



Masters Project Final Report

January 2017

Project Title	PHP Framework for Inventory Management System for UCSC		
Student Name	D.M.G.N. Dissanayake		
Registration No. & Index No.	2014MIT014 14550141		
Supervisor's Name	Dr. Ajantha Athukorale		
Please Circle the appropriate	Master's Program		Type
	<u>MIT</u>	MCS	Research <u>Implementation</u>

For Office Use ONLY



PHP Framework for Inventory Management System for UCSC

A dissertation submitted for the Degree of Master of Information Technology

D.M.G.N.Dissanayake

University of Colombo School of Computing

2017



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: D.M.G.N.Dissanayake

Registration Number: 2014 MIT 014

Index Number: 14550141

Signature:

Date:

This is to certify that this thesis is based on the work of

Ms. D.M.G.N.Dissanayake

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

Signature:

Date:

Abstract

It is a well-known fact, that accurate control and management of inventory of an organization is one of the foremost tasks, in order for it to reach its vision and mission. 'Inventory' can be defined as the objects or goods that are sold, distributed or used by an organization. Primary objective of management of the inventory is to determine/control stock levels within the physical distribution system. Inventory management involves processes that identify inventory requirements, setting targets, providing replenishment methods, reporting inventory status, and handling all functions related to the tracking and management of goods. Controlling the inventory mainly focuses on using the inventory efficiently keeping up with changes in demand to avoid overstock and avoiding item spoilage.

University of Colombo School of Computing (UCSC), a unit of University of Colombo is a non-profitable organization, which is currently experiencing some challenges regarding their inventory management and control, such as troubles of cross-referencing the locations of inventory items, inaccurate need analysis, and uncoordinated purchasing processes that most of the times cause duplication of records and data redundancy. In addition to that, there are certain issues identified as too much paper work and unawareness on available stock. Several studies have shown that paper based supplies records are far from a good solution whenever timely reports are a major requirement.

The goal of this research is to identify needs specified to the organization to design and develop a web-based inventory control platform, which can be actually initiated in UCSC, which should be capable of inventory registration, ordering and distribution with maximum efficiency and reliability, and efficient stock controlling.

The study is done in a qualitative manner. It mainly follows SDLC. Interviews and document reviews were effectively used to gather data and the system designing is done using the prototyping methodology.

Acknowledgements

I would like to acknowledge my deepest appreciation to my supervisor Dr. Ajantha Atukorale, Senior Lecturer of University of Colombo School of Computing for his valuable support and guidance.

I am also grateful to my external supervisor Dr.K.G.H.D.Weerasinghe, Senior Lecturer, University of Kelaniya for his support and guidance.

Special thanks goes to Mr.Weerasiri,Deputy Bursar/Supplies, other staff members of the supplies division, UCSC for providing me with this opportunity to develop the system according to their requirements, providing necessary information amidst their busy work schedule.

Also I would like to give my gratitude for the academic staff members of University of Colombo School of Computing for all the knowledge and support given throughout the MIT degree program which immensely helped to complete the dissertation and the final project.

Finally I would like to thank Ms.H.R.M.P.Ranaweera, Mr.L.W.V.Priyadarshana and my family for their support and encouragement throughout the whole time.

Table of contents

Abstract	iii
Acknowledgements	iv
Chapter 1: Introduction	2
1.1 Background	2
1.2 Motivation.....	3
1.3 Problem Statement	4
1.4 Scope & Objectives	4
1.5 Structure of the Thesis	6
Chapter 2: Literature Review	8
2.1 Introduction.....	8
2.2 Related Literature on Automated Inventory Control	8
2.3 Summary	10
Chapter 3: Methodology	12
3.1 Introduction.....	12
3.2 Research Design	12
3.2.1 Research Approach.....	12
3.2.2 Study Population & Sample Size.....	12
3.2.3 Data Sources	12
3.2.4 Requirement Gathering.....	12
3.3 System Development	14
3.3.1 System Development Life Cycle (SDLC)	14
3.3.2 Database Design	15
3.3.3 Prototyping	16
3.3.4 Design and Modeling Tools	18
3.3.5 Development Process	20
3.4 Summary	22
Chapter 4: Implementation	24
4.1 Introduction.....	24
4.2 Importance of System Implementation	24
4.3 Implementation Methodology.....	24
4.4 Introduction to IMS	24
4.5 Development Environment	24
4.5.1 Hardware and Software Environment	25
4.5.2 Development Tools	25
4.6 Software	26

4.7	Database Implementation	26
4.8	Code Segments	27
4.9	User Interfaces	30
4.10	Network Implementation.....	31
4.11	Summary	32
Chapter 5: Testing and Evaluation		34
5.1	Introduction.....	34
5.2	Testing Strategy	34
5.2.1	Unit Testing	34
5.2.2	Integration Testing.....	35
5.2.3	System Testing	35
5.2.4	Acceptance Testing.....	35
5.3	Test Plan	35
5.4	Test Cases	35
5.5	Evaluation	37
5.6	User Evaluation.....	37
5.7	Summary	37
Chapter 6: Conclusion and Future Work.....		39
6.1	Introduction.....	39
6.2	Lessons Learnt	39
6.3	Future Improvements	40
References		41
Appendix A: Test Plan		43
Appendix B: User Manual.....		44
Appendix C: Test Cases		57

List of Tables

Table 4.5.1:Hardware and Software Environment	25
Table 4.5.2:Web server configurations.....	25
Table 4.5.3:Client machine configurations.....	25
Table 5.3.1: Test case structure	35
Table 5.4.1: Test cases for login panel	36
Table 5.4.2: Test cases for adding funds	37
Table A.1: Test plan	43
Table C.1: Test cases for inventory user interface	57
Table C.2: Test cases for category user interface	58
Table C.3: Test cases for sub category user interface	58
Table C.4: Test cases for PO user interface.....	60
Table C.5: Test cases for funds interface	60

List of Figures

Figure 1.1.1: Information systems	3
Figure 3.2.1: High level use case diagram.....	13
Figure 3.3.1: SDLC activities	15
Figure 3.3.2: Database design.....	16
Figure 3.3.3: Prototyping process for IMS	17
Figure 3.3.4: Category registration user interface	17
Figure 3.3.5: Item registration user interface	18
Figure 3.3.6 : Flow chart of inventory management in UCSC.....	20
Figure 3.3.7: Conceptual Data Model for IMS.....	21
Figure 4.7.1: Creating the database	27
Figure 4.7.2: Creating a table	27
Figure 4.8.1: Coding of user login interface.....	28
Figure 4.8.2: Coding for Inventory Registration	29
Figure 4.9.1: User login.....	30
Figure 4.9.2: Inventory registration.....	31
Figure 4.9.3: Supplier registration	31
Figure 4.10.1: Network implementation	32
Figure 5.2.1: Testing methodology	34
Figure B.1: User login interface	44
Figure B.2: Item category interface.....	45
Figure B.3:Item sub category interface	46
Figure B.4: Inventory item user interface.....	47
Figure B.5: Supplier category interface.....	48
Figure B.6: Supplier registry interface	49
Figure B.7: Funds user interface	50
Figure B.8: Fund allocation user interface	51
Figure B.9: PO user interface	52
Figure B.10: GRN interface.....	54
Figure B.11: Report dashboard.....	55
Figure B.12: Report generation tray	55
Figure B.13: User registration interface	56

List of Abbreviations

CSS – Cascading Style Sheet

GIN – Good Issue Note

GIO – Good Issue Order

GRN – Good Received Note

GUI – Graphical User Interface

HTML – Hypertext Markup Language

HTTP – Hypertext Transfer Protocol

IMS – Inventory Management System

LAMP – Linux Apache MySQL PHP

PDF - Portable Document Format

PHP – Hypertext Pre Processor (Recursive term)

PO – Purchase Order

POS – Point of Sale

RAM – Random Access Memory

RDBMS – Relational Database Management System

SDLC – Software Development Lifecycle

SO - Supply Order

SQL – Structured Query Language

UCSC – University of Colombo School of Computing

UML – Unified Modelling Language

VAT – Value Added Tax

CHAPTER 1

INTRODUCTION

Chapter 1: Introduction

1.1 Background

Inventory management is the practice of overseeing and controlling of the ordering, storage and use of components that a company uses in the production of the items it sells. Inventory management is also the practice of overseeing and controlling of quantities of finished products for sale [1]. Use of a computerized inventory management system can make everything from inputting information to taking inventory easier. Doing a hand count of inventory can take days, but with a computerized inventory management system, it is a matter of hours. Such a system can be used to generate all kinds of documents automatically, from purchase orders and cheques to invoices and account statements, for managers and workers. It is a common weakness in manual systems, that the data is only as accurate and up to date as the last hand count. But with a computerized inventory management system, the management team can pull a report and instantly see how many items are left in the stock, how many are sold/distributed and which products have the fastest movement.

Non profitable organizations such as universities mostly rely on government grants for their expenses. Other than that, UCSC has allocated their funds also for item purchases for the organization. UCSC is composed of several divisions and units which function separately, under the administration of the director, deputy directors, senior assistant registrars and deputy bursar. Other than that there are other non-academic staff members who directly involve in the duties related to administration of the organization. The supplies division makes purchases, store items, distribute items for all those units and divisions. The main store, item requests, suppliers and fund allocations are managed by them.

Successful organizations leverage technology to manage financial processes and assist in making decisions. Information systems are used by the organizations to collect data and process them according to the needs of the management/administration. Efficiency on operations is the key objective of using information systems in organizations, in addition to interact with external vendors and partners, curtail costs and generate revenues. There are several types of information systems that can be used for UCSC supplies division, as shown in Figure 1.1.1.

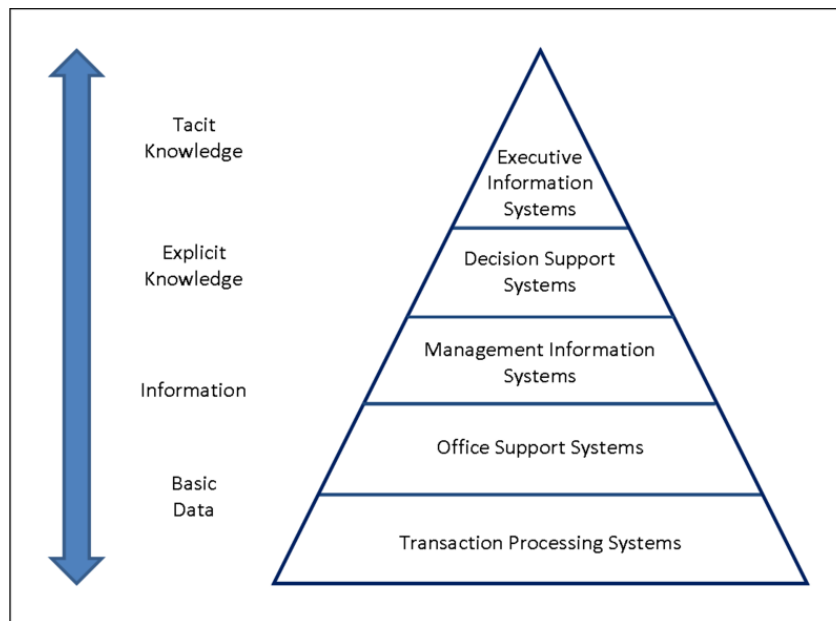


Figure 1.1.2: Information systems

Among these types of information systems, a Management Information System was found to be most suitable for management of inventory in a university. A management information system regulates all organizational data of an organization, in a computerized database. Management/administration can retrieve data from all sections of the organization including inventory, purchases, and issues to see how efficiently each unit is operating.

1.2 Motivation

Keeping track of inventory item purchases, delivering, identifying, sorting of items and keeping track of their location inside the organization are essential tasks for public sector. An organization can avoid the delays, paperwork, effort, which it has to be put into managing inventory, by carrying out its purchasing / supply management function in whole or in part in an electronic environment. There are very few organizations in public sector who manage their inventory in an automated manner.

UCSC currently does not use a fully automated methodology to manage their inventory related processes. Ongoing UCSC inventory management process flows on a standalone procurement platform whereas supplies division places purchase orders with suppliers of their own choosing for goods and services as and when they are required. They maintain a small database of suppliers, who were registered at the beginning of the year, objects within their premises, and fund allocated for each unit or department. Whenever an item is requested by any unit or department of UCSC, they generate a purchase order to a relevant supplier. When the supplier

supplies the item/items, a payment voucher is created to pay the supplier for the goods. When the requested item goes to the stores, a Good Received Note is generated by the stores, and when it is issued to the unit or department, a Good Issue Note is generated. Most of the data, which are relevant to these processes are recorded manually on a daily basis, on specially designed forms which are then filed for reference purposes. A properly automated process for these tasks will allow normalization of all data in the supplies division and will make the process of inventory management easier for employees.

1.3 Problem Statement

Although the current process for inventory managing in UCSC is fulfilling the daily needs of inventory management, it is outdated and has some major drawbacks.

- It lacks information about items including indications of how these items were obtained, by which grant, their availability in the stores, purchase history, current location and behavior.
- It lacks information about vendors, relevant purchase history for each vendor, payments and contact details of them.
- It lacks information about units and departments, details of their requests history.
- Each partially automated tasks are being processed in standalone computers by a single user (book-keeper, clerk, computer applications assistant).
- It takes time to summarize inventory, purchases, good arrivals and issues.
- A large volume of paperwork is required to carry out the whole process.
- Absence or mistake of a single employee can hold a process for a while because the whole process is a collection of single user tasks. Other employees might not be aware of his/her job done within the process.
- Most essential features of payment involved processes, like security and confidentiality of information are not enough.

1.4 Scope & Objectives

The proposed system will automate the following processes.

Supplier Management

- A new supplier can be added to the supplier registry.
- Details like address, contact details of the supplier can be edited.

Inventory Management

- An inventory registry is maintained. Several categories of items and details of items within those categories are inserted.
- Details of items and categories are updated.

Purchase Order Management

- Making purchase orders for items are generated as per requests from any department/unit.
- A purchase order can be cancelled or edited if any need arises (wrong quantity inserted by mistake, the department wants to change the order etc.).

Good Receival Management

- Making good received notes for items which are received from suppliers as per requests.
- A purchase order can be cancelled or edited if any need arises (wrong quantity inserted by mistake, the department wants to change etc.).

Users

- System users are assigned privileges according to the different roles they would perform within the system.
- Assigned users are provided with their login details (username, password).

Reports

- Reports containing detailed summaries of inventory/stock, purchase orders are generated.
- A suppliers registry containing details of suppliers is generated annually.

The main objective is to develop a working model of an automated inventory management system that is capable of performing the main functions of registering assets and materials purchased, minimizing paperwork and manpower, ensuring user satisfaction, and controlling inventory within the premises. Other main objectives are,

- To generate identification codes for assets within the client's premises, according to government policy.
- To initiate an automated strategy for inventory handling of the client.

- Reduce man-hours spent for report generation.
- Enhancing consistency and ease of management of procurement.
- Developing a system with lower budget and higher effectiveness.
- Bring the advantages of having the most efficient control with less effort and employees.
- Develop a common conceptual framework for inventory management of universities that is both complete and internally consistent.

1.5 Structure of the Thesis

Hereafter, Chapter II provides an overview of the most relevant academic & industrial literature.

Chapter III describes both research and development methodologies used to complete the research objectives. Presents IMS model with details of all the notations. Object oriented approach adopted during the project is explained including diagrams used for designing and outlines of the main user interfaces with screen shots.

Chapter IV describes the development procedure, implementation environment, software tools and technologies used at the time of the development of the system.

Chapter V describes how the system was tested using sample data and the outcomes of the assessment.

Conclusion and Future work is the final chapter of the dissertation, which includes the critical evaluation of the completed system, including details about the future improvements and the lessons learnt.

CHAPTER 2
LITERATURE REVIEW

Chapter 2: Literature Review

2.1 Introduction

The purpose of this chapter is to review past efforts that relate to the proposed IMS, with their limitations or weaknesses, which make them insufficient for the required solution. There are several researches and implementations done in the areas of production, operations and inventory management. Inventory management is the process of acquiring goods from another organization, managing their behavior, and equipment in a legal and ethical manner, which happens to be one of the most important stages in the supply of goods and service delivery. It also plays a critical role in the quality of service. The studies cited in this chapter highlight the key features in the subject of inventory management.

2.2 Related Literature on Automated Inventory Control

A good inventory tracking system will tell you what merchandise is in stock, what is on order, when it will arrive and what you've sold. With such a system, you can plan purchases intelligently and quickly recognize the fast-moving items you need to reorder and the slow-moving items you should mark down or specially promote. An inventory management software is such a computerized inventory control system for tracking inventory, inventory levels, orders etc.

As the author points out, generally, inventory related data were stored in hard-copy form or in spreadsheets. Several enhancements can upgrade such a system to create orders, invoices and other related documents. Usage of inventory control systems in large organizations pave the way to avoid product overstock and outages. It is stated that the required features are similar to a general POS system[2].

An IT Enabled Inventory Management System has been developed for National Informatics Centre, Department of Information Technology, Govt. Of India, as an enhanced version of a manual system, to keep track of detail regarding the equipment they use. It runs on a Windows platform and was developed using Visual Studio .Net 2003, with SQL Server 2000. It provides the basic services related to the supply of the equipment to maintain their PRE-SO (Supply Order) and POST-SO details. Also, it manages all supply orders, keeping records of each supply order. The system assigns a unique ISG Number generated by BRO (Border Read Organization). It is responsible for supplying items to different project departments of BRO,

maintaining the POST-SO worksheet to prepare the supply and liability to the current year and forming a guide sheet that includes budget allotted; liabilities of last three years cleared in current year, liabilities to be carried forward to next year.

The system has been able to provide the interface to the user so that he can replicate his desired data. Also foreground interaction with user is as smooth as possible. Each request details are preserved along with their transaction related to them. The system is also secured as all the updating and transaction can be done by the authorized personnel.

Although there are several advanced features than a manual system, this system does not normalize data of items, according to categories/sub categories, which is an essential task for inventory management. Also there is no categorization of the suppliers. Data entry errors cannot be checked, only the validation required with reference to the system is checked[3].

ERP Sales and Inventory Management System is another study which targets on developing an ERP Sales and Inventory Management System (SIMS) for a departmental store, for all types of retail companies and top level multinational companies, those who are manufacturing fast moving products for customer attraction towards the systems of respective companies.

The system is to be used to store the details of the inventory, stock maintenance, update the inventory based on the sales details, generate sales and inventory reports periodically etc. It categorizes individual aspects for sales and inventory management system, solving various problems affecting direct sales managed by RSM (Regional Sales Manager), ASM (Area Sales Manager) and SO (Sales Officer) those who are monitoring response in term of target on various industrial products. The goals are to provide the user an efficient working environment, support decision making, in an adaptable, more secured manner[4].

2.3 Summary

This chapter of related literature and studies, points out the specific outcomes expected from an inventory management system, mostly in large organizations. There are common goals to achieve in this context in professional inventory control.

In general, inventory control systems maintain information about activities within organizations that supports their tasks. The subsystems that perform these functions can include sales, stock, manufacturing, warehousing, ordering, and receiving. Sometimes these functionalities can be in separate subsystems. These functions are essential altogether to have a fully functioning inventory control system.

CHAPTER 3

METHODOLOGY

Chapter 3: Methodology

3.1 Introduction

This chapter describes the methodologies used in both research and development based on the objectives to be met. The research methodology includes the data collection methods used while the development methodologies includes the tools used in gathering requirements and system development techniques. The final detailed project schedule with deliverables to be achieved at the end of the entire project is also included here.

3.2 Research Design

3.2.1 Research Approach

Among the different research approaches that can be used; in this case, the approach used was case study research. A case study, in a business context is a report of an organization's implementation of something, such as a practice, a product, a system or a service. The case study can be thought of as a real-world test of how the implementation works, and how well it works[5]. Therefore, it was found to be the most suitable approach for this study. It paves the way to carry out a deep study for one aspect of a problem, identifying organizational features, work processes, their effect on the implementation of systems and their influence on organization's workflow.

3.2.2 Study Population & Sample Size

The study population is gathered from the non-academic staff of Supplies division, UCSC. This group includes deputy bursar/supplies, several book keepers and several computer application assistants.

The projected number of people to gather information from was about 10 people and was able to interview 7 people.

3.2.3 Data Sources

Primary data sources were interviews, brochures, emails, procurement plan, year planner, organizational structure, etc.

Other sources of data were payment vouchers, purchase orders, expenditures, supplier listings, stock listings, invoices and receipts, purchase order books, etc.

3.2.4 Requirement Gathering

Requirements gathering was done by analysis of current business processes. Interviews were focused on understanding the processes in the current system and were conducted with staff of

the supplies division, UCSC. Following, Figure 3.2.1 shows the high-level use case diagram of the current manual process.

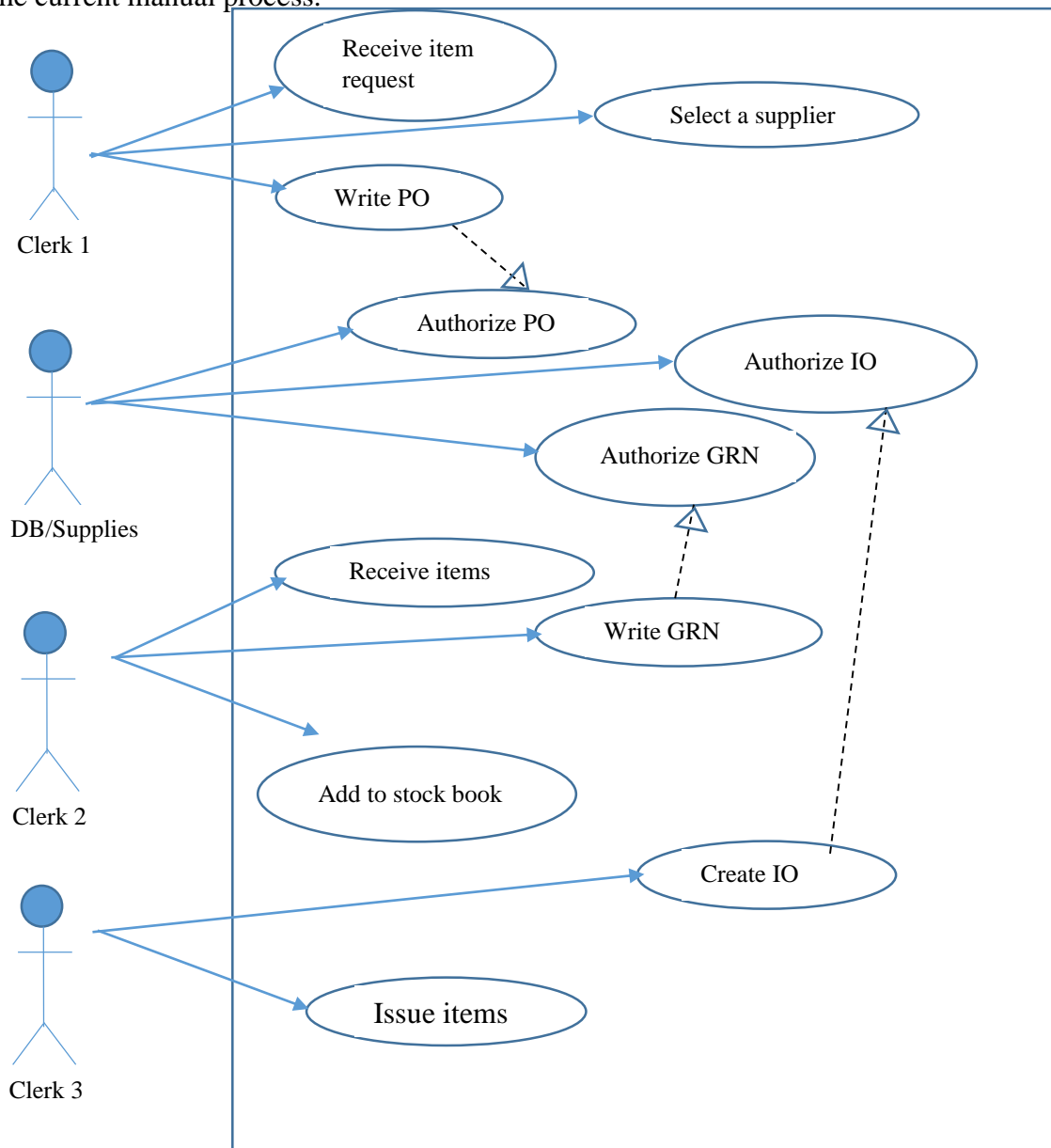


Figure 3.2.1: High level use case diagram

Potential IMS users were asked several questions, and they were observed to establish the daily activities/processes, to see what applications were being used and how they were used in carrying out daily tasks of the division. The main objective of conducting interviews was to define functional and non-functional requirements, to gain an understanding about what the intended system users would want incorporated in the proposed system.

3.2.4.1 Functional Requirements

- Users should be able to maintain inventory and suppliers registries.
- Inventory should be divided into categories and their sub categories.

- Users should be able to allocate funds to different units.
- Users should be able to generate POs, GRNs.
- Users should be able to print generated POs, GRNs whenever required.
- Users should be able to take summarized reports of purchases and inventory.

3.2.4.2 Non-functional Requirements

Non-functional requirements are the constraints on the functions offered by the system. Similar to the name, these requirements are not directly concerned with the specific functions delivered by the system. They relate to the system as a whole rather than to individual system features. Given below are the non-functional requirements expected in IMS.

- The system should be easy to use and user friendly. The interfaces should be simple and consistent. It should give an opportunity to users with an average computer skill to operate the system in a short time.
- The system should be accurate and consistent because it involves usage of university funds.
- Security is a very important feature in the system because it contains financial data. Hence access privileges should be implemented and granted for the appropriate users in the supplies division in order to enhance confidentiality.
- Reliability plays a major role here because it's performance effects UCSC's academic and administrative functions.
- Regular backups should be taken and stored in a secure location where it can be used in a case of system/ data failure.

3.3 System Development

3.3.1 System Development Life Cycle (SDLC)

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process [6]. The basic steps of SDLC are shown in Figure 3.3.1.

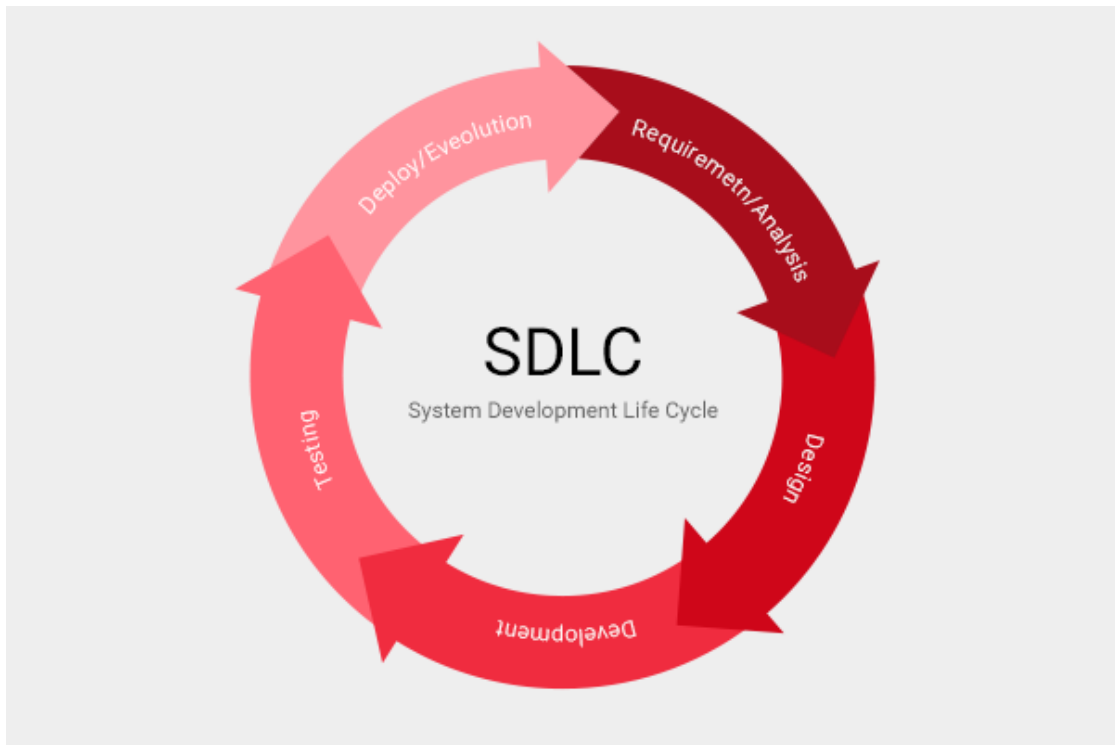


Figure 3.3.1: SDLC activities

Although SDLC follows a sequential flow of activities, some requirements would not be completely specified. Prototyping helps in this matter to identify the requirements clearly and specifically, when some functionality is not fully understood.

3.3.2 Database Design

Redundant data will be minimized with the usage of databases, by using database normalization methods. Default values are assigned for some fields to minimize the data mismatch and collision. Figure 3.3.2 shows the database design of IMS.

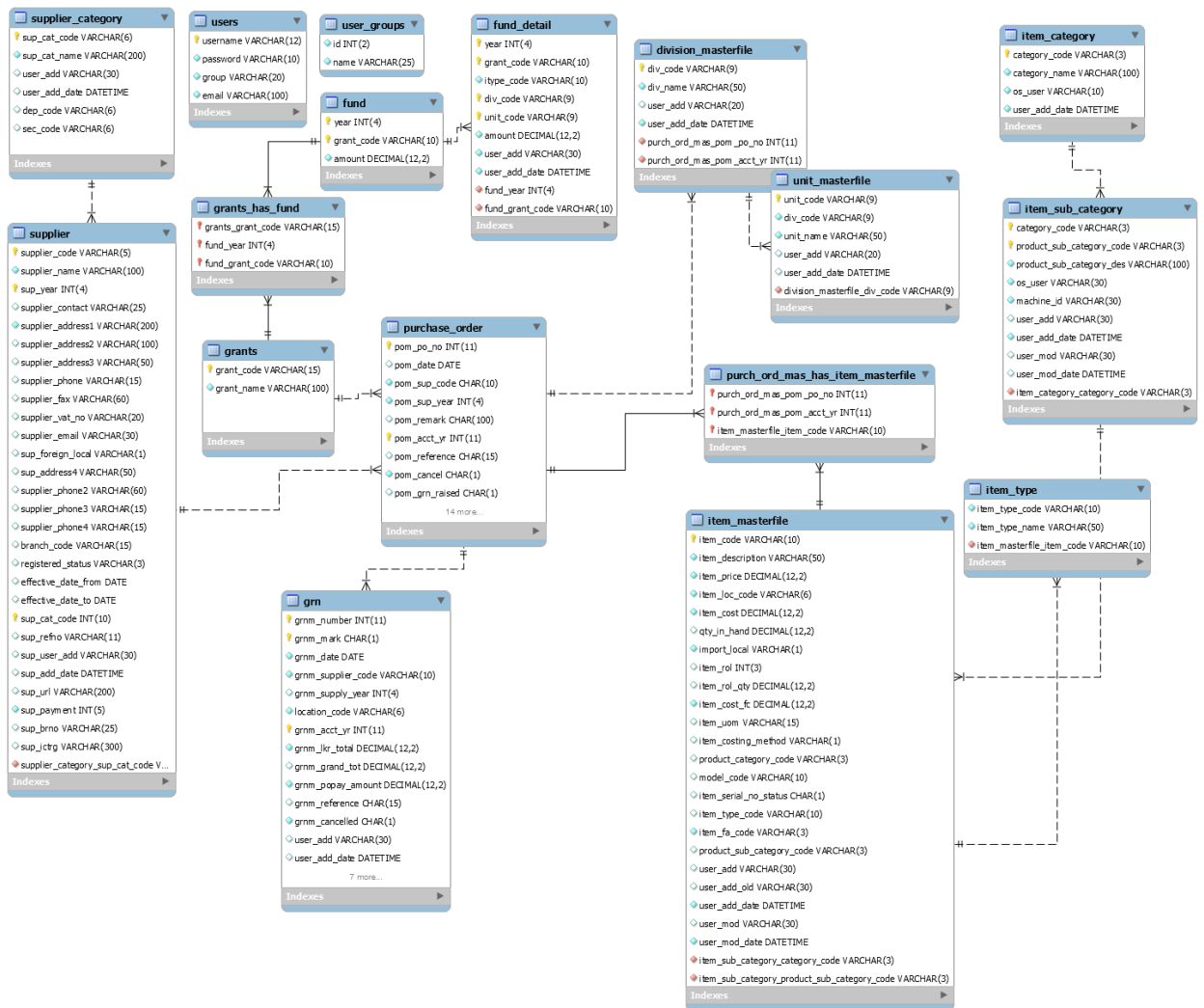


Figure 3.3.2: Database design

3.3.3 Prototyping

In software development, a prototype is a rudimentary working model of a product or information system, usually built for demonstration purposes or as part of the development process. In the systems development life cycle (SDLC) Prototyping Model, a basic version of the system is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can be developed [7].

A prototype was developed and shown to the users where they provided comments and feedbacks, which were used to re-analyze, re-design, and re-implement a new version of prototype that provided a few more features. Figure 3.3.3 shows this process.

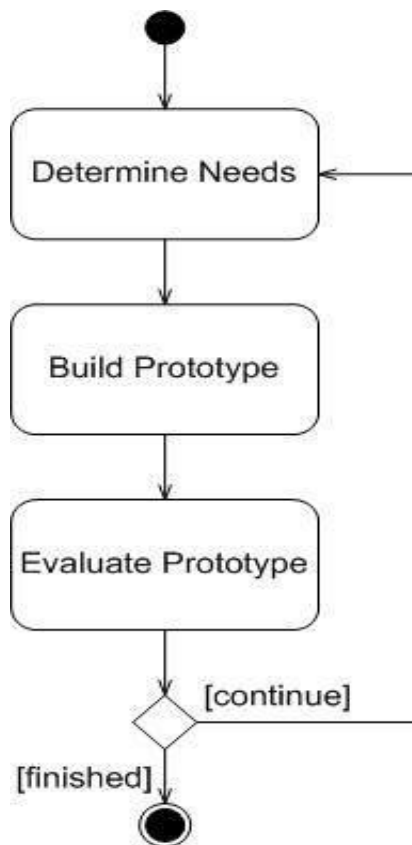


Figure 3.3.3: Prototyping process for IMS

Following are some interfaces from the prototype developed.

Figure 3.3.4 shows the category registration user interface. This generates the category code automatically and allows the user to give a preferred category name.

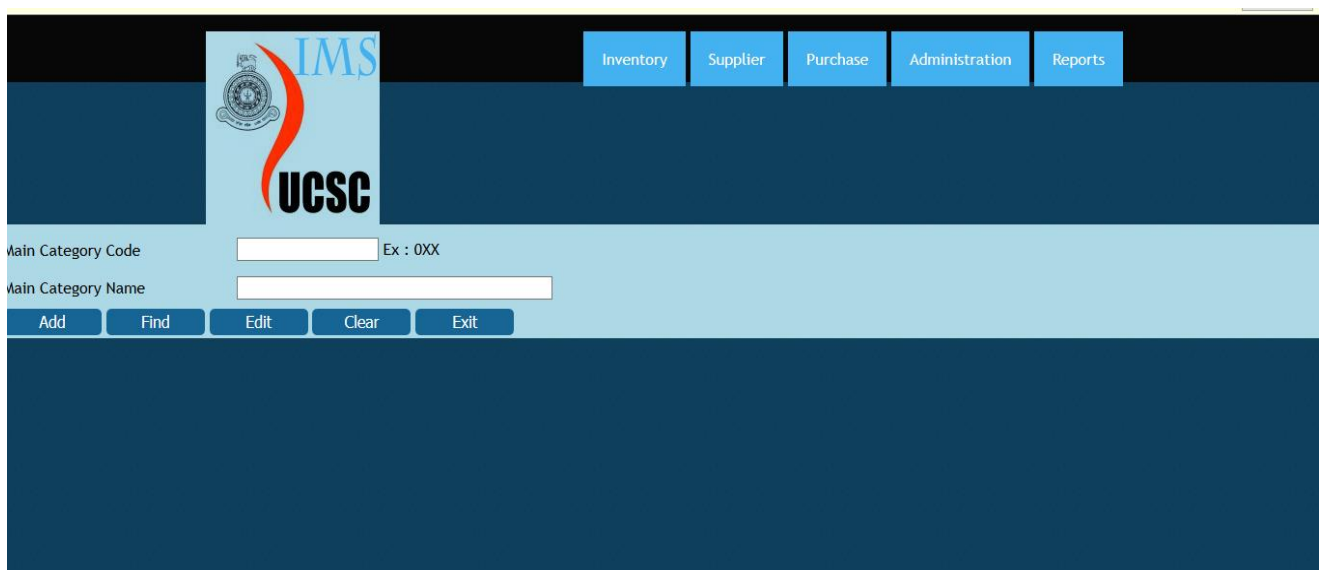


Figure 3.3.4: Category registration user interface

Figure 3.3.5 shows the item registration user interface. Whenever a new item should be purchased, it can be added to the inventory registry through this interface.

The screenshot displays the 'Item registration user interface' for the IMS UCSC system. The interface is organized into several sections:

- Navigation:** A top bar with tabs for 'Inventory', 'Supplier', 'Purchase', 'Administration', and 'Reports'.
- Item Detail:** A section containing dropdown menus for 'Main Category' and 'Sub Category', and a 'Local' dropdown under 'Imported / Local'.
- Existing Items:** A section with input fields for 'Code', 'Item Type', and 'Description'.
- Unit Information:** A section with dropdown menus for 'U.O.M' and 'Costing' (set to 'FIFO'), and input fields for 'Re-Order Level' and 'Re-Order Qty.'.
- Actions:** A row of buttons labeled 'Save', 'Clear', and 'Exit'.

Figure 3.3.5: Item registration user interface

3.3.4 Design and Modeling Tools

3.3.4.1 Importance of Design Phase

Designing phase usually involves problem solving and planning up solution for software. The output of this phase will consist of the specification, which describes both WHAT (system will do) and HOW (it will work) [8].

The design phase of IMS starts with requirement gathering from the UCSC supplies division and maps the requirements into design. The design defines the components, their behaviors and user interfaces. The design documentation puts forward a plan to implement the identified requirements. A good design has several key factors.

- Accuracy - the extent to which a program's outputs are sufficiently precise to satisfy their intended use [9].
- Reliability - Once a software system is functioning, as specified, and delivered the reliability characteristic defines the capability of the system to maintain its service provision under defined conditions for defined periods of time. One aspect of this characteristic is fault tolerance that is the ability of a system to withstand component failure [10].
- Usability – the degree to which the software is able to be used.

- Interoperability - the ability of the software to exchange and make use of information.
- Security - an idea implemented to protect software against potential risks so that the software continues to function correctly under such condition.
- Maintainability - ease of updating the software to satisfy new requirements after it has been developed.
- Efficiency - the ratio of the useful work performed by the system to the resources needed.

3.3.4.2 Design Techniques

Unified Modeling Language (UML)

Unified Modeling language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. UML is an important aspect involved in object-oriented software development. It uses graphic notation to create visual models of software systems [11]. In this study, UML was used to model most if not all of the diagrams needed in the system development process, including use cases, class diagrams, activity diagrams, entity-relationship diagrams, and the conceptual data model. All these diagrams were designed using the Smart Draw 2016 tool.

Software requirements

- Linux server with LAMP enabled
- Apache server 2.4.9
- PHP 5.5.12
- MySQL 5.6.17

Programming language

PHP - an HTML-embedded script language that assists in developing dynamic pages. It is used to connect the user interfaces to the database, create forms, user interfaces, sessions to store user information, etc.

Database Management System

MySQL - a free open source relational database management system. It can run on many operating systems. It is also easy to use, fast, and is scalable and hence accommodates large amounts of data thus providing high level security for the stored information.

Hardware

2.0+ GHzs processor, 8 GB system RAM, and 16 GB free hard drive space.

3.3.5 Development Process

3.3.5.1 System Planning

This is an over view of the intended IMS, identifying the need and determined scope of the entire project, work plan and scope specification determined.

3.3.5.2 System Analysis

During this phase, requirements were gathered and analyzed. Models were built which would be used at the design phase and these models include the process flows at UCSC, use-case diagrams, conceptual data model, and functional and non-functional requirements. Given below in Figure 3.3.6 is the process flow in the inventory controlling in UCSC.

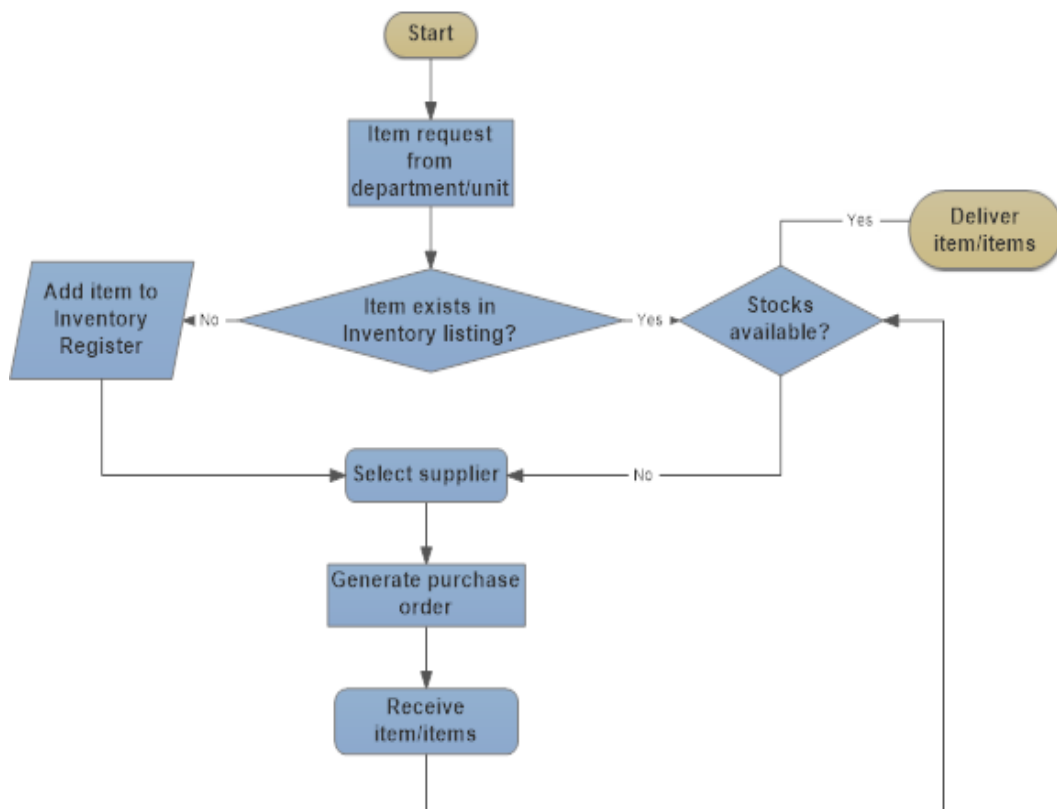


Figure 3.3.6 : Flow chart of inventory management in UCSC

Conceptual Data Model

The conceptual data model shows the relationships between the proposed entities. Figure 3.3.7 shows the conceptual data model in IMS.

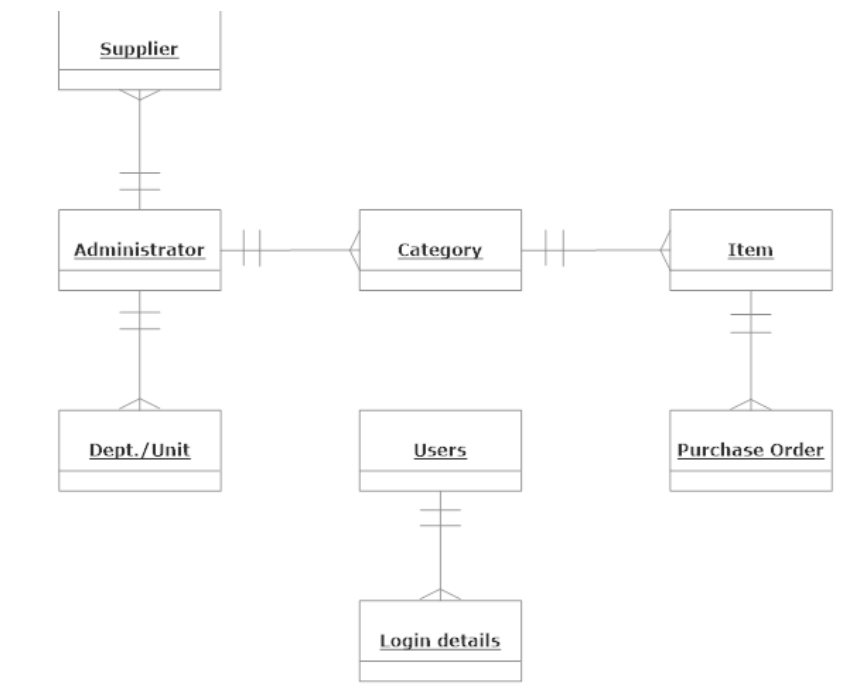


Figure 3.3.7: Conceptual Data Model for IMS

3.3.5.3 System Design

This phase includes selecting the preferred database software and building the design, and later the actual designing of the system using inputs from the analysis phase. This provides inputs for the implementation phase of the system and includes functional design, entity relationship diagram, logical design of the database, physical design of the database, user interface designs, systems architecture, network model and the program design.

3.3.5.4 System Implementation

This is the final phase in the project life cycle. Conversion of the system design into the actual system takes place, developing interfaces and programming using selected software and linking to the database. The system was then tested, planned for the installation and user training is organize for users on how to use the system. Documents such as user manual were compiled at this phase.

3.4 Summary

The aim of this chapter was to explain the methods used in both research and development in the designing and implementation of an automated inventory management system for UCSC supplies division. Also an overview of the next phases is provided along with the deliverables and a detailed schedule for the IMS development.

CHAPTER 4

IMPLEMENTATION

Chapter 4: Implementation

4.1 Introduction

This chapter describes the steps carried out in the implementation phase of IMS. It includes details about overall implementation of the proposed system, its key features, their intended usage, the selection of tools for the implementation and justifications for selecting those tools. This chapter also includes several parts of the codes used in the system development.

4.2 Importance of System Implementation

The purpose of the implementation is to deploy and enable operations of the new information system in the actual environment. During this phase, physical system specifications are transformed into a working and reliable system and the work is documented to assist current and future system users. Design specifications are translated into codes, database is created and user interfaces are created to interact with the system users. Successful completion of the implementation phase should comprise system deployment and training on the system.

4.3 Implementation Methodology

Agile methods or Agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals [12]. Agile implementation methodology was found to be very much suitable for this system due to the fact that it can be used to develop a working version that the client also can involve from beginning. It was easy to develop user interfaces in a way that is simple and user friendly as well as to meet the exact user requirements with minimal documentation.

4.4 Introduction to IMS

IMS is a GUI based application. The chapter presents the reader with the overall implementation of the proposed system, its major components, their intended usage, the selection of tools for the implementation and justifications for selecting those tools.

4.5 Development Environment

Selecting the most suitable development tools for a software system is very essential for its successful implementation. Below discussed is the selection of the programming language, database system and other tools which are used to develop the system effectively.

4.5.1 Hardware and Software Environment

Table 4.5.1 shows the hardware and software configuration for development environment.

Hardware Configuration	Software Configuration
2.30GHz Intel Processor	Apache Web Server
1GB RAM	PHPMyAdmin
2GB free hard disk drive space	MySQL 5 +
1024 x 786 Compliant color monitor	PHP 5+
	Notepad++
	Adobe Photoshop

Table 4.5.1:Hardware and Software Environment

Table 4.5.2 shows the web server hardware and software configurations.

Hardware Configuration	Software Configuration
2.0+ GHz Intel Processor	Linux Server with LAMP enabled
2GB RAM	Apache server 2.4.9
10GB free hard disk drive space	PHP 5.5.12
	MySQL 5.6.17

Table 4.5.2:Web server configurations

Table 4.5.3 shows the client machine's hardware and software configurations.

Hardware Configuration	Software Configuration
1.0+ GHz Intel Processor	Java script enabled Web Browser
500+ MB RAM	PDF Reader software

Table 4.5.3:Client machine configurations

4.5.2 Development Tools

Adobe Photoshop was used to design the logos and other imagery items.

Notepad++ is a free source code editor which supports several programming languages running under the MS Windows environment [13], like HTML, CSS and scripting languages like PHP, JavaScript. This tool was used to code and develop the system.

4.6 Software

As per the client's request was to provide a stable economical solution, open source software were used as much as possible. Implementation was done using PHP and MySQL server technologies.

PHP - (recursive acronym for *PHP: Hypertext Preprocessor*) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML[14]. It is a server side, user interactive, programming language, works nearly in on all platforms. It can be programmed with various databases such as MySQL, SQL, Oracle, MS SQL etc.

JavaScript - is a scripting language which enables programmatic access to computational objects within a host environment. It is mostly used to implement dynamic web sites.

My SQL - is a reliable open source RDBMS which is used for many web based and standalone software systems. MySQL was selected as the database for IMS to store data used within the processes.

Apache - is an open source HTTP web server which is most widely used in the internet. It is a powerful web server program on many system platforms, with many features, therefore it is a reliable and efficient solution for the system as the web server.

4.7 Database Implementation

WAMP server was used in implementation of the database. Managing and handling data in the database were done using SQL queries.

The following figures, Figure 4.7.1 and Figure 4.7.2, include the database queries used to create database for IMS.



Figure 4.7.1: Creating the database



Figure 4.7.2: Creating a table

4.8 Code Segments

Figure 4.8.1 shows the code segment of user login interface, displaying the PHP code used for communicating with the database and html code used for creating the outline of the user login interface.

```

D:\IMS\index.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
ptitem.php x item-io.php x ptitem-io.php x index.php x login.php x inventory.php x php.ini-development x php.ini-production x php.ini x php.ini x phpForApache.ini x New Num...

1 <?php
2 ob_start();
3 session_start();
4 require('db.inc.php');
5
6 if (isset($_POST['btnLogin']) || $_POST['btnLogin']=="Login")
7 {
8     $username=$_REQUEST['username'];echo "<script> alert('".$username."');</script>";
9     $username=strtolower($username);
10    $password=$_REQUEST['password'];
11    $datetime=strftime("%Y-%m-%d %H:%M:%S");
12    $sql = "SELECT *,count(*) as login FROM `users` WHERE `username`='$username' and `password`= '$password'";
13    $result= mysql_query($sql) or die("<script> alert('Invalid Username or Password'); document.history(-1);</script>");
14    if($row=mysql_fetch_assoc($result))
15    {
16
17        if($row['login']==0)
18        {
19            echo "<script> alert('Invalid Username or Password'); document.history(-1);</script>";
20        }
21        else
22        {
23            $_SESSION["login"]= true;
24            $_SESSION["username"] = $username;
25            $_SESSION["currentdate"] = $datetime;
26            header("location:inventory.php");
27        }
28    }
29 }
30 }
31 ?>
32 <!DOCTYPE html>
33 <html>
34 <head>
35     <meta charset="UTF-8">
36     <title>IMS UCSC</title>
37     <link rel="stylesheet" href="css/style.css" type="text/css">
38 </head>
39 <body>
40     <div id="header">
41         <div class="wrapper clearfix">
42             <div id="logo">
43                 <a href="index.php"></a>
44             </div>
45         </div>
46     </div>
47 </div>
48 <div id="contents" align="center">
49     <div id="adbox">
50         <div class="wrapper clearfix">
51             <div id="ad">

```

Figure 4.8.1: Coding of user login interface

Figure 4.8.2 shows the code segment of inventory registration interface, displaying the PHP code used for saving a new item in the database and a part of html code used for creating the outline of the user interface.

4.9 User Interfaces

Some interfaces of IMS are listed below. Rest of the interfaces are shown under the user manual in Appendix B.

Figure 4.9.1 shows the user login interface. The user has to enter a valid username and password in relevant fields to access the system.

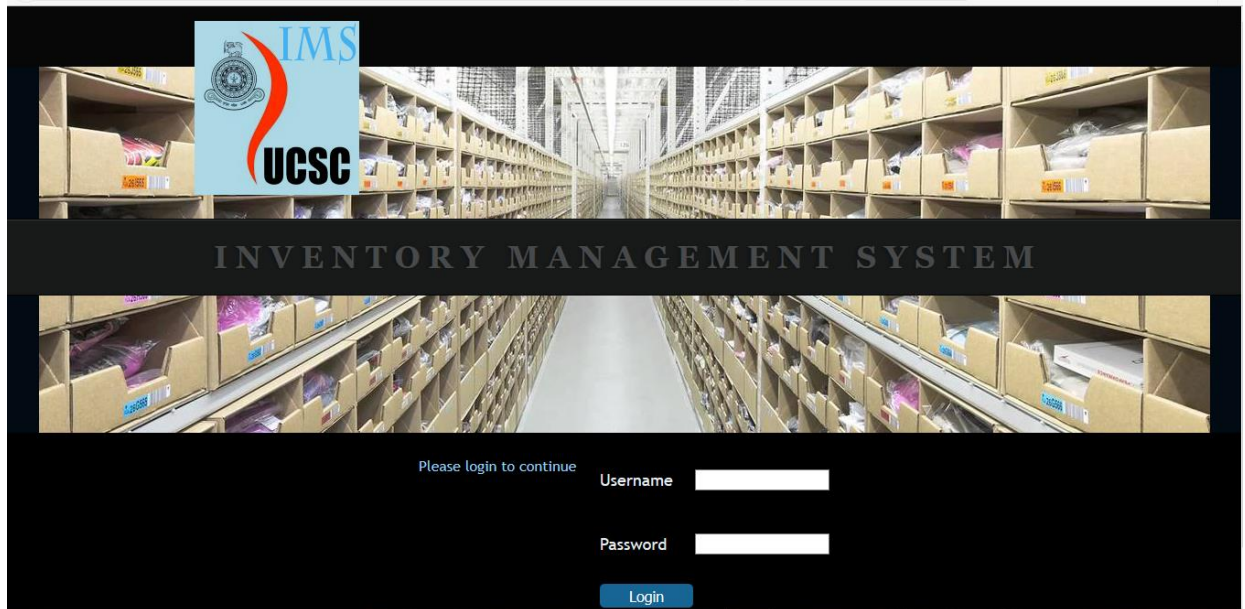


Figure 4.9.1: User login

Figure 4.9.2 shows the inventory registration interface. The user should select the relevant category and sub category. This will display the existing items of the same sub category to make sure that the item is not listed before. Item code is then automatically generated. Then the user should enter other details of the new item and save the item.

Figure 4.9.2: Inventory registration

Figure 4.9.3 shows the supplier registration interface. The user can put a single supplier into one or more supplier categories regarding the types of goods supplied by them. Then the user should insert the details of the supplier such as name, address, contact etc.

Figure 4.9.3: Supplier registration

4.10 Network Implementation

The client server network implementation utilized in the system is shown in Figure 4.10.1.

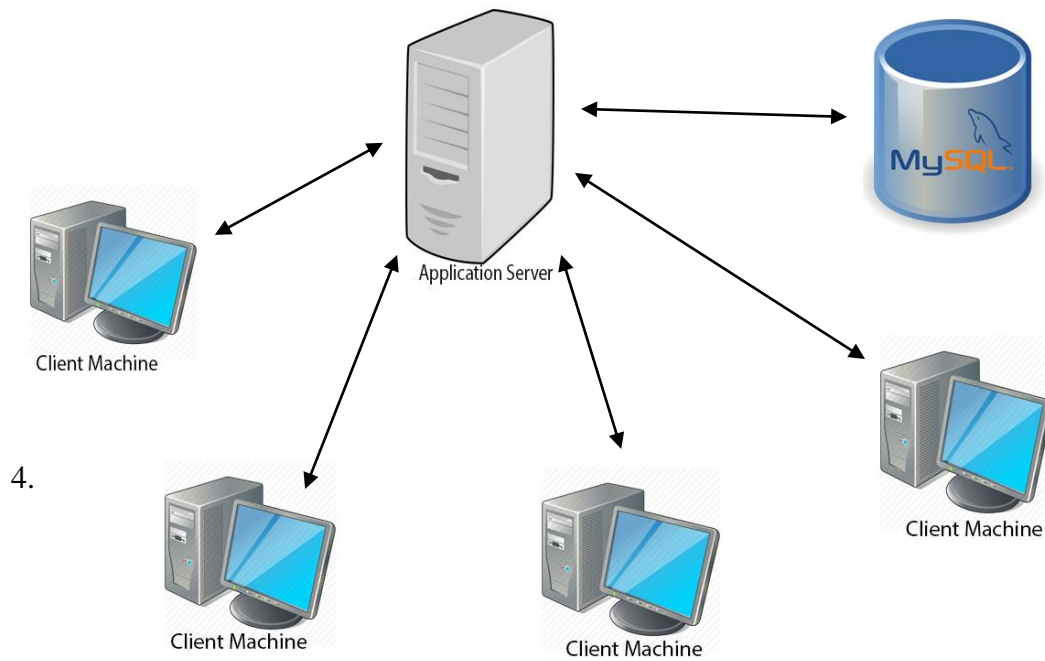


Figure 4.10.1: Network implementation

4.11 Summary

Implementation is a critical stage of the process, which is done in a well-organized and a structured manner, considering all hardware and software requirements. Agile methodology was used to get continuous user involvement from the beginning and to complete the system development rapidly. PHP was mainly considered as the programming language since it is more familiar with the developer.

CHAPTER 5

TESTING AND EVALUATION

Chapter 5: Testing and Evaluation

5.1 Introduction

Testing is the process of validating and verifying a system. The system or its components are compared against requirements and specifications through testing. Evaluation of test results combines with assessing progress of design, performance, supportability, etc. This chapter discusses how a comprehensive evaluation was carried out to assess the degree of success associated with the project. The chapter focuses on giving the reader an idea of the importance of evaluation, user evaluation approach and defining a set of evaluation criteria. Afterwards, the chapter will step into selection of target audience for testing and discussion of the evaluation results obtained.

5.2 Testing Strategy

Manual testing was used to test IMS, without using any test automation tool. Testing was performed for different levels such as unit testing, integration testing, system testing, and user acceptance testing. Following figure, Figure 5.2.1 displays the hierarchy of testing methodology used.

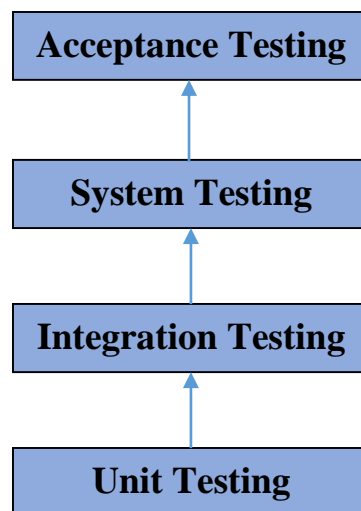


Figure 5.2.1: Testing methodology

5.2.1 Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed[15]. Unit testing is performed under white box testing methodology.

5.2.2 Integration Testing

In integration testing, individual units are combined and tested as a group, to expose faults in the interaction between integrated units. The relevant definition of ‘unit’ decides the methodology to be any of black box testing, white box testing, and gray box testing.

5.2.3 System Testing

System Testing is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. Usually, black box testing method is used[16].

5.2.4 Acceptance Testing

In acceptance testing stage, system is tested for acceptability, to evaluate the system’s compliance with the identified business requirements and assess whether it is acceptable for delivery to the client. Black box testing method is used in this phase.

5.3 Test Plan

A test plan is a document which states the testing scope and process. It contains the functionalities to be tested, testing procedures, test data and the expected result. Following Table 5.3.1 describes the structure which was followed in testing the system.

Test case no.	Test data	Purpose	Expected result	Actual result

Table 5.3.2: Test case structure

A test plan was designed to meet all the requirements stated in the software requirement specification, covering all important aspects of tests that will be conducted on the IMS. The relevant test plan is shown in Appendix A.

5.4 Test Cases

Some of the test cases that are used to test the system are listed below.

Table 5.4.1 displays the test cases used for a user sign in. Several test cases try different entries of data fields.

Test case no.	Test data	Purpose	Expected result	Actual result
1.	Enter a wrong username	Test username	Display error message. User not logged in.	Display error message. User not logged in.
2.	Enter a wrong password	Test password	Display error message. User not logged in.	Display error message. User not logged in.
3.	Click login button with empty username	Test login without an account	Display error message. User not logged in.	Display error message. User not logged in.
4.	Click login button with empty password	Test login without a password	Display error message. User not logged in.	Display error message. User not logged in.
5.	Enter correct username and password	Test correct login to system	User logged in.	User logged in.

Table 5.4.2: Test cases for login panel

Table 5.4.2 shows the test cases used in testing the fund registration scenario.

Test case no.	Test data	Purpose	Expected result	Actual result
1.	Enter an invalid/empty amount	Test saving fund with invalid amount	Display error message. Funds not added.	Display error message. Funds not added.
2.	Click save button without selecting grant	Test grant	Display error message. Funds not added.	Display error message. Funds not added.
3.	Click save button without selecting year	Test year	Display error message. Funds not added.	Display error message. Funds not added.

4.	Click save button after selecting year, grant and inserting valid amount	Test correct funds addition	Funds added.	Funds added.
----	--	-----------------------------	--------------	--------------

Table 5.4.3: Test cases for adding funds

5.5 Evaluation

Evaluation is a process to assess a project’s effectiveness, efficiency, user friendliness, and to understand whether it meets the user requirements or not. Assessment of information is used to provide feedback for the development process. It is the key indicator to measure the degree of success of the system development process, also paving the way to identify new user requirements.

5.6 User Evaluation

After the developer performed the tests, system is finally tested by the end users. This is done using black box testing approach. After this step, the end users accepted the system with few modifications, showing that the system has been accepted with both verification and validation concepts.

5.7 Summary

Testing and evaluation of the system were examined in this chapter since it is a decisive area in measuring any system’s accuracy. A test plan has been defined in order to categorize and understand the important functionalities of the system that has to be tested. Test cases were organized in such a manner to cover the evaluation of all those features. Evaluation is carried out to get actual feedback from the end users of the system.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Chapter 6: Conclusion and Future Work

6.1 Introduction

In this chapter the document will be concluded with a critical evaluation of the system and suggestions for any future work, including a critical discussion and assessment of results of the work carried out. It is important to note that not only failures or shortcomings are included but successes as well.

The study reveals that an inventory management system has a positive impact on the procurement performance measures in the organization. Speed and efficiency are key advantages of automated inventory management. Rather than doing a hand count of inventory, like in the manual process, taking days, the same process can be done in a matter of hours with a computerized inventory management system. Once the inventory management system is in place, administrative staff and clerks can use it to automatically generate all kinds of documents, from purchase orders to good issue notes. Unlike a manual system, in which the data is only as accurate and up to date as the last hand count, the data and reports are more timely in an automated strategy. Altogether these factors engage in reduction in errors and improving employee productivity. User satisfaction is the most favorable impact of an automated system. Comparing to the UCSC supplies division staff's experience of the manual inventory management, this new system has significantly made their task easier.

The system maintains not only the item details; can keep track of grants, suppliers, purchases, good receivals and good issues. Furthermore, this system can be used to generate day- to-day or monthly or annual reports, which are extremely helpful in decision making process.

6.2 Lessons Learnt

The lessons learned throughout the lifecycle of the system represent several project knowledge areas; procurement management, risk management, integration management, quality management, time management, cost management, scope management, human resource management, and communications management.

Among those lessons, most important one was to work according to a time schedule and how to manage the time, taking the maximum use of it.

Flexibility of the design of the software, avoiding implementation of very specific operations concepts into the software, reduces rework when the concepts and requirements are changed. On the other hand clients usually try to add scope to the system. There should be an approval mechanism to changes after requirement analysis is done.

Problems with integrating different components, can be minimized by performing the testing well. It reduces debugging time in a large scale.

Constant user feedback and supervisor's comments from the very beginning of the process are very important for validation and verification.

6.3 Future Improvements

This system can be considered as the first step for designing and developing a supply chain management system which is suitable for the specific procedures which has to be followed by the university supplies division.

As the system is highly flexible, users can generate more reports as and when required.

Future modifications will include online bidding, online purchasing and online registration of suppliers as per the latest requirement.

Also, inventory item issues, fixed assets would be completely maintained by the system.

Grant allocation details should be automatically mailed to the admin users. For this email configuration and servers needed to be implemented.

References

- [1] "Inventory Management" [Online]. Available: <http://www.investopedia.com/terms/i/inventory-management.asp>. [Accessed 22June 2016].
- [2] "Tracking Inventory" [Online]. Available: <https://www.entrepreneur.com/article/21852>. [Accessed 22June 2016].
- [3]"It Enabled Inventory Management System" [Online]. Available: <https://www.scribd.com/doc/33349050/Web-Based-Inventory-management-System>. [Accessed 08August 2016]
- [4] : Puja S.Prasad, Hitesh R.Yerekar, Parag G.Satpute, Gaurav P.Borkar, Ajinkya S. Shendre (2013,January) ERP Sales and Inventory Management System. *International Journal of Soft Computing and Engineering*.
- [5]"What is Case Study?" [Online]. Available: <http://whatis.techtarget.com/definition/case-study>. [Accessed 28August 2016].
- [6] "The software development life cycle (SDLC) Overview" [Online]. Available: http://www.tutorialspoint.com/sdlc/sdlc_overview.htm. [Accessed 20 September 2016].
- [7] "What is Prototype?" [Online]. Available: <http://searchmanufacturingerp.techtarget.com/definition/prototype>. [Accessed 20 September 2016].
- [8] Mohd. Ehmer Khan , Farmeena Khan (March, 2014) Importance of Software Testing in Software Development Life Cycle : *International Journal of Computer Science*.
- [9] Jennifer L. Gasston, A Quality Model for the Evaluation of Software Requirements Specifications, CONSTRUCTING THE INFRASTRUCTURE FOR THE KNOWLEDGE ECONOMY [Online]. pp 281-291. Available: <https://link.springer.com/book/10.1007/978-1-4757-4852-9>. [Accessed 22 September 2016].
- [10] "ISO 9126 Software Quality Characteristics " [Online]. Available: <http://www.sqa.net/iso9126.html>. [Accessed 22 September 2016].
- [11] "Unified Modeling Language (UML)" [Online]. Available: <https://www.techopedia.com/definition/3243/unified-modeling-language-uml>. [Accessed 22 September 2016].

[12]" What Is Agile? What Is Scrum?" [Online]. Available:
<https://www.cprime.com/resources/what-is-agile-what-is-scrum/>. [Accessed 24 September 2016].

[13]"What is Notepad++?" [Online]. Available : <https://notepad-plus-plus.org/>. [Accessed 24 September 2016].

[14] "What is PHP?" [Online]. Available : <http://php.net/manual/en/intro-what-is.php>. [Accessed 29 September 2016].

[15]"Unit Testing Fundamentals" [Online]. Available:
<http://softwaretestingfundamentals.com/unit-testing/>. [Accessed 13October 2016].

[16]"System Testing Fundamentals" [Online]. Available:
<http://softwaretestingfundamentals.com/system-testing/>. [Accessed 15October 2016].

Appendix A: Test Plan

Test No	Purpose	Expected outcome
1	Can registered users access the system with correct authentication details and are any other users rejected?	Registered users should be able to login with accurate authentication details and other users should be blocked.
2	Can users navigate easily through pages?	Users should be able to navigate easily through pages.
3	Can users add data easily?	Users should be able to add items, funds, POs, GRNs, GINs accurately.
4	Can users view data easily?	Users should be able to view inventory, PO, GRN, GIN details accurately.
5	Can users get reports within a short time?	Users should be able to get reports after submitting correct field values for reports.
6	Can users continue even after making a minor error?	System should provide relevant error messages.

Table A.1: Test plan

Appendix B: User Manual

User login

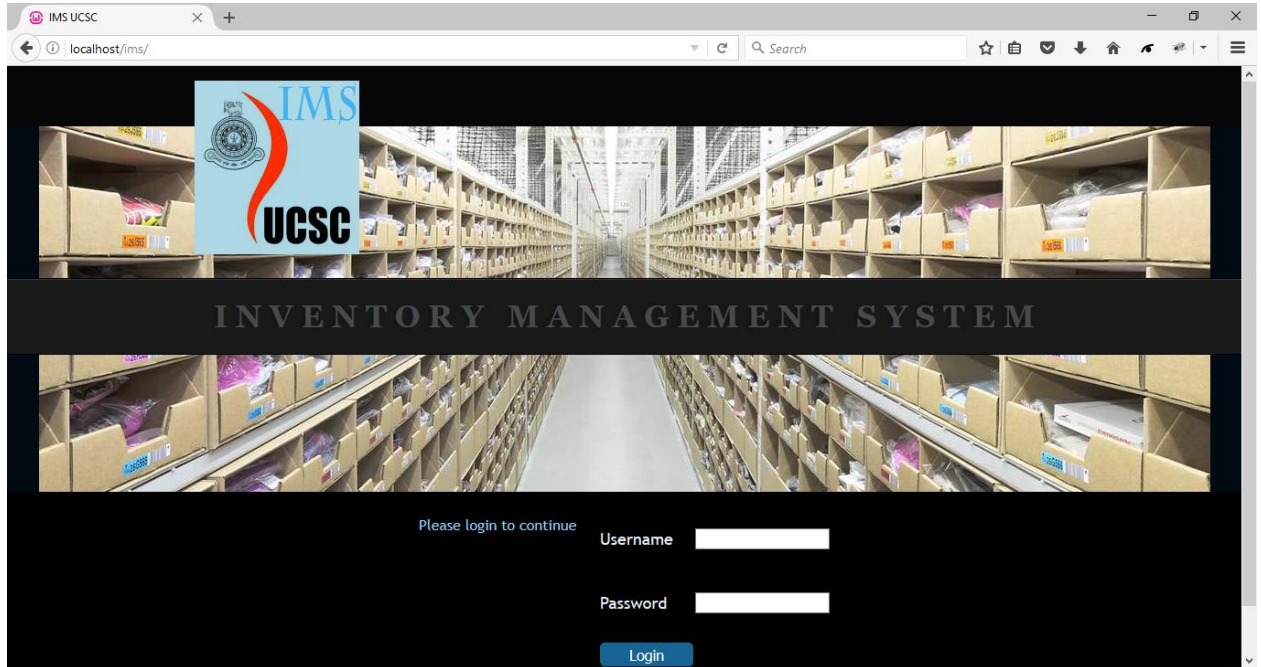


Figure B.1: User login interface

- Type your username and password in the fields.
- Click Login.

Item category registration

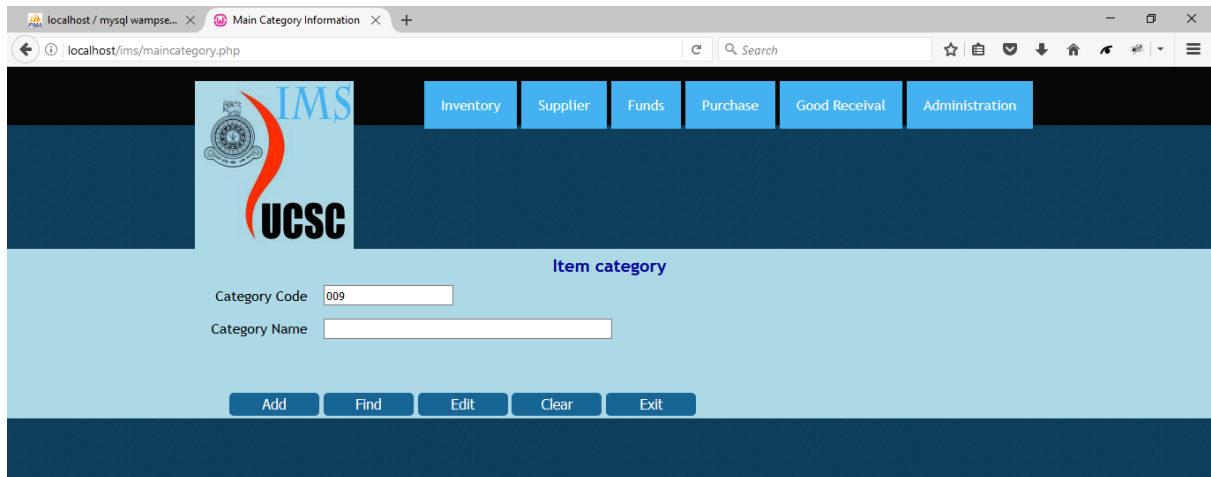


Figure 3.2: Item category interface

- Under 'Inventory' tab group, click 'Category'.
- Type category name to add a new category.
- Click Add.
- To change an existing category name, type category code on 'Category Code' field.
- Click Find.
- Make changes in category name.
- Click Edit.

Item sub category registration

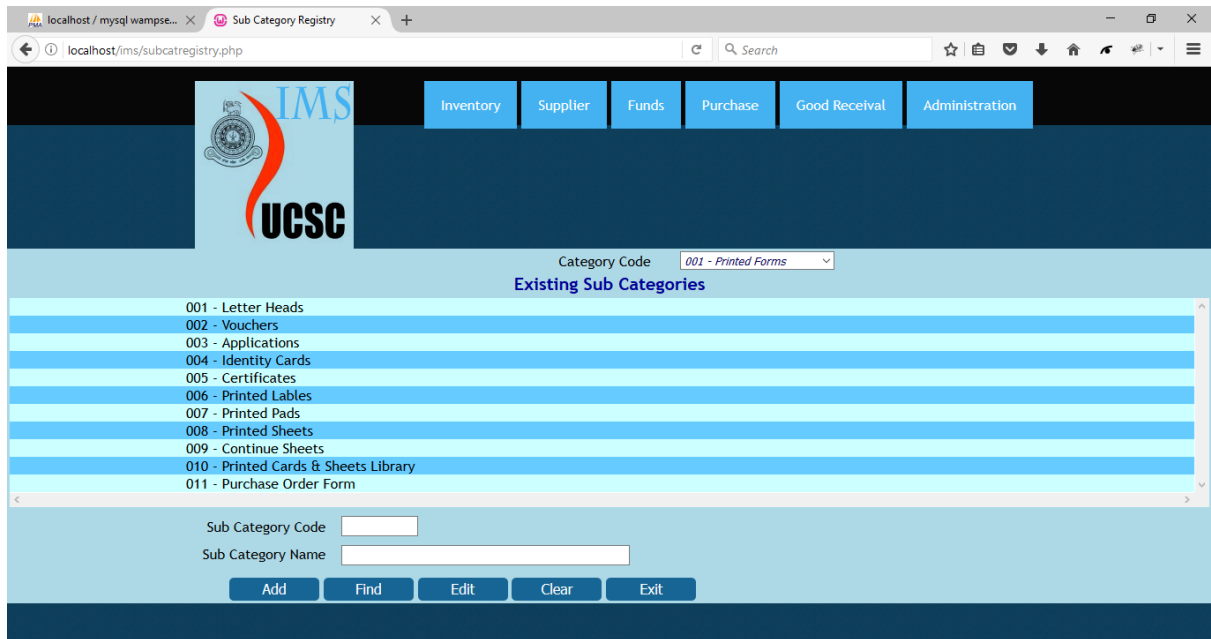
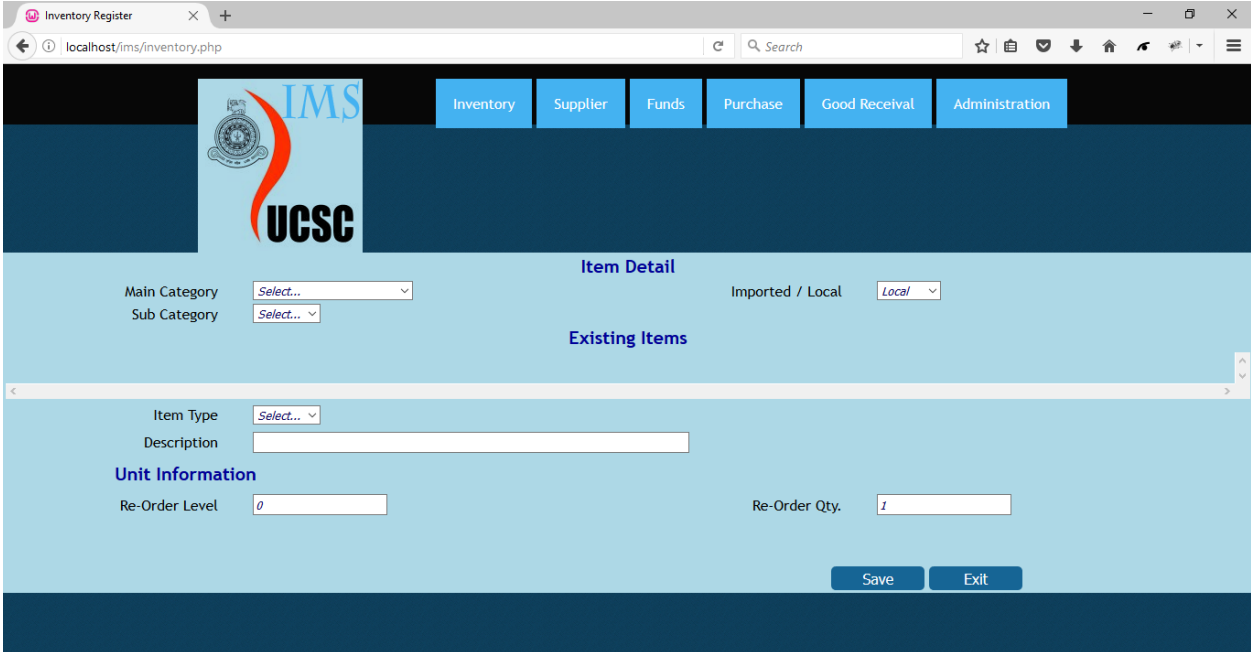


Figure 3.3:Item sub category interface

- Under 'Inventory' tab group, click 'Sub Category'.
- Select category from the dropdown list. Existing sub categories under the category are displayed.
- Type new sub category code in 'Sub Category Code' field.
- Type new sub category name in 'Sub Category Name' field.
- Click Add.
- To change an existing sub category name, type sub category code on 'Sub Category Code' field.
- Click Find.
- Make changes in sub category name.
- Click Edit.

Inventory item registration



The screenshot shows a web browser window titled 'Inventory Register' with the URL 'localhost/ims/inventory.php'. The page features a navigation menu with tabs for 'Inventory', 'Supplier', 'Funds', 'Purchase', 'Good Receival', and 'Administration'. The 'Inventory' tab is active. The main content area is titled 'Item Detail' and includes the following fields and sections:

- Main Category:** A dropdown menu with 'Select...' as the placeholder.
- Sub Category:** A dropdown menu with 'Select...' as the placeholder.
- Imported / Local:** A dropdown menu with 'Local' selected.
- Existing Items:** A section header for a scrollable list of items.
- Item Type:** A dropdown menu with 'Select...' as the placeholder.
- Description:** A text input field.
- Unit Information:**
 - Re-Order Level:** A text input field with the value '0'.
 - Re-Order Qty.:** A text input field with the value '1'.
- Buttons:** 'Save' and 'Exit' buttons at the bottom right.

Figure 3.4: Inventory item user interface

- Under 'Inventory' tab group, click 'Items'.
- Select main category and sub category for the new item from the dropdown lists 'Main Category' and 'Sub Category' respectively. Existing items of the relevant sub category will be displayed. Proceed if the item to be added does not exist under the same sub category.
- Select item type from the dropdown list 'Item Type'.
- Type the name of the new item in the 'Description' field.
- Fill in the fields re-order level and re-order quantity under unit information. Default values are set for these fields. Change the values as per need.
- Click Save.

Supplier category registration

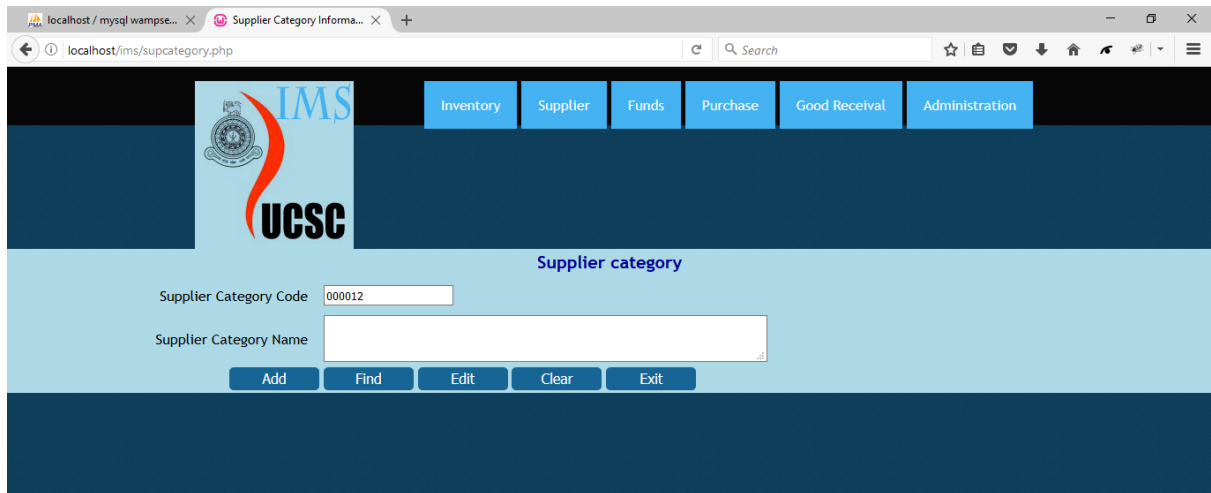


Figure 3.5: Supplier category interface

- Under 'Supplier' tab group, click 'Category'.
- Type new supplier category name to add a new category.
- Click Add.
- To change an existing supplier category name, type supplier category code on 'Supplier Category Code' field.
- Click Find.
- Make changes in supplier category name.
- Click Edit.

Supplier registration.

The screenshot shows a web browser window with the URL `localhost/ms/supplier.php`. The page features the UCSC logo at the top left. Below the logo, there is a 'Financial year' dropdown menu set to 'Select...'. To the right, there is a 'Supplier reg. no:' field. The main content area is titled 'Supplier category(ies)' and contains a list of 11 categories, each with a checkbox: 0001 - Stationery, including photocopy, digital transparency and typing papers etc.; 0002 - Printing/Rubber/Date Stamps, etc.; 0003 - Office and other furniture - wooden; 0004 - Office and other furniture - Steel; 0005 - Fibreglass Products; 0006 - Office equipment/photocopiers, fax machines and accessories, Magi Boards, etc.; 0007 - Tyres, Tubes and Motor Spare Parts; 0008 - Sawn Timber (Local & Imported); 0009 - Laboratory and Medical equipment/Dental equipment; 0010 - Chemicals, Medicines, Pharmaceuticals; 0011 - Test1. Below the categories is the 'Supplier detail' section. It includes radio buttons for 'Foreign/Local' with 'Local' selected. There are text input fields for 'Name', 'Address line 1', 'Address line 2', 'Postal Code', 'Country' (pre-filled with 'Sri Lanka'), and 'Contact I'. On the right side of this section, there are input fields for 'Reference No.', 'VAT Reg. No.', 'Contact II (Foreign/Range)', and 'Fax'.

Figure 3.6: Supplier registry interface

- Under 'Supplier' tab group, click 'Supplier registry'.
- Select year of the entry of supplier from the dropdown list 'Financial Year'.
- Tick the checkboxes of relevant categories for the new supplier. You can select one or more categories.
- Insert supplier's location by clicking the radio button 'Local' or 'Foreign'. 'Local' is default selected.
- Fill the textboxes with relevant details. Name, Address and contact 1 are mandatory fields.
- Click Save.

Fund registration interface

The screenshot shows a web browser window with the URL 'localhost/ims/funds.php'. The page features a dark blue header with a logo on the left and a navigation menu with tabs: 'Inventory', 'Supplier', 'Funds', 'Purchase', 'Administration', and 'Reports'. The 'Funds' tab is active. Below the header, the page title is 'Funds'. There are three input fields: 'Year' (a dropdown menu), 'Grant' (a dropdown menu), and 'Amount' (a text box containing '0.00'). At the bottom of the form are two buttons: 'Save' and 'Exit'.

Figure 3.7: Funds user interface

- Under 'Funds' tab group, click 'Funds registry'.
- Select year of fund from the dropdown list.
- Select the grant you want to add funds to, from the dropdown list 'Grant'.
- Type the amount to be added in 'Amount' field.
- Click Save.

Fund allocation interface

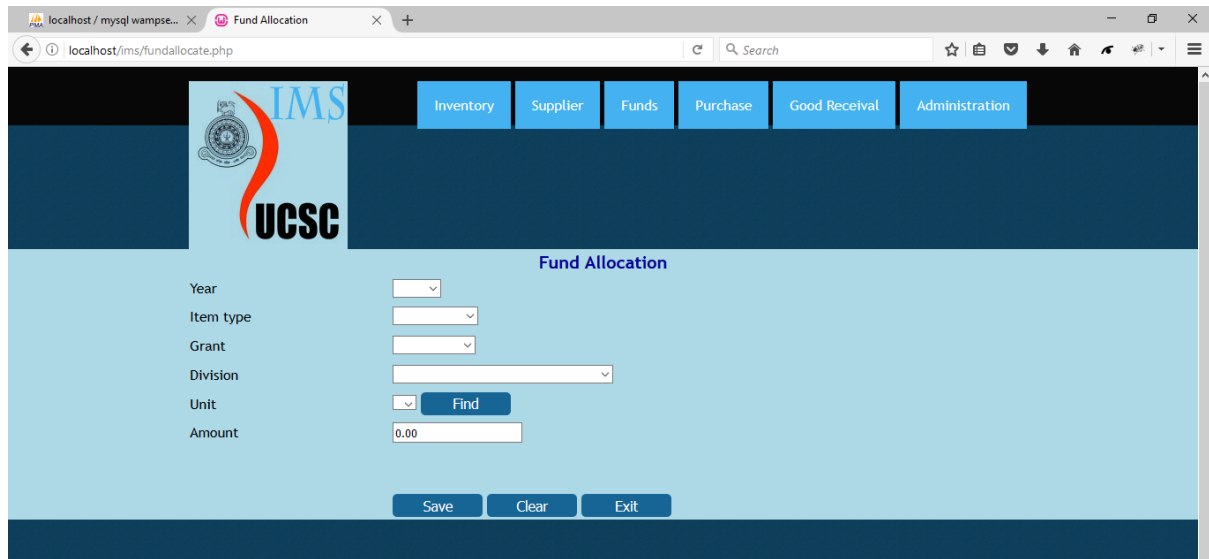


Figure 3.8: Fund allocation user interface

- Under 'Funds' tab group, click 'Allocation'.
- Select year of fund from the dropdown list.
- Select the item type for which funds are to be reserved for, from the dropdown list 'Item Type'.
- Select the grant you want to add funds to, from the dropdown list 'Grant'.
- Select the division of UCSC for which funds are to be reserved for, from the dropdown list 'Division'.
- Select the unit of UCSC for which funds are to be reserved for, from the dropdown list 'Unit'.
- Click Find to know the current amount allocated for the same unit, under the same grant, for the same item type.
- Type the amount to be kept allocated in 'Amount' field.
- Click Save.

Purchase order interface

The screenshot displays the UCSC IMS Purchase Order interface. The browser address bar shows 'localhost/ms/purchase.php'. The interface features a top navigation bar with tabs for 'Inventory', 'Supplier', 'Funds', 'Purchase', 'Good Receipt', and 'Administration'. The main content area is a form for creating a purchase order. It includes fields for 'Date' (2017/03/06), 'Division' (dropdown), 'PO No.' (UCSC/P/ORD/2017), 'Unit.' (dropdown), 'Supplier' (dropdown), 'Address' (text area), 'Item Type' (dropdown), 'Item' (dropdown), 'Ref. No.' (text), 'VAT%' (text), 'Warranty (years)' (text), 'Qty' (text), 'Unit Price' (text), 'Grant' (dropdown), 'Allocation' (dropdown), and a 'Purchase' button. At the bottom, there is a table header with columns: 'Item Code', 'Description', 'Qty', 'Unit Price', 'VAT%', 'Warranty', and 'Value'.

Figure 3.9: PO user interface

- Under 'Purchase' tab group, click 'PO'.
- Select division for which the purchase is made from the dropdown list 'Division'.
- Select unit for which the purchase is made from the dropdown list 'Unit'.
- Select the supplier from whom the purchase will be made, from the dropdown list 'Supplier'.
- Select the item type for which funds are to be reserved for, from the dropdown list 'Item Type'.
- Select the item to be purchased from the dropdown list 'Item'.
- Type university reference number of the item in the field 'Ref.No.' (This is not mandatory).
- Type quantity needed and unit price of the item in the fields 'Qty' and 'Unit Price' respectively. These fields are mandatory.
- Type VAT as a percentage in 'VAT%' field.
- Type warranty period of the item in years in the field 'Warranty'.
- Select the grant from the dropdown list 'Grant'.
- Select the allocation from the dropdown list 'Allocation'.
- Click 'Purchase'.
- Select any more items from the dropdown list 'Item' and add to the purchase order in the same manner.

- After adding the required items to the purchase order, click Save.
- To print an order, type order number in 'PO no.' field.
- Click Find.
- Click Print.
- To cancel a previously made purchase, type order number in 'PO no.' field.
- Click Find.
- Click Cancel Order.

Good receipt note

Good Receipt Note

localhost/ims/GRN.php

IMS UCSC

Inventory Supplier Funds Purchase Good Receipt Administration

GRN No. 2017/000003 Date 05/03/2017

Supplier Year Select... Supplier Select...

Reference PO no. Select... Invoice no.

Item Code & Description	Quantity				Price				
	Received	Ordered	Rec. Already	Balance	PO_Unit Price	Edited_Unit Price	VAT(%)	Tax Per Unit	Value
								Grand Total	

Save Clear Exit

Figure 3.10: GRN interface

- Select the registered year of the supplier for whom the GRN will be made for, from the dropdown list 'Supplier Year'.
- Select the relevant supplier's name from the dropdown list 'Supplier'.
- Select the purchase order number from which the items were ordered from the dropdown list 'PO no.'.
- Type reference, remarks for the GRN, invoice number of the supplier's invoice in the fields 'Reference', 'Remark', 'Invoice no.' fields respectively. (Remark is mandatory).
- Click Save.

Report dashboard

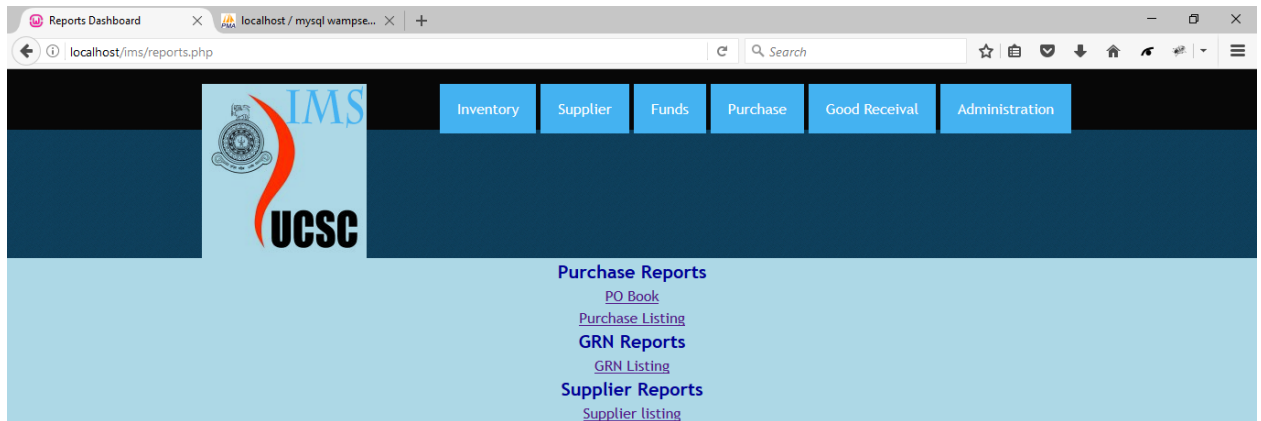


Figure 3.11: Report dashboard

- To generate reports, go to 'Reports' menu in 'Administration' tab group.
- Click on the report type you want to generate.
- A new interface will appear.

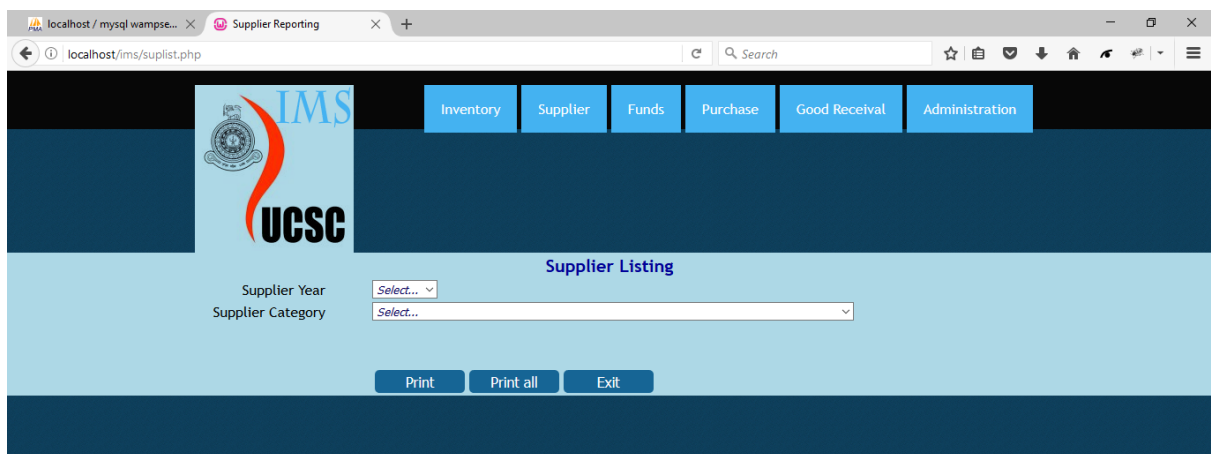


Figure 3.12: Report generation tray

- Select the values from the dropdown lists as per the need.
- Click relevant Print button. This may vary according to the report type.

User registration

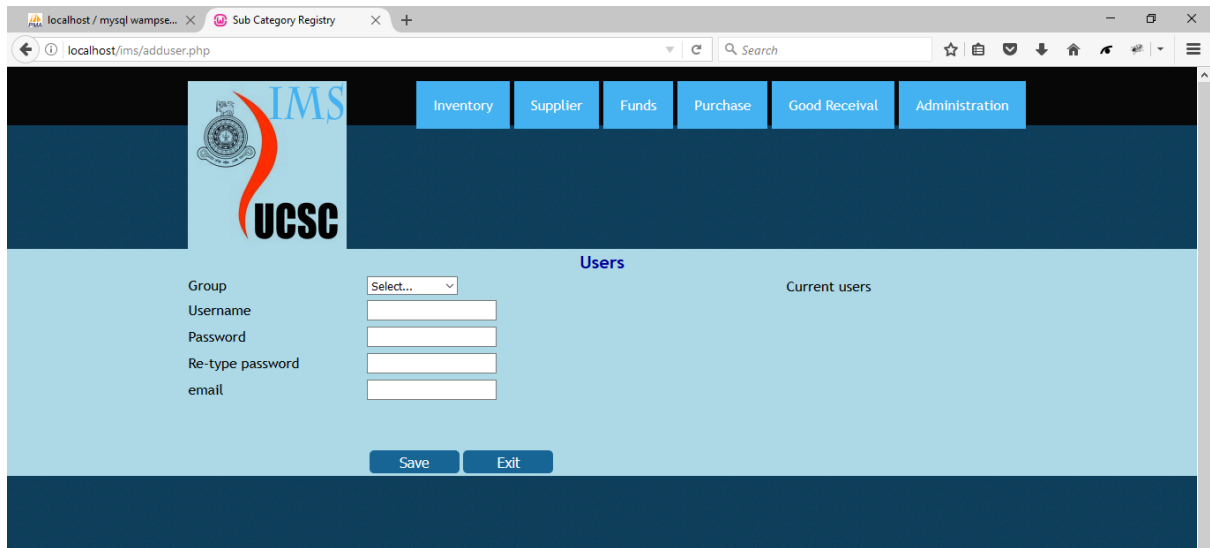


Figure 3.13: User registration interface

- Select the type of user to be added from the dropdown list 'Group'. Existing users of that type will appear in the right side.
- Type username and password in the relevant fields. Make sure the username does not exist.
- Re-type the given password in the 'Re-type password' field.
- Type email address of the user in the field 'email'.
- Click Save.

Appendix C: Test Cases

Test cases for inventory item registration

Test case no.	Test data	Purpose	Expected result	Actual result
1.	Click save button without selecting main category	Test main category field	Display error message. Item not added.	Display error message. Item not added.
2.	Click save button without selecting sub category	Test sub category field	Display error message. Item not added.	Display error message. Item not added.
3.	Click save button without selecting item type	Test item type field	Display error message. Item not added.	Display error message. Item not added.
4.	Click save button without typing item description	Test description field	Display error message. Item not added.	Display error message. Item not added.
5.	Click save button after selecting main category, sub category, item type and inserting description	Test correct item addition	Item added.	Item added.

Table C.1: Test cases for inventory user interface

Test cases for category addition

Test case no.	Test data	Purpose	Expected result	Actual result
1.	Click save button without typing category name	Test category description field	Display error message. Category not added.	Display error message. Category not added.
2.	Click save button after typing category name	Test correct category addition	Category added.	Category added.

Table C.2: Test cases for category user interface

Test cases for sub category addition

Test case no.	Test data	Purpose	Expected result	Actual result
1.	Click save button without typing sub category code	Test sub category code field	Display error message. Sub category not added.	Display error message. Sub category not added.
2.	Click save button without typing sub category name	Test sub category name field	Display error message. Sub category not added.	Display error message. Sub category not added.
3.	Click save button giving invalid sub category code	Test sub category code field	Display error message. Sub category not added.	Display error message. Sub category not added.
4.	Click save button after typing valid code and name	Test correct sub category addition	Sub category added.	Sub category added.

Table C.3: Test cases for sub category user interface

Test cases for PO addition

Test case no.	Test data	Purpose	Expected result	Actual result
1.	Click save button without selecting division	Test save button	Save button not activated.	Save button not activated.
2.	Click find button without giving valid PO number	Test find button	Display error message.	Display error message.
3.	Click Purchase button without typing quantity	Test quantity field	Display error message. Purchase not added.	Display error message. Purchase not added.
4.	Click Purchase button without typing unit price	Test unit price field	Display error message. Purchase not added.	Display error message. Purchase not added.
5.	Click Purchase button giving invalid quantity	Test quantity field	Display error message. Purchase not added.	Display error message. Purchase not added.
6.	Click Purchase button giving invalid unit price	Test unit price field	Display error message. Purchase not added.	Display error message. Purchase not added.
7.	Click Purchase button giving invalid VAT	Test VAT field	Display error message. Purchase not added.	Display error message. Purchase not added.
8.	Click Purchase button giving invalid warranty period	Test warranty period field	Display error message. Purchase not added.	Display error message. Purchase not added.
9.	Click Purchase button after giving valid inputs	Test Purchase button	Purchase added.	Purchase added.

10.	Click save button after adding purchases	Test save button	PO added.	PO added.
11.	Click print button after giving valid PO number	Test print button	PO pdf displayed.	PO pdf displayed.

Table C.4: Test cases for PO user interface

Test cases for funds addition

Test case no.	Test data	Purpose	Expected result	Actual result
1.	Enter an invalid/empty amount	Test saving fund with invalid amount	Display error message. Funds not added.	Display error message. Funds not added.
2.	Click save button without selecting year	Test year field	Display error message. Funds not added.	Display error message. Funds not added.
3.	Click save button without selecting grant	Test year	Display error message. Funds not added.	Display error message. Funds not added.
4.	Click save button after selecting year, grant and inserting valid amount	Test correct funds addition	Funds added.	Funds added.

Table C.5: Test cases for funds interface