", and a "Submit" button. A "Resources" sidebar on the left contains links for "Login to PO System" and "Register to Access PO's". The footer text reads "Wednesday, 05 October 2011 © :: Kansas State University :: Manhattan, Kansas".

Figure 5: User Registration Page

This page is used for User Registration. All the fields in this page are mandatory. When the user enters the information and submits the page, an email is sent to the User along with a hidden confirmation code. When the user accepts the link provided in the email the user is then registered into the database and allowed to login and redirected to the Login page. This page also checks if a valid email address is being provided. It also checks if the username is already used or present in the database, if found then it shows a warning message.

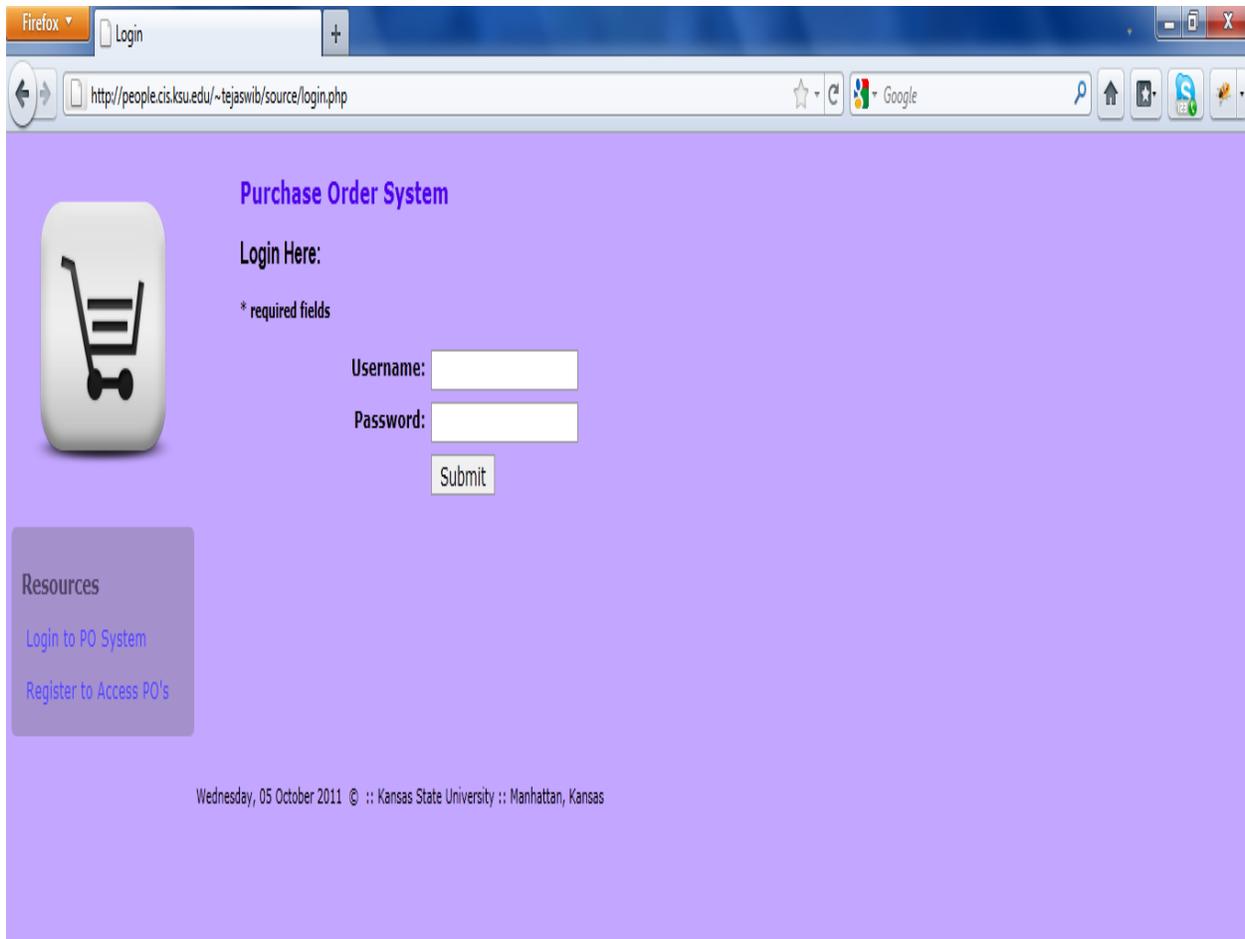


Figure 6: Login Page

This page is used for logging into the website. It validates the username and password before logging in the user.



Figure 7: The above screenshot shows the main Portal Page.

This is the main portal page that gives a summary of all the purchase orders made by the user until date. In this page we can also see that the purchase orders are divided into Locked and Unlocked PO's. The locked PO's are those that are confirmed for shipping and are ready to be finalized after the APO payment is made. Unlocked PO's are still the raw orders waiting for a confirmation from the Admin.



Figure 8: Showing the page to create a New PO

By using this page a user or admin can create a new PO. That means items can be ordered for a particular faculty member or staff member from the vendors already registered with the department. Here funding source is a required field. Once the PO is created, it is redirected to the view PO page where all the information of the PO is given.



Figure 9: Viewing the Purchase Order

This page briefs the full information about the purchase order by the po_number. This page allows the user to view the PO in PDF format for printing the order, editing the po, lock the po or finalize the po.



Figure 10: Editing the Purchase Order.

This page allows the user to edit the vendor information, shipping address, or add new items from the same vendor. Once the PO is updated, the user is redirected to the view po page.



Figure11: Viewing the po after updating it.

The above screenshot depicts a full summary of the purchase order after the items are added to the PO. It calculates the order total along with the freight and shows the total amount.

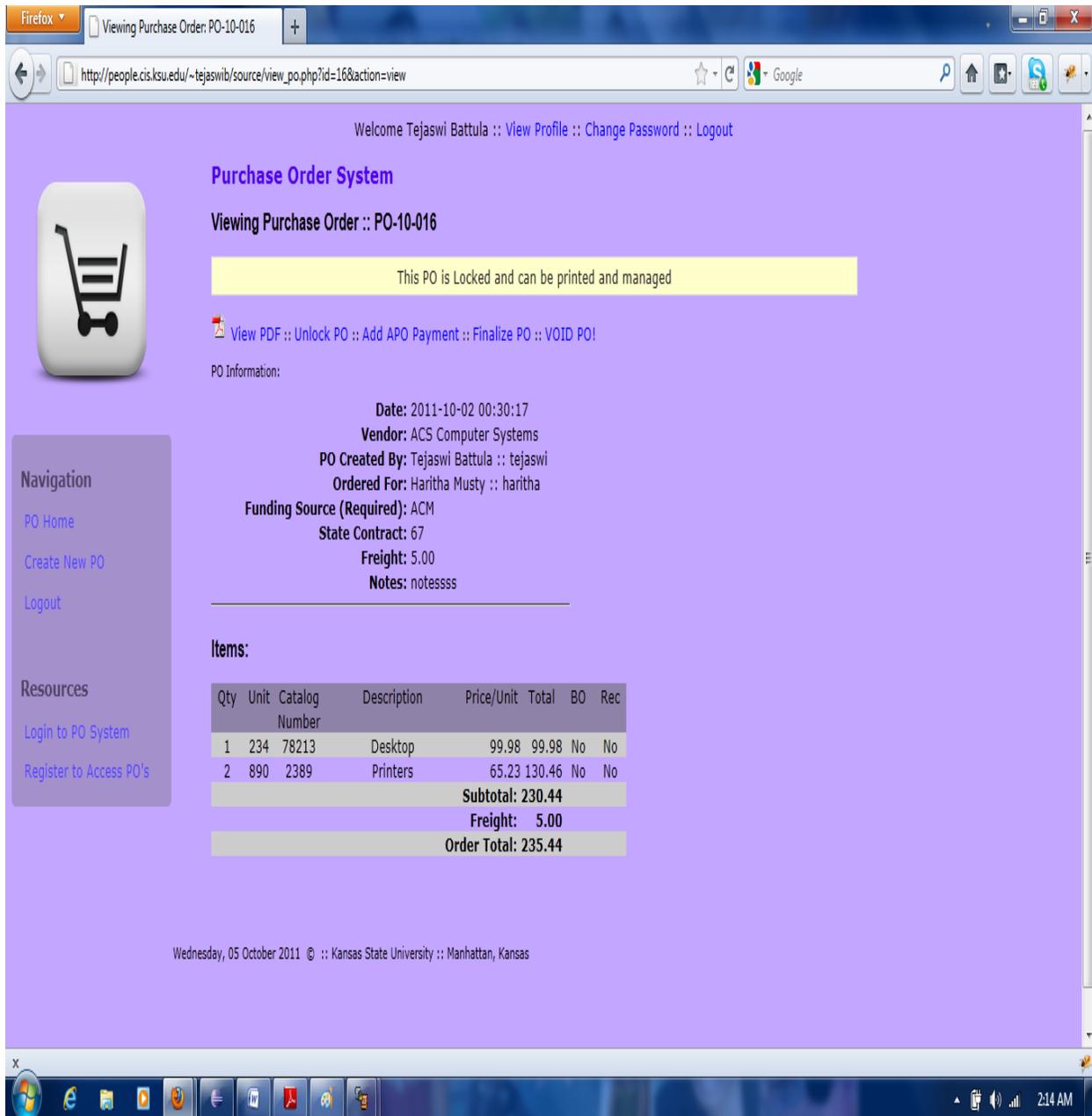


Figure 11: Showing the warning message “The PO is locked and can be printed and managed” once the PO is locked.

Once the PO is locked it is ready for adding the APO Payment and to be finalized or void the PO.

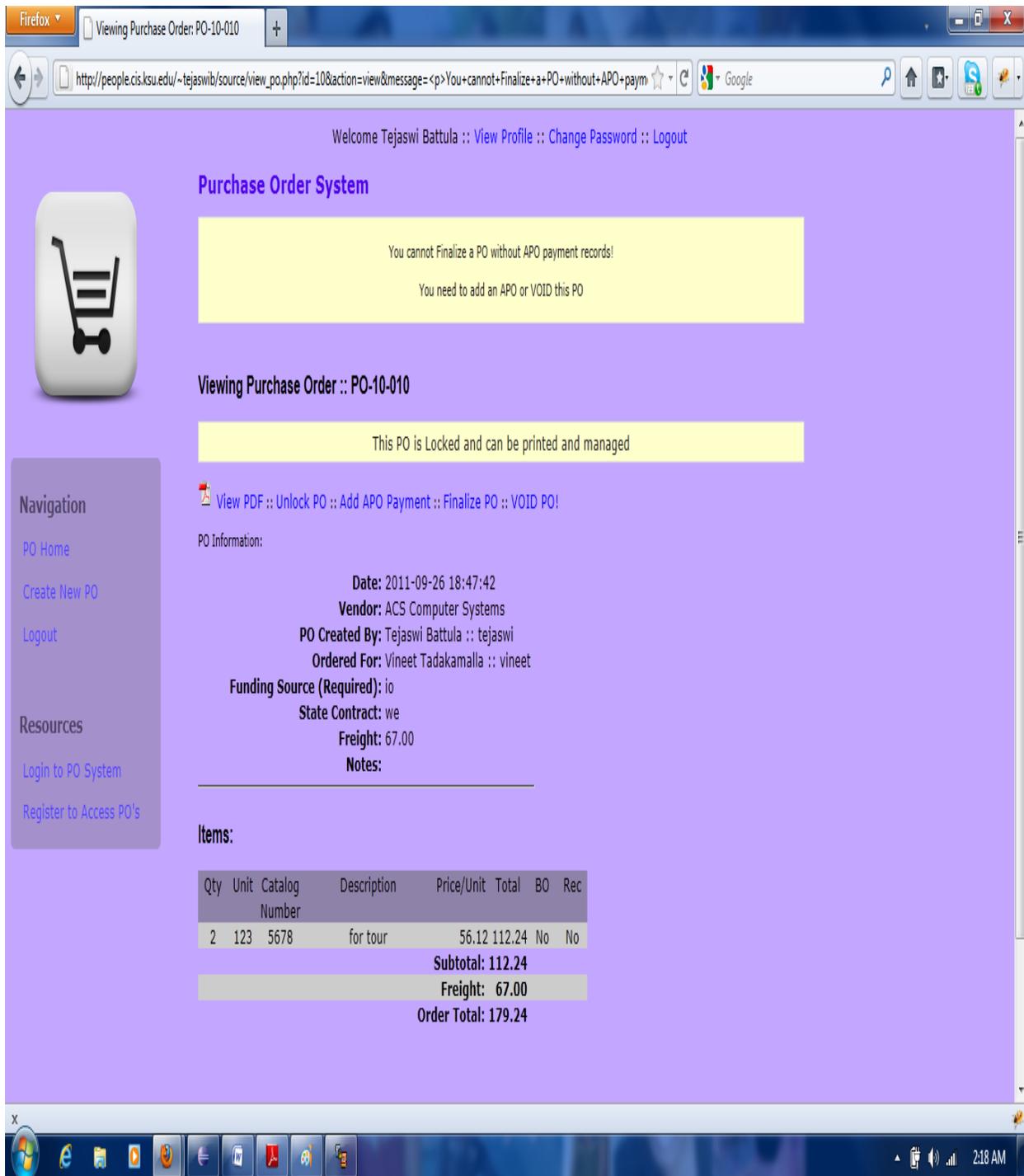


Figure 12: Showing the warning message when trying to finalize the PO before adding APO payment.

It does not allow finalizing the po without making it a void one or adding an APO payment. Else a warning message is generated prompting to enter the payment.



Figure 13: Adding the APO Payment.

This page shows the total order amount to be paid for all the items purchased to the vendor by providing the already registered APO number.



Figure 14: Screenshot showing after adding the APO Payment.



Figure 15: Finalize PO.

Once the PO has the APO payment provided and locked, then the PO is ready to be finalized. Finalizing the po determines that the po is complete and can be viewed and ready to print.

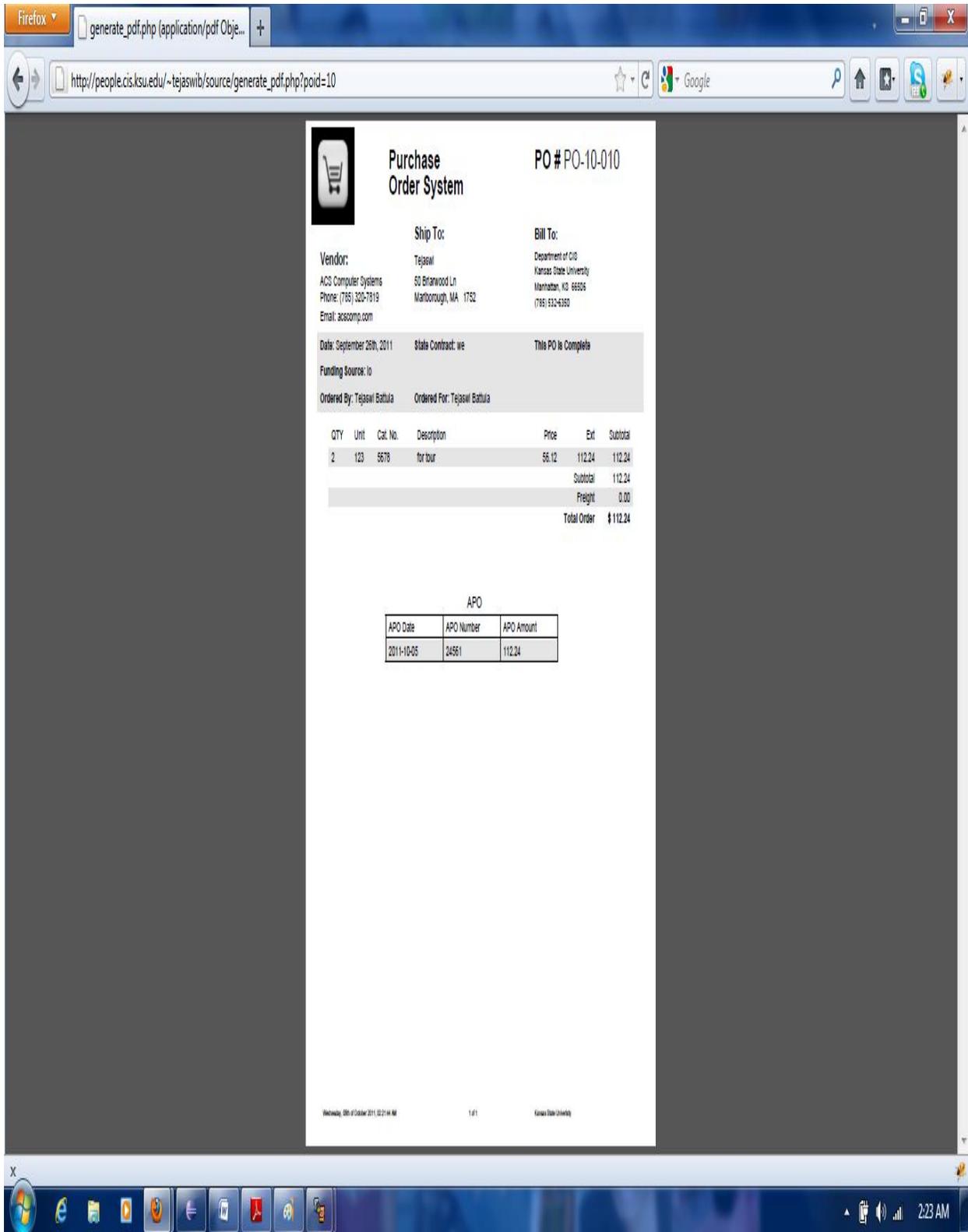


Figure 16: PDF format of the PO with all the details.



Figure 17: View Profile

This page shows the information about the user and allows him to edit his information.

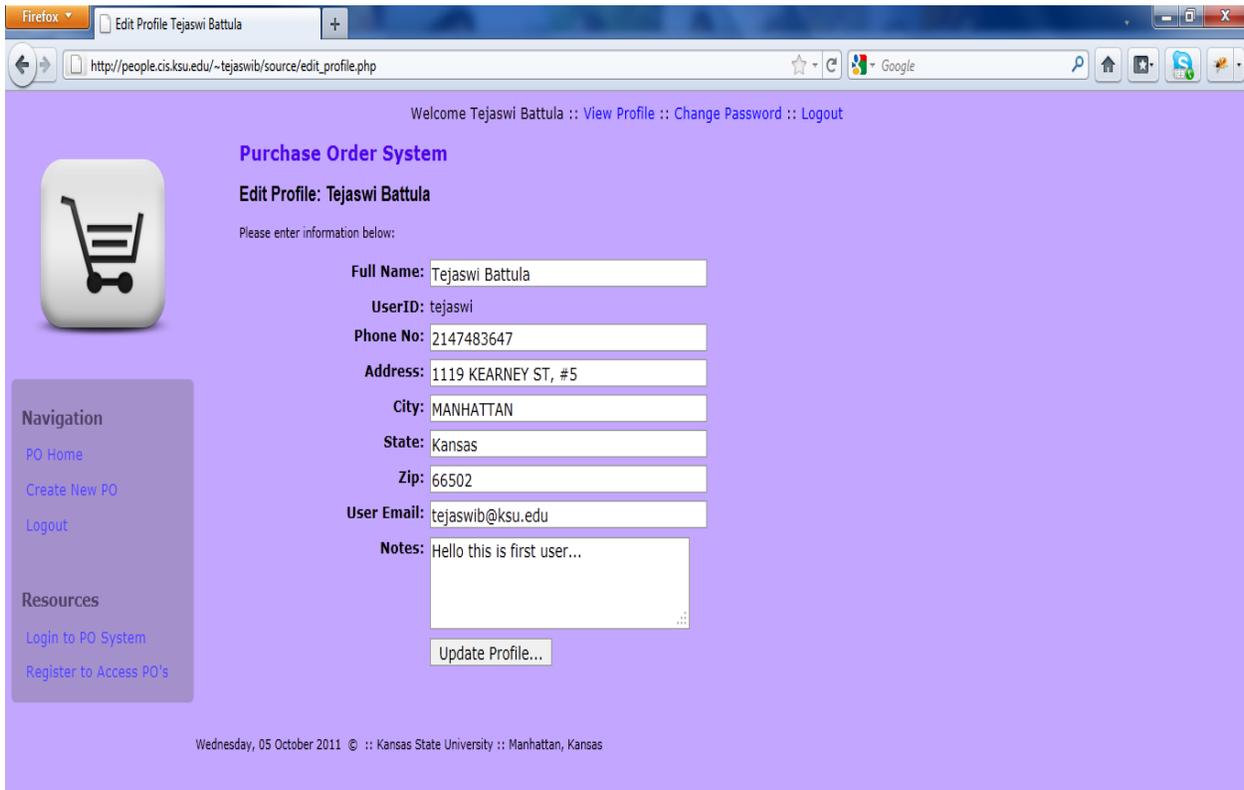


Figure 18: Update profile.

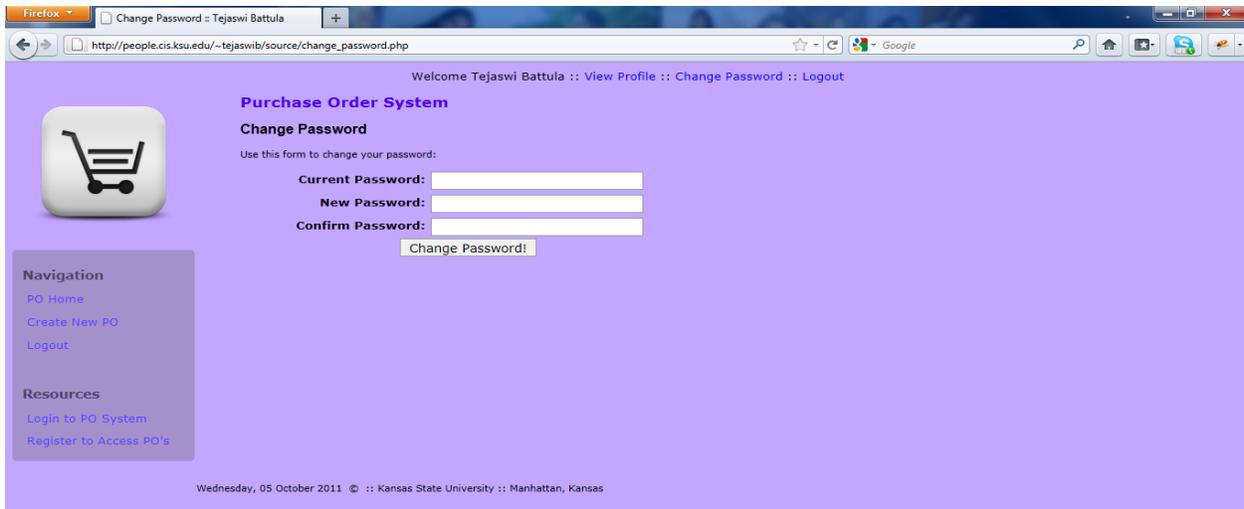


Figure 19: Change Password.

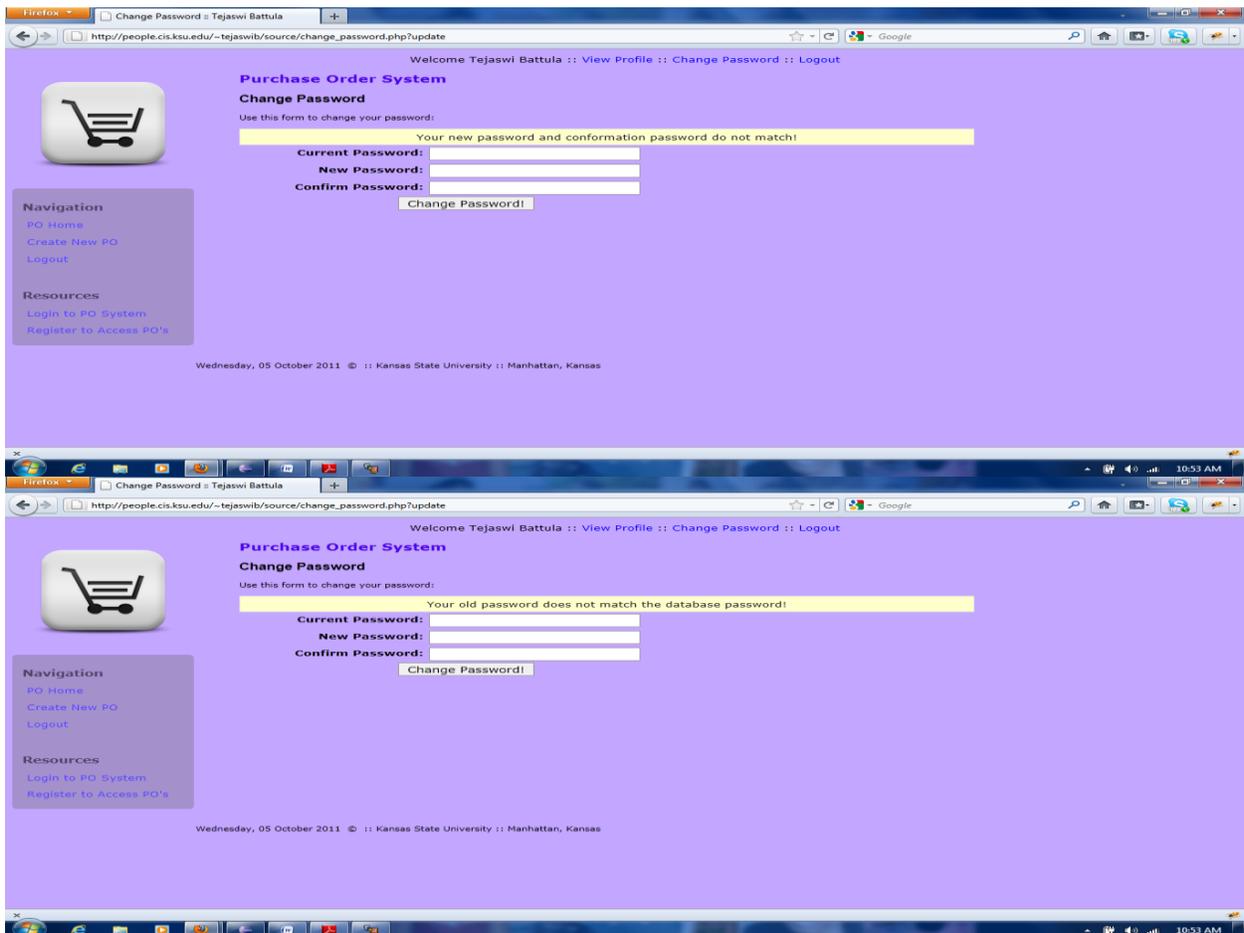


Figure 20: Showing the warning message if the old password or the new password and the confirmation password do not match.



Figure 21: Admin Page – Showing all PO's.

This page shows all PO's created by the users in the database. He can view and edit the PO's.



Figure 22: Admin Page – View/Manage Users.

The Admin can view all the users in the database. He also has the permissions to view the user information. He can also view all the PO's made by the user and can edit them.

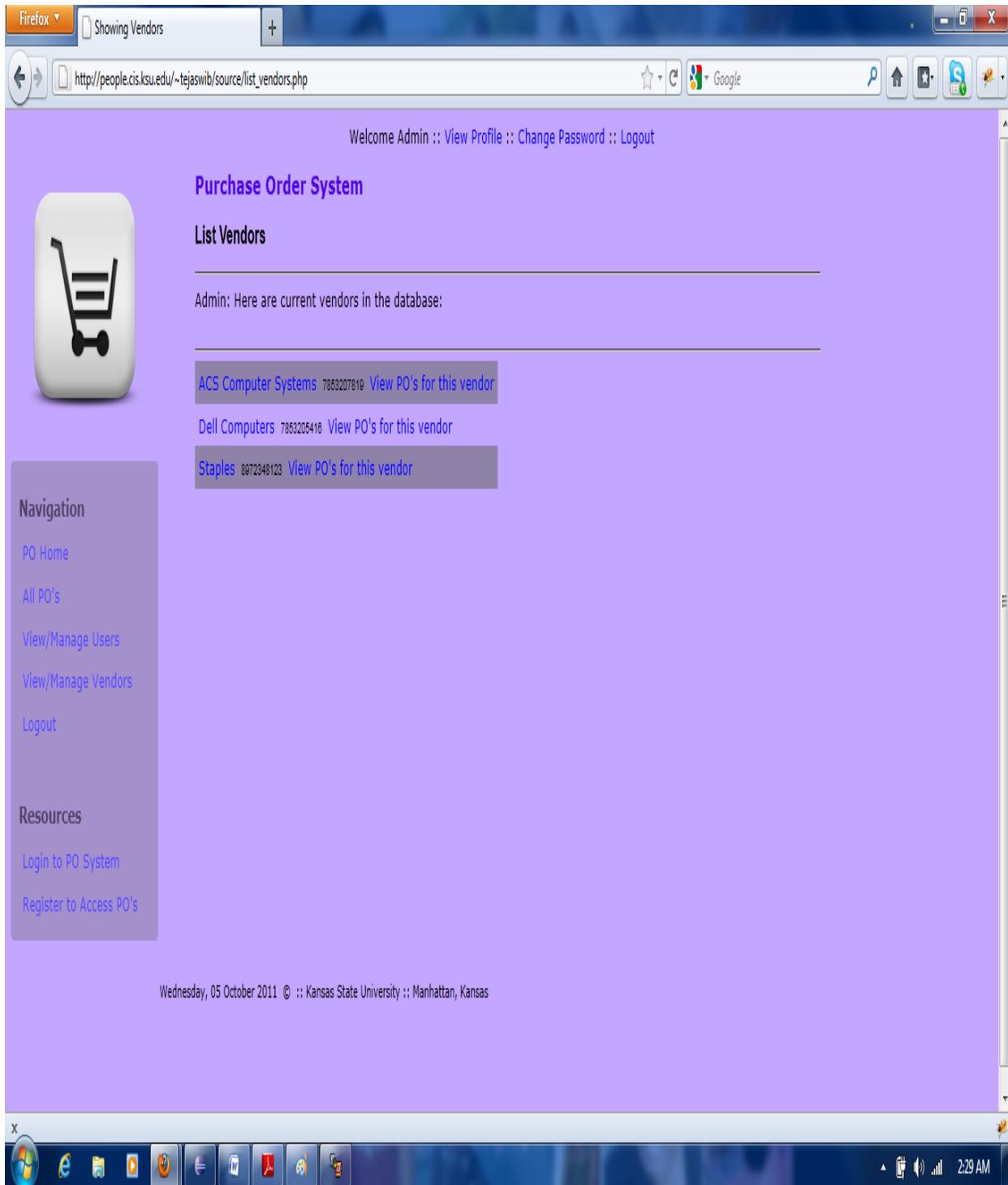


Figure 23: Admin Page – View/Manage Vendors.

The Admin can view all the vendors in the database. He also has the permissions to view the vendor information. He can also view all the PO's for a particular vendor.

5.3 Technical Discussion

This application was initially started for maintaining the purchase orders made for a department in a university. One of the best features of the system is that the admin has all the permissions to lock, view and manage the PO's before making any orders. Thus the orders are reviewed before they are placed.

Break down of the program code: Initially my estimation of lines of code was about 6500 lines. After the application was fully built the estimate has been increased to 7309 lines which include jQuery, Automated PHP code, handwritten code, HTML and CSS code. This application has admin site and a user site and are fully developed reasoning for this many number of lines of code. The breakdown is listed below in the table

Code Type	Number of Lines
Java Script	65
Automated PHP	2013
Hand Written PHP	3265
HTML	1512
CSS	454

Table 1: Code Type and Number of lines.

The total time taken for designing, developing this application and fully testing the application for its correct functionality was approximately 220 working hours in the span of 3 months.

6. TESTING

Software testing is the process of executing a program or system with the intent of finding errors. Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well.

6.1 White Box Testing

White-box testing is verification technique software engineers can use to examine if their code works as expected. It is a security testing method that can be used to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities. White box testing ensures if all the logical paths in the code are executed at least once. It also checks if conditional statements have solution for both the true and false conditions. Also checks if all the loops in the code are executed correctly.

The above testing method includes static testing. All the modules in the application have been tested rigorously tested for both correctness as well as the faults by giving both valid and invalid inputs and generating error and success messages. This kind of testing has been done throughout the lifecycle of the application. All the inputs that are given to the BLL have been validated against various criterions and any non-conformity to these validation rules have been dealt with user friendly messages.

6.2 Unit Testing

Unit testing is performed by the unit tool. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure. In object-oriented programming a unit is usually an interface, such as a class. Unit tests are created by programmers or occasionally by white box testers during the development process. All modules were tested during their coding phase for its functionality until modules have produced the desired results.

- The login module was tested by entering an invalid user not present in database. It was also tested by providing wrong password. A warning message would appear resulting invalid username or password.
- Test cases were written such that all modules are tested.
- The user cannot complete or finalize the purchase orders until and unless they make an APO payment.
- No void purchase orders can be created at the first stage.
- The user cannot change his password without presenting the right old password. Else a warning would be raised.
- Admin can view all the purchase orders and be able to edit them.
- Admin can also manage/view users and manage/view vendors.

The unit generates all possible scenarios and the tests were passed. Each and every module has been tested by generating test cases thoroughly.

6.3 Non Functional Testing

Non functional testing demonstrates how well the product behaves. It is concerned with the design that evaluates the readiness of a system. It ensures that an application performs in a right manner when undesired inputs are given. For example the email needs to contain @ symbol else it would pop up an error message asking to enter a valid email address. Also checks if the user accepts the confirmation email while registering for the first time else it would not allow the user to login to the application.

Validation testing also ensures that both the functional testing and non functional testing at various places in the application.

6.4 Performance Testing

Performance testing is performed to determine how fast a system performs under a particular load. It is also used to validate and verify other attributes of the system such as scalability, reliability and resource usage. Load testing is primarily concerned with testing that can continue to operate under specific load be it large amount of data or be it numerous users.

To perform load testing I have used Apache JMeter which can be used to test performance both on static as well as dynamic resources. This simulates a heavy load on the server, network or object to test its strength or to analyze overall performance under different load types.

I have tested on the pages such as portal page and login page all these tests have been performed with the application and load testing tool on the same server. I have tested the application for constant load as well as under peak load conditions to see how well it behaves.

Users	Ramp –Up period	Loop Count	Portal Page Average Response (ms)
50	5	500	945
100	5	500	995
500	5	500	1011

Table 1: showing the average response speed for Portal page

The above results show that the average response time for the portal page increases as the number of users increase.



Figure 24: Meter screenshot for portal page.

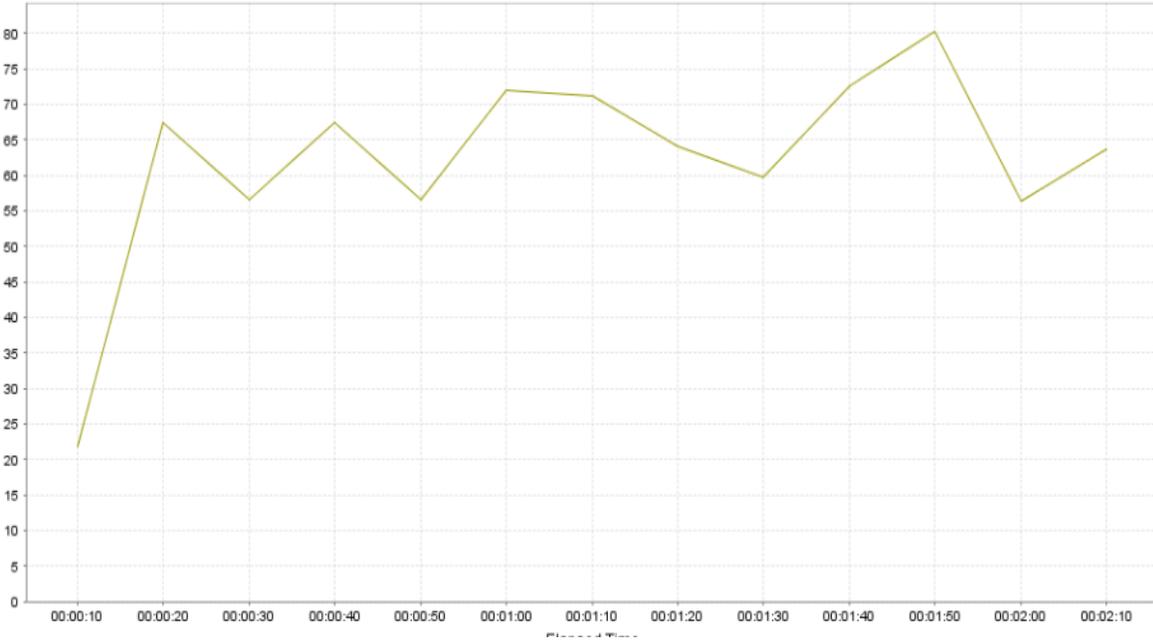


Figure 25: Throughput for a simple page in the application, most of the controls are static.



Figure 26: Throughput for a complex page, with all dynamic information.

The throughput for this kind of page drastically decreases after certain number of requests.

7. RESULTS & CHALLENGES

For now this application has been designed in such a way that it can be used for only a particular department for Faculty members or staff members to order any purchase orders. But it can be extended more to manage or organize the purchase orders for medium scale business organizations since it has different features like filtering products on various criterion, easy navigability through the website, very simple yet effective method of adding and updating the purchase orders.

7.1 Challenges

Major challenges that I have encountered while developing and designing the website are:

- Effective designing of web pages so that they are rendered properly in the various web browsers like IE, Mozilla Firefox, and so on.
- Database design was also one of the major challenges as there shouldn't be any redundancy.
- Normalization of tables that have been designed.
- Working with PHP to publish PDF files in the website to publish the receipt of the purchase order made.
- Using web services in the application.

8. CONCLUSIONS

Purchase order system an online purchasing system has been designed and developed to serve the purpose of web application and user friendly interface. This application has various features to view, edit or update the purchase orders, easily manageable, editable and updatable profile. The Administrator has other features like managing the users and vendors, can create new vendors and new PO's.

8.1 Limitations

In the present system the users only can create a PO and provide all the information from which vendor he want to buy and to which address the PO should be shipped for and for which faculty member he is ordering for. Then Admin would place the order requested by the Faculty or Staff.

8.2 Future Work

From the above limitation we can say that we can extend the application to be more users friendly by giving full access to the user itself to make the purchase orders. The vendors can also be given access to the system. Drag and drop of PO's into the cart can be added.

9. REFERENCES

[1] Overview of PHP

<http://www.php.net/>

[2] PHP Scripting Language

<http://www.w3schools.com/php/default.asp>

[3] How to use query

http://docs.jquery.com/How_jQuery_Works

[4] Software Testing Overview

http://en.wikipedia.org/wiki/Software_testing

[5] Interactive Diagrams

http://en.wikipedia.org/wiki/Interaction_overview_diagram