

COM814: Project 2015 - 2016

# Dissertation

School of Computing & Information Engineering

Inventory Management System [Inventory Manager]

B00668298      Abiodun Oke

Supervisor:      Dr Gaye Lightbody

Second Marker: Dr Pat Corr

1 September 2016



## **Plagiarism Statement**

I declare that this is all my own work and does not contain unreferenced material copied from any other source. I have read the University's policy on plagiarism and understand the definition of plagiarism. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file.

## **Acknowledgements**

I would like to thank my supervisors and second marker Giuseppe Trombino, Gaye Lightbody and Pat Corr, for their guidance, assistance and feedback given throughout the project. Also thanks to tutors on Udemy and YouTube for all their help and support.

Thanks to all the lecturers, students and staffs who have kindly participated in my research study and website testing, the feedback received from these participant was instrumental in successful completion of the project.

Finally, I would like to thank my family and friends for their continuous love and support throughout the project and the MSc Software Development course.

# Contents

<b>Abstract .....</b>	<b>11</b>
<b>Introduction .....</b>	<b>12</b>
1.1 Introduction.....	12
1.2 Problem Statement .....	13
1.3 Aim.....	13
1.4 Objectives .....	13
1.5 Requirements for Development.....	14
1.6 Dissertation Outline.....	15
<b>Analysis .....</b>	<b>16</b>
2.1 Introduction.....	16
2.2 The Current Situation .....	16
2.2.1 Key findings .....	18
2.3 Recommendations.....	19
2.4 Summary.....	19
<b>Requirements Analysis .....</b>	<b>20</b>
3.1 Introduction.....	20
3.2 Requirements .....	20
3.2.1 Focus Group Analysis.....	20
3.2.2 Key Findings.....	21
3.2.3 User Stories .....	22
3.3 Software Requirements.....	22
3.3.1 System Overview .....	22
3.3.2 Assumptions and Constraints.....	23
3.4 Non-Functional Requirements .....	23
3.5 Functional Requirements .....	23
3.5.1 Processing Users.....	23
3.5.2 Processing Assets/Items .....	24

3.5.3 Processing Asset/Item Category.....	24
3.5.4 Processing Borrowed and Returned Items.....	25
3.6 User Permissions .....	25
3.6.1 Admin/ Sub-Admin .....	25
3.6.2 Academics.....	26
3.6.3 Student .....	26
3.7 Summary.....	26
<b>Design.....</b>	<b>27</b>
4.1 Introduction.....	27
4.2 Application Architecture .....	27
4.3 Initial Design .....	30
4.4 User Interface Design .....	31
4.4.1 Storyboard.....	32
4.5 Database Design .....	33
4.6 Table Structure .....	33
4.7 Summary.....	34
<b>Implementation .....</b>	<b>35</b>
5.1 Introduction.....	35
5.2 The Prototype.....	35
5.3 Development Methodology .....	36
5.4 Platform Implementation .....	38
5.4.1 Resource Implementation.....	39
5.5 Database Implementation.....	39
5.6 Implementation of Front-End (HTML, CSS and JS) .....	40
5.7 Core Functionalities.....	41
5.7.1 Issue/Borrow Item.....	41
5.7.2 Import CSV file.....	42
5.8 Summary.....	42

<b>Testing and Evaluation .....</b>	<b>43</b>
6.1 Introduction.....	43
6.2 Functional Testing .....	43
6.3 Non-Functional Testing .....	43
6.4 Focus Group Testing .....	44
6.5 Summary.....	45
<b>Conclusion and Recommendations .....</b>	<b>46</b>
7.1 Introduction.....	46
7.2 Summary of the Project Objectives.....	46
7.3 Results Assessment .....	48
7.4 Experience gained and Issues encountered .....	49
7.4.1 Experience Gained.....	49
7.4.2 Issues Encountered .....	49
7.5 Recommendations.....	49
7.6 Summary.....	50
References.....	51
<b>Appendices.....</b>	<b>53</b>
Appendix One: Rich Pictures Outlining Current Problem and Proposed Solution .....	53
1.1: Current Problem.....	53
1.2: Proposed Solution .....	54
Appendix Two: Example of Completed Research Questionnaire.....	55
2.1: Example Questionnaire .....	55
Appendix Three: Wireframe Design .....	60
3.1: Login page .....	60
3.2: User's Profile page.....	60
3.3: Manage Items Page .....	61
3.4: Report Page .....	61
3.5: Manage Users Page .....	62

3.6: Change Password Page.....	62
3.7: Forgotten Password Page.....	63
3.8: Sign up Page .....	63
Appendix Four: Storyboard .....	64
4.1: Example storyboard of Admin dashboard.....	64
Appendix Five: Database Tables.....	65
5.1: Users Table .....	65
5.2: Item Categories Table.....	65
5.3: Items Table .....	65
5.4: Issued Table .....	66
5.5: Users online Table .....	66
5.6: Returned Table .....	66
Appendix Six: Script to prevent item on loan from being deleted.....	68
Appendix Seven: Script to prevent duplicate item import into database.....	69
Appendix Eight: Testing.....	70
8.1: Example Test Case (complete test case attached on separate document) .....	70
8.2: Functional Testing Results.....	70
8.3: Example User Testing Questionnaire .....	71
Appendix Nine: User Manual .....	82
Quick Start .....	<b>Error! Bookmark not defined.</b>
Demo .....	83
Screenshots & Description .....	84
Homepage .....	84
Login Page.....	84
Admin Dashboard.....	85
Manage Users.....	86
Manage Items.....	86
Issue Item .....	87

Item Category .....	88
Print Barcode .....	88
Circulation .....	89
Issue History .....	89
Return History .....	89
Generate Report.....	90
Profile .....	90
Support multiple device screens .....	91
Appendix Ten: Test Cases.....	92

## List of Figures

Figure 1: User Stories from Research Questionnaire .....	22
Figure 2: Site Map for 'IMS   Inventory Manager' .....	29
Figure 3: Wireframe Design of the Homepage.....	30
Figure 4: Wireframe Design of the Item page .....	31
Figure 5: Example Storyboard of the Login process .....	32
Figure 6: Entity Relationship Diagram (ERD) .....	34
Figure 7: Prototype of UI Validation.....	35
Figure 8: Prototype Admin example -"All Items" .....	36
Figure 9: Personal Development process used (Agile Model) .....	37
Figure 10: Example of Sprint Backlog used in application development .....	38
Figure 11: Connecting to MySQL database (db. php).....	39
Figure 12: JavaScript form validation (client-side) .....	40
Figure 13: Example php form validation (server-side) .....	41
Figure 14: User Guide starts .....	82
Figure 15: Homepage .....	84
Figure 16: Login Page.....	84
Figure 17: Admin dashboard .....	85
Figure 18: Manage Users.....	86
Figure 19: Manage Items.....	87
Figure 20: Issue Item .....	87
Figure 21: Item Category .....	88
Figure 22: Print Barcode .....	88
Figure 23: Issue History .....	89
Figure 24: Return History .....	89
Figure 25: Generate Report.....	90
Figure 26: Profile .....	90
Figure 27: Support multiple device screen.....	91



## List of Tables

Table 1: Review of available IMS .....	17
Table 2: Key Findings of Research Study .....	21
Table 3: User Defined Testing Results .....	45
Table 4: Admin Menu .....	85
Table 5: Test Case ID: IMS_001 .....	93
Table 6: Test Case ID: IMS_002 .....	94
Table 7: Test Case ID: IMS_003 .....	95
Table 8: Test Case ID: IMS_004 .....	96
Table 9: Test Case ID: IMS_005 .....	97
Table 10: Test Case ID: IMS_006 .....	98
Table 11: Test Case ID: IMS_007 .....	99
Table 12: Test Case ID: IMS_008 .....	100
Table 13: Test Case ID: IMS_009 .....	101
Table 14: Test Case ID: IMS_010 .....	102
Table 15: Test Case ID: IMS_011 .....	103
Table 16: Test Case ID: IMS_012 .....	104
Table 17: Test Case ID: IMS_013 .....	105
Table 18: Test Case ID: IMS_014 .....	106
Table 19: Test Case ID: IMS_015 .....	107
Table 20: Test Case ID: IMS_016 .....	108
Table 21: Test Case ID: IMS_017 .....	109
Table 22: Test Case ID: IMS_018 .....	110
Table 23: Test Case ID: IMS_019 .....	111
Table 24: Test Case ID: IMS_020 .....	112
Table 25: Test Case ID: IMS_021 .....	113
Table 26: Test Case ID: IMS_022 .....	114
Table 27: Test Case ID: IMS_023 .....	115

Table 28: Test Case ID: IMS_024 .....	116
Table 29: Test Case ID: IMS_025 .....	117
Table 30: Test Case ID: IMS_026 .....	118
Table 31: Test Case ID: IMS_027 .....	119
Table 32: Test Case ID: IMS_028 .....	120
Table 33: Test Case ID: IMS_029 .....	121
Table 34: Test Case ID: IMS_030 .....	122
Table 35: Functional Testing Results .....	123
Table 36: Compatibility Testing (Hardware).....	124
Table 37: Compatibility Testing (Browser) .....	124
Table 38: Compatibility Testing (Operating Systems) .....	124

## Abstract

Inventory Management is one of the basic problems in almost every organisation. Before the computer generation, paperwork solutions were being used as inventory management tools which was very far from being a solution. With the advancement in technology and the need to address a decline in the number and skills of students studying Computer Science, smaller devices such as Raspberry Pi are used as a teaching and learning tools in educational institutions. Hence, a number of Raspberry Pi development kits were acquired by the University which are used in the laboratory sessions and borrowed by both staffs and students. As a result, the current management of these assets has become highly labour-intensive to operate and requires continuous monitoring to ensure that each development kit is accounted for.

This dissertation focuses on the development of an Inventory Management System (IMS) for tracking Raspberry Pi kits and other assets with all the necessary functionality. The IMS will be able to track all assets within the department over a period of time. Tracking the usage of assets using IMS will create more efficient and effective use of staff resources.

“Inventory Manager” was developed as a cross-browser and cross platform system using the current guidelines and standards of good practice in web accessibility and usability to provide a positive experience for users. Along with the stakeholder’s requirements, a research questionnaire was administered to a focus group of both lecturers and students to gain design and user requirements used to develop and implement a prototype. Manual testing was performed according to the ISTQB standards and further system testing by administering a second questionnaire to some members of the focus group who expressed agreement in testing the system. Comments and feedback gained were documented and recommendations for future development were made.

**Keywords:** IMS, Inventory Manager, Raspberry Pi, assets, cross-browser, cross-platform.

# Introduction

## 1.1 Introduction

The advancement in technology and growth in the use of smaller devices have opened up new doors for product improvement. This has led to both public and private sectors creating awareness and promoting the use of Raspberry Pi as a teaching and learning tool in the study of computer science in various institutions across the world. According to Ranjit Atwal, a research director at market analysis firm Gartner, “these small devices had emerged because of the massive changes in the way people used computers” (Ward, 2013). Since it was launched in February 2012, more than 7 million have been sold around the world, a landmark considering the group of scientist behind it thought they might sell 1,000 at most. The idea of using Raspberry Pi as an educational tool is to encourage pre-university learners to engage with programming, and to develop an interest in and understanding of programming and computer science concepts. From a wider perspective, the idea addresses a general decline in the number and skills of students applying for Computer Science classes and encourage enthusiastic programming language teachers.

It is essential to acknowledge the importance of computer science as a profession and the way universities prepare people who enrolled in it. In line with this concept, a number of Raspberry Pi development kits were acquired by the University to be used in the laboratory sessions and borrowed by both staffs and students to use on projects. As a result, the current management of these assets has become highly labour-intensive to operate. They require continuous monitoring to ensure that each development kit is accounted for and the time spent processing and monitoring these kits could be used on more productive activities for students.

Through the implementation of Raspberry Pi into the laboratory sessions, allows students to learn through interactivity and participation. But most importantly this allows the lecturer an additional resource to prepare the students to be better programmers. Currently at the University of Ulster, the management of these devices have been difficult due to not having a proper inventory management system.

An opportunity presented itself within the School of Computer and Mathematics by showing that there was a developing need for an Inventory Management System. The “Inventory Manager” application was designed as a solution to the vast amount of time required for a lecturer to devote to managing these devices. This application primarily would allow them to decrease time spent in manual inventory management and also allow the lectures to track the usage and keep inventory of these kits.

## 1.2 Problem Statement

“To help lecturers and students of the School of Computing and Mathematics, it would be highly beneficial to have a website which is easily accessible from any device that will provide a dynamic inventory management and borrowing system”.

## 1.3 Aim

The aim of the project is to develop a website with the ability to run from multiple devices and allow easy inventory management of the Raspberry Pi development kits and other assets. This website will include a range of essential features that will make it easier for individuals to use and easy to adapt by other Schools within Ulster University with similar inventory management requirements.

## 1.4 Objectives

In order to achieve the aforementioned aim, it was necessary that a clear set of objectives were outlined and established:

- To conduct background research on the problem and demonstrate why this website constitutes an adequate solution.
- To examine and investigate the current market and analyse available inventory management system websites. Content of these applications will be evaluated as well as the overall layout and design. These details provide a good basis on which to design a prototype and further develop the final application.
- To compose and administer a research questionnaire to a group of lecturers and students in order to establish the key elements users would expect in terms of design, features, and content. The feedback from questionnaires will be fundamental in the overall design of the website, both in terms of understanding essential content and potentially introducing new ideas to enhance its functionality.
- To develop a simple user interface, containing clearly defined sections which were easy to navigate, usually attractive and provided clear, concise information. This will consider the content and layout of available applications and build on these strengths to develop an effective website.
- To evaluate the website prior to testing and as a result provide recommendations for additional features or modifications.

- To create and administer a scenario based testing questionnaire to a group of individuals who have expressed agreement to test the system in the research questionnaire. This will ensure efficiency is of a high level and the project aim is being achieved.

### 1.5 Requirements for Development

The application was developed using existing hardware and software which is readily available at little or no cost in order to complete the aim and objectives as described above. The development also required the use and understanding of various programming languages such as HTML, PHP, MySQL and JavaScript with access to secure server and a database. In addition, the following assets were also required for a successful development and implementation of the application.

- A laptop computer capable of running a development package for designing and creating websites.
- Front-End framework and PHP framework for Back-End development.
- A web server package.
- A tablet which will allow the system to be tested in a live environment in order to evaluate mobile capabilities.

## 1.6 Dissertation Outline

**Chapter 2 – Analysis** – This chapter will consider the current market, discussing the main issues as well as ideas which will be developed in the proposed application. A literature review will be included to identify the problem and proposed solution.

**Chapter 3 – Requirements Analysis** – This chapter will discuss the methodology used in the development of the application. Stakeholder's requirements and focus group research questionnaire identifies the end-user requirements for the application.

**Chapter 4 – Design** – The focus of this chapter will be the User Interface (UI) and System design. A detailed overview of the designs for the website, database schemas and the relational diagrams.

**Chapter 5 – Implementation** – This chapter details the implementation of the designed solution discussing the development process and the problems encountered when accomplishing objectives and how said problems were overcome for the final version of the website.

**Chapter 6 – Testing and Evaluation** – This chapter will discuss the testing of the application which was carried out using functional testing standards and focus groups questionnaires in conjunction with defined user testing.

**Chapter 7 – Conclusion & Recommendations** – This chapter concludes the dissertation with an overall view of the final system with the inclusion of a review of the work conducted. It analyses whether the system was successful in meeting the core aim and key objectives discussed in the introduction, with the inclusion of recommendations for future development of the system.

# Analysis

## 2.1 Introduction

This chapter provides a summary of the analysis in relation to the proposed application. It is important to acknowledge the current situation and availability of products in today's market. In doing so, ideas can be developed and key requirements for the website can be measured and prioritised. Reviewing the current situation and analysing the need for specific requirements will form the basis on which the website will be developed. The initial stages of this review involved compiling rich picture diagrams to outline both the current situation and the later proposed solution. The illustrations of both the current situation and proposed solution can be found in **Appendix One: 1.1 and 1.2**.

The review will then look in more details at the current problem probable in Inventory Management. In relation to the current problem, a number of available software have been analysed. These will be discussed both within the body of the review as well their effectiveness. Advantages and disadvantages of content and design will also be discussed and evaluated which will allow for ideas to be considered in the development of the new website.

## 2.2 The Current Situation

The existing system which is available as a mobile app is limited in scope as it does not support multiple users. This is due to the fact that the app is available only on a single mobile device accessible by a single user over a period of time. Hence, asset tracking and monitoring is difficult as the access to the location of the assets is not restricted but available to other lecturers who make use of the Raspberry Pi kits as a teaching resource for their module. Therefore, it is difficult to establish asset availability since not all the lectures that use the resources have access to the software that controls its circulation. The only advantage currently from the app is the use of a centralised database for managing assets and student records when mobile device on which the app is installed is available.

Through research, it is evident that there is a lack of application available to serve the inventory management needs of the department. This may be due to the level of difficulty combining inventory management with borrowing features which will meet the inventory management need of the department. It may also be the fact that many inventory management software is designed mainly for commercial purposes to be used by business to control and manage their price lists, product variants and quantities.

Further review was conducted on five existing inventory software among many available in the market designed to meet the inventory management needs of any institution. This review was conducted to look at the services and features offered in more details. The software discussed include: tradegecko,



inflow, iMagic Inventory, Inventory Tracker Plus, and Stitch. The main aspects of the software that were reviewed includes:

- **Pricing**
- **Ease of Use**
- **Inventory Feature**
  - Barcode System
- **Administrative Features**
  - Client Database
  - Assign level of Access Users
  - Customizable Reports
- **Loan Feature**
  - Issue and Return of assets
- **Platforms**
  - Online
  - Mobile
  - On Premise (Installation)

The result of this review are presented in Table 1 below:

	tradegecko	inFlow	iMagic Inventory	Inventory Tracker	Stitch
<b>Pricing</b>	\$49 - \$349/month	Free - \$799/one-time	From \$249/month	\$199/one-time	\$29 - \$449/month
<b>Ease of Use</b>	Easy	Easy	Easy	Easy	Very Easy
<b>Inventory Feature</b>					
Barcode System	✓	✓	✓	✓	✓
<b>Administrative Features</b>					
Client Database	✓	✓	✓	✓	✓
Assign level of access	✓	✓	✓	✓	✓
Customizable reports	✓	✓	✓	✓	✓
<b>Loan Feature</b>					
Issue & Returns	✗	✗	✗	✗	✗
<b>Platforms</b>					
Online	✓	✓	✗	✗	✓
Mobile	✓	✗	✗	✗	✗
On Premise (installation)	✗	✓	✓	✓	✗
{Best inventory management software of 2016 (Top Ten Reviews, 2016)}					

Table 1: Review of available IMS

### 2.2.1 Key findings

The key findings of this review confirms the difficulty in adapting existing software in the market with the University system.

1. None of the software reviewed are open source and therefore only available at a price quote which covers software licence, maintenance, support, customisation and installation for premise based. Although, the purchase of software for use on University owned equipment is carried out via the departmental software co-ordinator, none of the software reviewed is capable of satisfying the specific needs of the department. It may be a good idea to develop and create a software which will meet the inventory needs as described by the stakeholder representing the School.
2. According to the review, most of the software are categorised as easy to use with Stitch being very easy to use in terms of User Interface, clarity, navigation and contents on the application are relevant. This is a technical aspect which needs to be considered in the proposed website to enhance the quality of the User Interface.
3. In terms of accessibility and usability, all the applications have appropriate font type and the overall presentation is professional and consistent.
4. In the review, the most common access platform for majority of the software is through on premise installation. Only a few such as tradegecko and inflow could be accessed through online or mobile.
5. The features of the applications reviewed were classified into three categories: Administrative, Inventory and Loans.
  1. The **Administrative features** for all the software reviewed have all required functions to maintain the database of both client and assets. Each software has the ability to assign access according to user level for security, and can customise report according to user's needs.
  2. The **Inventory features** are similar for all the software reviewed with the ability to implement a barcode system of inventory management and tracking.
  3. In relation to **the issue and return of the assets** for loan purposes, none of the software reviewed were configured with such protocol.

By establishing the problem and conducting research into the current applications already available in the market, it is clear that partial solutions are currently available. However, none of these would completely alleviate the problem as most solutions focus more on commercial inventory management protocols making its adaptation difficult for educational establishments.

## 2.3 Recommendations

The following key points should be taken into consideration when designing the new website:

- The website should be well presented with a professional appearance containing text that is clearly legible and should require minimal scrolling.
- The website should offer user account for lecturers and students in which to access information required. It should have consistent navigation based on its features and provide information on help and support for users.
- The website should be efficient and compatible with all major browsers and there should be uniformity in the process of information regardless of the browser type or devices used.
- Every component of the website should work quickly and correctly. It should work as expected including hyperlinks, registration forms, and navigation buttons.

## 2.4 Summary

This chapter has presented the background research on the problem, the available solutions through review of the current inventory software in the market and demonstrate why the development of an Inventory Management System particular to the needs of the School of Computer and Mathematics constitutes an appropriate solution. The knowledge and insight gained from this research allowed the author to design and develop an application that would assist in the delivery of a better inventory management of assets within the School.

# Requirements Analysis

## 3.1 Introduction

This chapter identifies the requirements of the end-user of the proposed website through discussions with the primary stakeholder and the use of a questionnaire which was administered to a target group of lecturers and students within the University community. The chapter includes a discussion and analysis of the answers and recommendations given by the group with regards to the websites content, features and requirements. The chapter concludes with the detailing of functional and non-functional requirements to be included in the applications design.

## 3.2 Requirements

From the initial conception, it was proposed that the system would be designed as a web based application compatible with multiple browsers and have cross-platform mobile capabilities. This is to ensure individuals that are authorised to access the application could do this at their own convenience at any particular period of time. Extensive research into applications currently available in the market also confirm easy access to the application as an important part of the decision process in the design and development of an application that meets usability guideline of a good website.

Developing the application as a web based will be the solution to current limitations experienced by individuals that do not have access to the app currently use for inventory management of assets and the available information will be unified as the process is performed in real time enhancing the overall experience of end-users.

### 3.2.1 Focus Group Analysis

By analysing applications currently available in the market facilitated the basis on which research questionnaire was created. This questionnaire consisted of 10 questions (See Appendix Two) and copies were distributed to academic/non-academic staffs and students. The aim of the questionnaires was to gain insight into people's understanding of how asset is being managed within the University community. In addition to this the questionnaires were also used in order to determine preferences that individuals may have in relation to the applications design in the following areas: content, layout, features and presentation.

A total of 20 questionnaires was distributed out of which 15 were received back from the focus group. The 15 questionnaires received were completed successfully corresponding to a 75% response rate which provides an adequate number of answers the research study required to be successful. In addition to this 47% of the respondents agreed to test the first version of the system which was essential in terms of gaining feedback to be considered when updating the website to its final version.

### 3.2.2 Key Findings

The results of the research study were analysed in order to determine the websites features and requirements as specified by lecturers and students. The key findings have been assembled and presented in Table 2.

Q1	20 questionnaires were given out and 15 were received. 33.5% are both academic and non-academic while 66.5% are students. This represents 75% of the total given out.
Q2	20% of respondents are aware of the top 3 inventory software but none of the respondent ever used any of the inventory software.
Q3	100% of respondents agree that teaching staffs and students would benefit from using the inventory website
Q4	According to the respondents, 86.5% rate all the features as very suitable for the website.
Q5	86.5% of respondents agree that all the standard and management features are very important for the website.
Q6	<p>According to the respondents, 93% found the Structure B to be very suitable for the website.</p> <pre> graph TD     Login[Login] --&gt; Student[Student/ Staff]     Login --&gt; Admin[Admin]     Student --&gt; Borrowed[borrowed item]     Student --&gt; Change[change / update details]     Admin --&gt; Raspberry[Raspberry Pi Database]     Admin --&gt; StudentDB[Student Database]     Admin --&gt; Report[Generate Report]     Raspberry --&gt; Search1[Search]     Raspberry --&gt; Add1[Add]     Raspberry --&gt; Delete1[Delete]     StudentDB --&gt; Search2[Search]     StudentDB --&gt; Add2[Add]     StudentDB --&gt; Delete2[Delete]     Report --&gt; Barcode[Generate Barcode] </pre> <p>Structure B: Very suitable</p>
Q7	All proposed features were found to be suitable by the majority of the respondents apart from the "Discussion board" with 86% finding this feature neither suitable nor unsuitable.
Q8	A tie between Century Gothic (Option 1) and Times New Roman (Option 2) with 40% of respondents found each to be suitable font family for the website.
Q9	According to the respondent, 73% found the Raspberry Pi image and Raspberry Pi kit image to be suitable for the website.
Q10	47% representing 7 individuals out of the respondent are willing to test the website when available.

Table 2: Key Findings of Research Study

### 3.2.3 User Stories

The websites feature and requirements which questionnaire respondents have recommended have been assembled and grouped into user stories which are presented below in Figure 1.

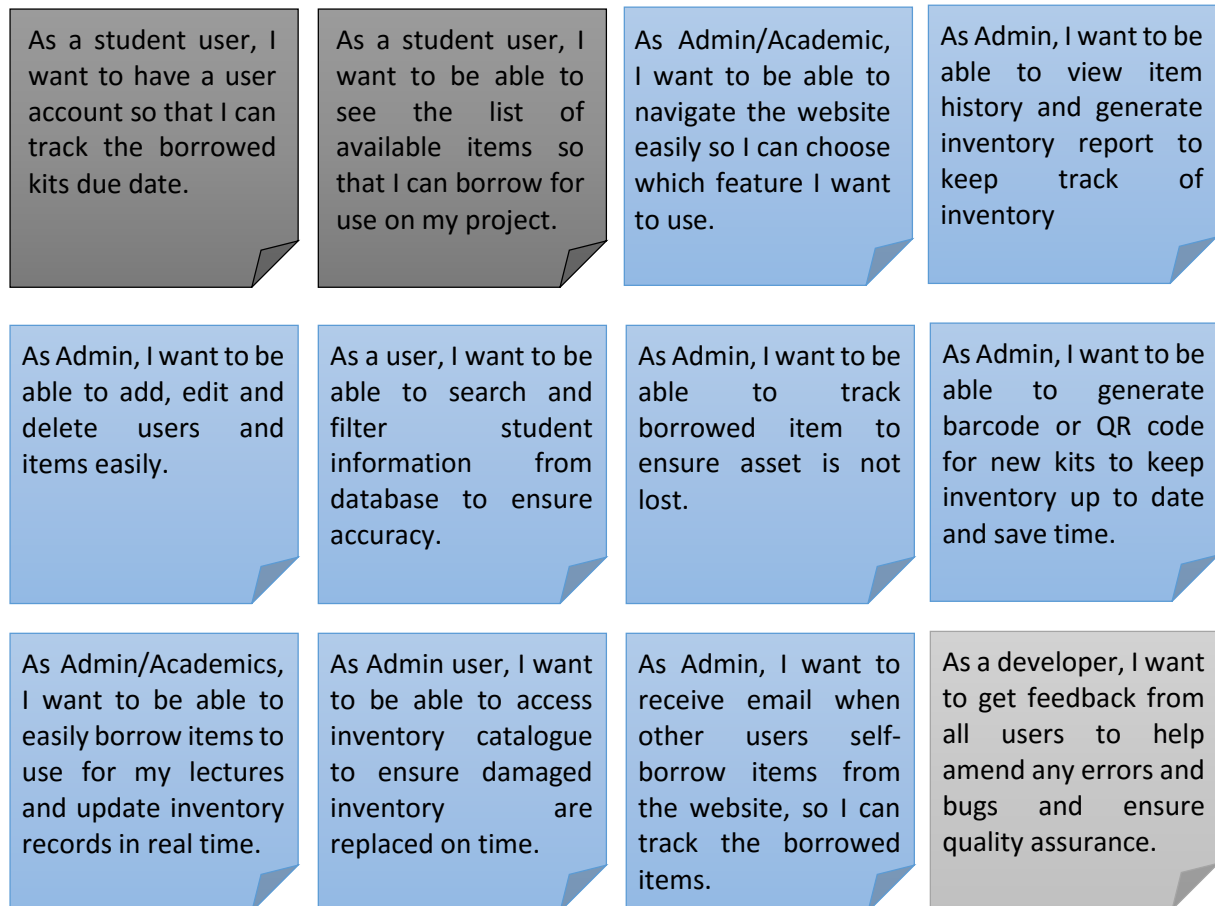


Figure 1: User Stories from Research Questionnaire

### 3.3 Software Requirements

From the initial discussion with the primary stakeholder along with results obtained from the analysis of applications currently available in the market and in conjunction with the recommendations made by research questionnaire respondents, requirements for the development of the application were identified and will be discussed further in the following sections.

#### 3.3.1 System Overview

This section describes software requirements for a new attempt to implement Inventory Management System for asset management. The current application described in Chapter 2 (2.2) was developed in 2014/15 as a mobile application but there are limitations to its features and generally not available to all users. The app did not target the need of the users and lacked core features that kept it from providing a complete solution to the inventory management needs of the department.

IMS “Inventory Manager” shall serve as the sole inventory management system of assets and provide means for controlling and monitoring its circulation. The primary user shall be the Administrator who shall create user account for other level of users and monitor activities of both users and assets. Other users shall have the option to create an account according to their role within the University to use the application.

### 3.3.2 Assumptions and Constraints

It is assumed that Inventory Management System (IMS) is designed for the School of Computer and Mathematics use only.

### 3.4 Non-Functional Requirements

- The system shall be able to deploy on University server
- The system shall support all major browser and different types of devices
- The system shall provide information on contact/help
- The system shall provide other link to the main University website and all other Ulster University campus websites
- The system shall provide appropriate security to ensure only authorised users have access to other users’ information.

### 3.5 Functional Requirements

#### 3.5.1 Processing Users

- The system shall provide a way for authorised users to log on by providing a valid username/password pair
- The system shall provide a means to add a new user to the database
- The system shall provide a means to edit a user into database
- The system shall provide a means to delete a user from the database
- The system shall provide a means to create a password that is between 6-18 characters
- The system shall prevent creating a user account if it does not contain the University ID (username), first name, last name, email and password
- The system shall be able to present a list of users that matches the criteria when an authorised user provides part of user details.
- The system shall provide a means to return list of users. For each user, it shall list user image, University ID, first name, last name, email, module, year, user role, status, and options to edit or delete user
- The system shall provide a means to import users in csv file format

- The system shall prevent adding a new user if it has the same University ID (username) as an existing user
- The system shall prevent changing the University ID (username) of an existing user to the same as another user
- The system shall prevent importing a csv file with duplicate entry to database
- The system shall prevent importing non-csv file into database

### 3.5.2 Processing Assets/Items

- The system shall provide a means to add new item to the database
- The system shall provide a means to edit an existing item to the database
- The system shall provide a means to delete an existing item from the database
- The system shall provide a means to set the name, category, description, quantity and barcode for an item being added or edited.
- The system shall provide a means to return list of items. For each item, it shall list; barcode ID, category, name, description, status, and option to issue item, edit or delete item
- The system shall prevent a user from adding an item with the same item name as an existing item
- The system shall be able to present a list of items that matches the criteria when an authorised user provides part of an item details
- The system shall provide a means to import items in csv file
- The system shall prevent importing duplicate item into the database
- The system shall prevent importing non-csv file into the database

### 3.5.3 Processing Asset/Item Category

- The system shall provide a means to add new item category to the database
- The system shall provide a means to edit an existing item category to the database
- The system shall provide a means to delete an existing item category from the database
- The system shall provide a means to set the category name being added or edited.
- The system shall prevent a user from adding an item category with the same category name as an existing category
- The system shall be able to present a list of item category that matches the criteria when an authorised user provides part of a category details
- The system shall provide a means to return list of category. For each category, it shall list ID, category and option to edit or delete



### 3.5.4 Processing Borrowed and Returned Items

- The system shall provide a means to add a new borrowed item to the database
- The system shall provide a means to update the database once item is returned
- The system shall provide a means to export issued history into csv file format
- The system shall provide a means to export returned history into csv file format
- When an issue item is created, the system shall provide a means to select an existing user as the borrower
- When an issue item is created, the system shall provide a means to select an item quantity required
- When an issue item is created, the system shall provide a means to select a date and time when the item is due for return
- The system shall provide a means to generate item quantity for each item name in database
- The system shall prevent a user from issuing/borrowing item if not available
- The system shall prevent a user from deleting an item that has been borrowed/on loan
- The system shall prevent a user from issuing/borrowing item if no due date selected
- The system shall prevent a user from issuing/borrowing item if it does not contain username
- The system shall prevent a user from entering item quantity greater than the available quantity

## 3.6 User Permissions

### 3.6.1 Admin/ Sub-Admin

If user is Admin or sub-Admin, IMS shall provide the following permissions:

- Permission to log into IMS
- Permission to edit user profile
- Permission to change password
- Permission to add, edit, and delete users
- Permission to add, edit, and delete category
- Permission to add, edit, and delete assets/items
- Permission to issue and return item
- Permission to confirm item borrowed and returned by academics

#### 3.6.1.1 Administration

- The system shall have the ability to generate item barcode ID automatically when new items are created
- The system shall have the ability to print item barcode when item barcode is selected

#### 3.6.1.2 Reporting

- The system shall provide a report on dashboard which includes the total users, total assets, total items on loan, and online users
- The system shall provide a Report capable of listing complete issued and returned item history
- The Report shall return list for each entry which includes username, barcode ID, item name, quantity returned, issue date, due date, and returned date
- The system shall provide a means for user to output the report in pdf and csv file format

#### 3.6.2 Academics

If Academic, IMS shall provide the following permissions:

- Permission to log into IMS
- Permission to edit user profile
- Permission to change password
- Permission to view all items
- Permission to borrow item
- Permission to view borrowed item history
- Permission to return item
- Permission to view returned item history
- Permission to generate report

#### 3.6.3 Student

If Student, IMS shall provide the following permission:

- Permission to log into IMS
- Permission to edit user profile
- Permission to change password
- Permission to view all items
- Permission to view borrowed item
- Permission to view borrowed item history

### 3.7 Summary

This chapter has analysed information obtained from the research into existing software in the market, recommendations made by respondents from the research questionnaire along with stakeholder's requirements in order to determine detailed requirement for the website. These requirements have been separated into different stages which are to be incorporated into the overall design of the website.

## Design

### 4.1 Introduction

The overall design of the application is extremely important as it ensures that any potential issues are acknowledged and dealt with swiftly. This chapter provides an overview of the software architecture used in the development of the application and for a better understanding of the layout and content within each page, a series of wireframes has been developed. The content within this will be discussed and a further illustration using storyboard which provides algorithms of the structure. It also includes a discussion of the structure of the relational database and the User Interface Design.

### 4.2 Application Architecture

Before designing the wireframes and prototype, it was important to ensure that there was a clear understanding of the application architecture and to ensure all pages were considered in the design process. This helps to organise and clarify the content required for the application and to eliminate unnecessary pages. The architectural design of the proposed application as outlined below start with the Home page which serves as the landing and welcome page for all users/ visitors. The homepage offers navigation to other content sections of the website such as Login and Sign up which is required to access the User Profile page where users can make use of the resources within the website.

The planning of the website architecture involves analysing the technical and functional criteria and putting them in categories to have a well-structured outlook. The first category as represented in the site map is the homepage and its contents. The homepage is further divided into three sections which are the navigation bar, the body and the footer. The navigation system which is placed at the top of the page aids users/visitors in accessing information on the website through various links to other areas of the website. Each link is defined as a text hyperlink with the relevant title being displayed. Once the user navigates to the area of interest, the relevant link title is displayed using the “hover” effect changing the text colour to white from grey. The navigation bar which has a black background is displayed horizontally on devices with larger screen size such as desktop or laptop. However, the navigation bar is hidden and replaced by a button in the top right corner on smaller devices with small screen size such as phones. Only when the button is clicked, the navigation bar will be displayed with various links which the user can select from.

The second section of the homepage which is the body includes a carousel plugin component for cycling through picture elements like a slideshow which contains four slides. It also includes two static pictures to its left /middle and links to Ulster University campus websites to its right. The last section of the homepage is the footer which contains copyright information.

The second category is acknowledging the software requirements according to its users. The users as identified from stakeholder's requirements includes: Students, Academics, and Admin /Sub-admin. Each user's category has some functions identical and some specific to Admin/Sub-admin. The functions as listed in the site map includes:

#### **Students**

- The user's ability to View All Items
- The user's ability to view all items borrowed (Issue History)
- The user's ability to view the list of all item returned (Return History)
- The user's ability to view and edit own profile (Profile / Edit)
- The user's ability to change account password (change password)

#### **Academics**

- The user's ability to View All Items
- The user's ability to Borrow Item
- The user's ability to view all items borrowed (Issue History)
- The user's ability to return borrowed item (Return Item)
- The user's ability to view the list of all item returned (Return History)
- The user's ability to generate own report (Generate Report)
- The user's ability to view and edit own profile (Profile / Edit)
- The user's ability to change account password (change password)

#### **Admin / Sub-admin**

##### **➤ Users:**

- The user's ability to View All Users
- The user's ability to Manage other users (Add, Edit, and Delete)

##### **➤ Items:**

- The user's ability to View All Items
- The user's ability to Manage All Items/Assets (Add, Edit, Issue, and Delete)
- The user's ability to Manage Items/Assets Category (Add, Edit, and Delete)
- The user's ability to print Barcode for an item

##### **➤ Circulation:**

- The user's ability to view all items borrowed/issued (Issue History)
- The user's ability to return borrowed item (Return Item)
- The user's ability to view the list of all item returned (Return History)

➤ **Generate Report:**

- The user's ability to generate own report for the system activity over a period of time.

➤ **Profile:**

- The user's ability to view and edit own profile (Profile / Edit)
- The user's ability to change account password (change password)

**INVENTORY MANAGER (SITE MAP)**

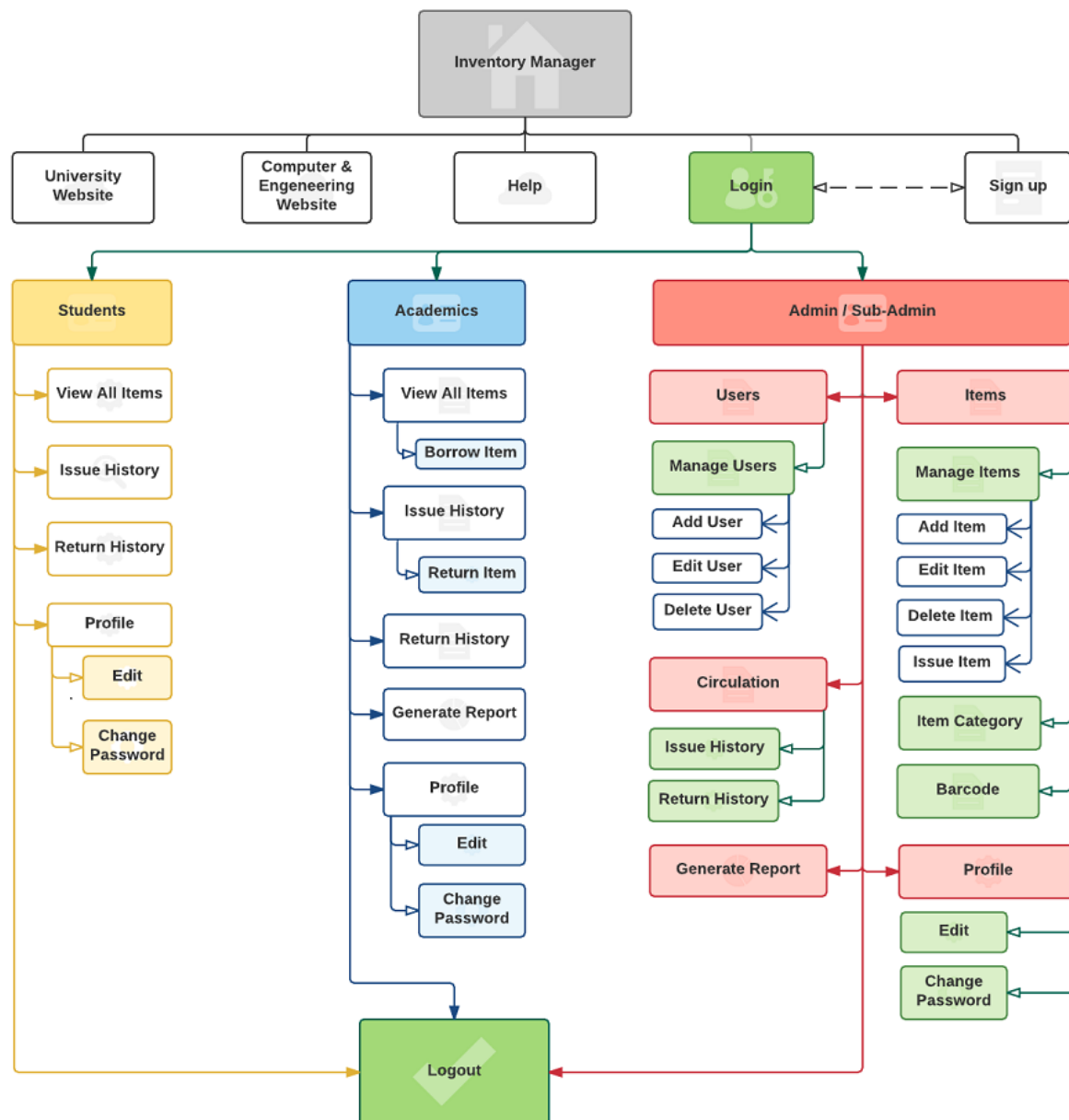


Figure 2: Site Map for 'IMS | Inventory Manager'

### 4.3 Initial Design

The design has two main objectives. The first objective is the use of wireframe to ensure consistency and capture the details of the software requirements outlined in the site map above ensuring all functions and features of the application are not overlooked. The second objective is to have a virtual and detailed representation of the website content arrangement, interface elements, navigational systems and how they work together. This approach allows for a proper interpretation and understanding of the stakeholder's requirements along with users' stories on which the development of the application is based. It also serves as a guide in the display of certain information and provide the means by which development process were prioritised. Figure 3 below is an illustration of the wireframe designed for home page. (See Appendix Three for more UI wireframes).

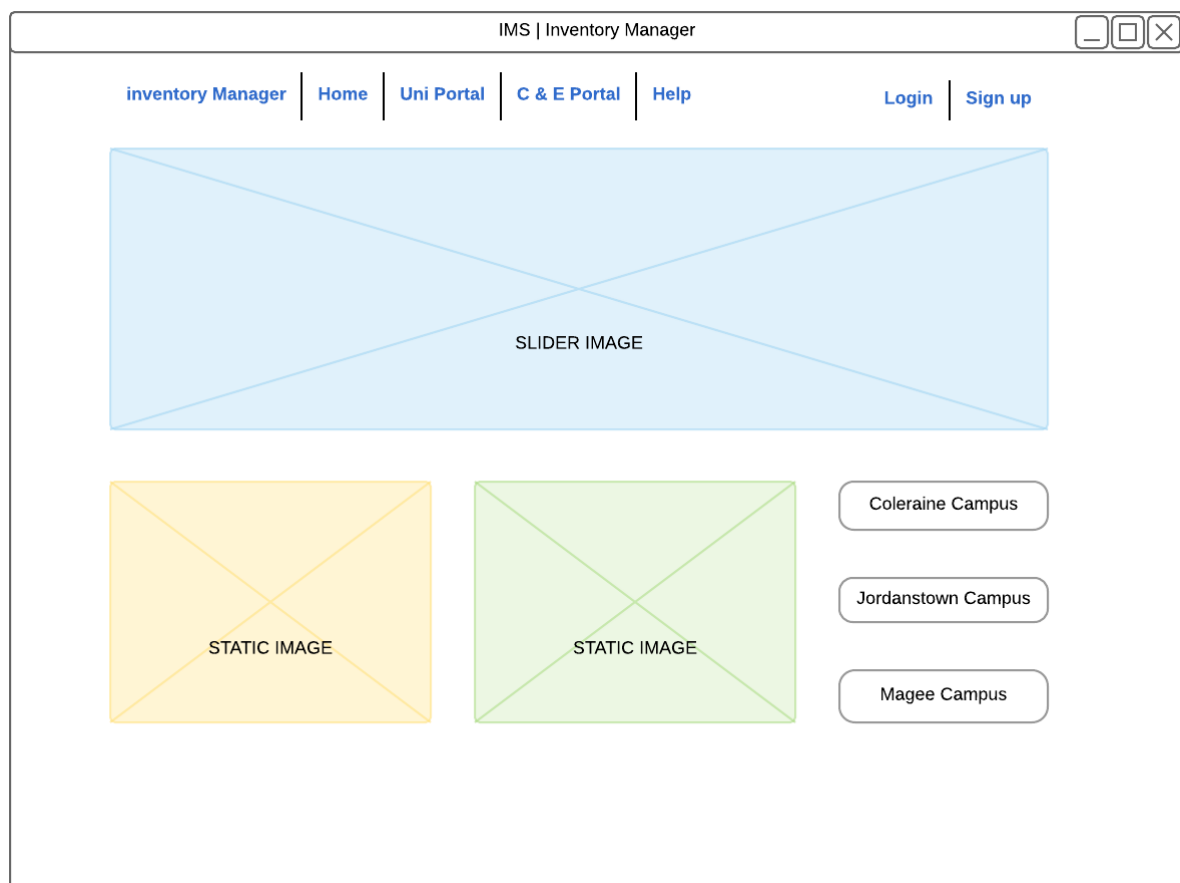


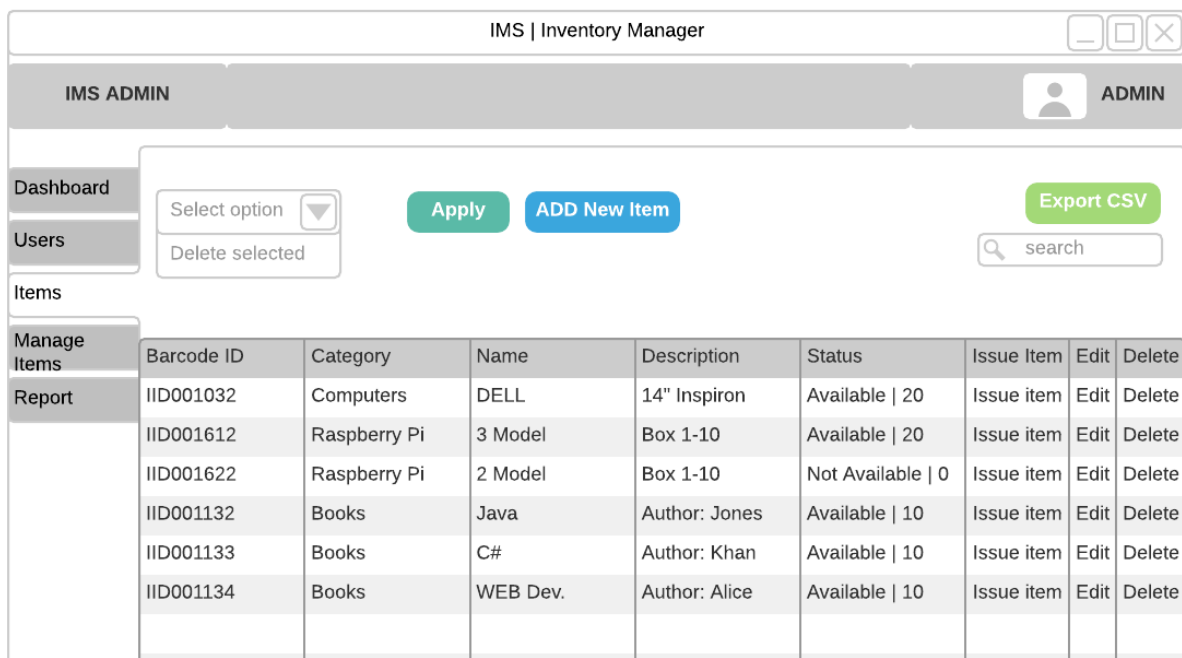
Figure 3: Wireframe Design of the Homepage

#### 4.4 User Interface Design

The User Interface was designed with the goal of being simple and intuitive whilst achieving consistency throughout the entire website. It is important to state that the UI design for the website was developed in similarity with the Ulster University website adopting some of its characteristics. This is deemed important to give users same look and feel as the website will be hosted on the University server. It will also give users some consistency if they choose to link to the University website from the Inventory Management website homepage.

The User Interface Design is made up of twelve individual wireframes with some design being used across all users. It also serves as a means by which the user and system interaction are captured. Some requires input to give users the expected outcome while some requires no input. As part of the solution to the existing problems, the system's ability to provide real time information is captured by the functions identical to all users such as "View All Items". This allows all users to have access to the same inventory information in real time.

It also captures some of the important functions particular to the Admin /Sub-admin and Academics which allows for the proper management of the inventory assets. Such includes the "issue and return" functionality available to the two key players (Admin/ Sub-admin and Academics). Once an item is borrowed or issued by any of them, the information is updated in the database in real time ensuring same level of information is available to all users.



The wireframe shows a web application titled "IMS | Inventory Manager". It features a top navigation bar with "IMS ADMIN" and a user profile icon labeled "ADMIN". A left sidebar contains a menu with "Dashboard", "Users", "Items", "Manage Items", and "Report". The main content area includes a "Select option" dropdown, an "Apply" button, an "ADD New Item" button, and an "Export CSV" button. A search bar is also present. Below these elements is a table with 9 columns: Barcode ID, Category, Name, Description, Status, Issue Item, Edit, and Delete. The table contains 6 rows of data, including items like "DELL 14\" Inspiron" and "Raspberry Pi 3 Model".

Barcode ID	Category	Name	Description	Status	Issue Item	Edit	Delete
IID001032	Computers	DELL	14" Inspiron	Available   20	Issue item	Edit	Delete
IID001612	Raspberry Pi	3 Model	Box 1-10	Available   20	Issue item	Edit	Delete
IID001622	Raspberry Pi	2 Model	Box 1-10	Not Available   0	Issue item	Edit	Delete
IID001132	Books	Java	Author: Jones	Available   10	Issue item	Edit	Delete
IID001133	Books	C#	Author: Khan	Available   10	Issue item	Edit	Delete
IID001134	Books	WEB Dev.	Author: Alice	Available   10	Issue item	Edit	Delete

Figure 4: Wireframe Design of the Item page

#### 4.4.1 Storyboard

To further understand the navigational architecture and information flow on the website, storyboards were designed to demonstrate how the pages are to work together in order to provide user's interactive experience. The Figure 5 below represents an example storyboard of an Admin user interacting with the website. (See Appendix Four for Admin dashboard example)

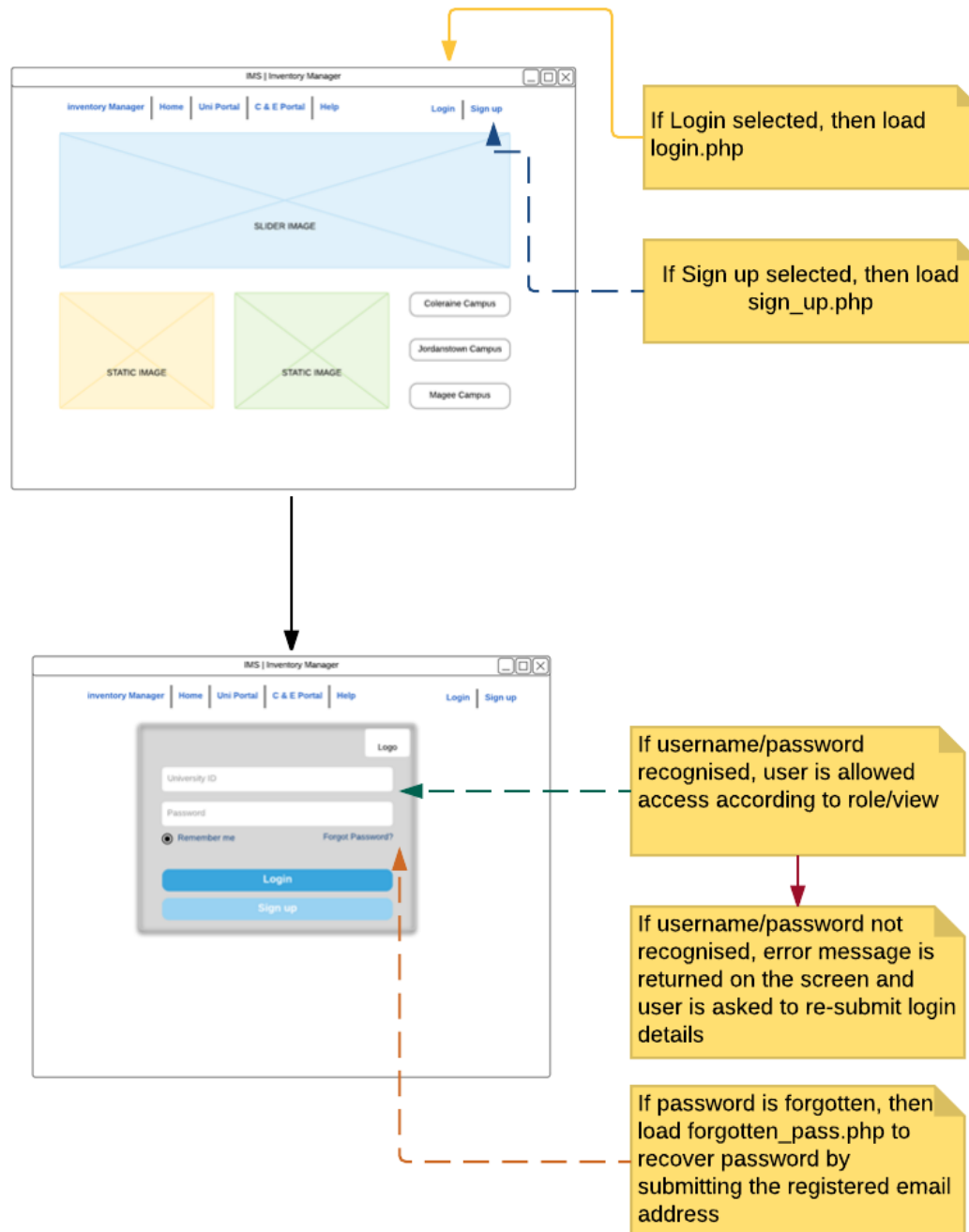


Figure 5: Example Storyboard of the Login process



## 4.5 Database Design

The database which provides the interaction with user and other application builds upon the foundation of relational database methodology. The Database Management systems (DBMS) is designed to allow the definition, creation, querying, update, and administration of the database by the user. The relational methodology outlines a series of tables which are interconnected by specific relationship allowing for the storage, retrieval and organisation of data. In order to develop an efficient database, it is important to apply an entity relationship (ER) model with the goal of establishing a visual representation of the information required for storage and the establishment of relationship between tables.

## 4.6 Table Structure

The database for this application consists of six tables, each of which has been allocated a unique and meaningful name which identifies individual tables. In addition, each entity is assigned attributes which can be described as properties or characteristics of the entity. Each table has a given primary key which uniquely identifies individual tuples in a given table. The key assigned are unique and not null ensuring duplicate values are not created in adherence to entity integrity. The six tables within the database are users, items, item\_categories, issued, returned, users\_online. The users table consists of attributes relating to the user's interaction with the application while the items table is constructed to hold information about the inventory assets. When an item is borrowed by the user, the information relating to the transaction is required to be stored in a separate table within the database hence the issued table was created. On the return of the item borrowed, the details of the transactions are store in returned table. The users are logged into the system; their activity is saved in the database using the users\_online table. Using the principles of normalisation (3NF), the tables in Appendix Five have been constructed.

To further understand the relationship between these tables, an entity-relationship diagram was developed. The ER diagram illustrates the relationship between the entities and attributes of the database. It also helps for the informal design of the database and help to understand how the entities and attributes are defined in terms of their relationship either Many-One, One-One, etc. Figure 6 below represents the ER diagram of the table structure.

## IMS | INVENTORY MANAGER - ERD

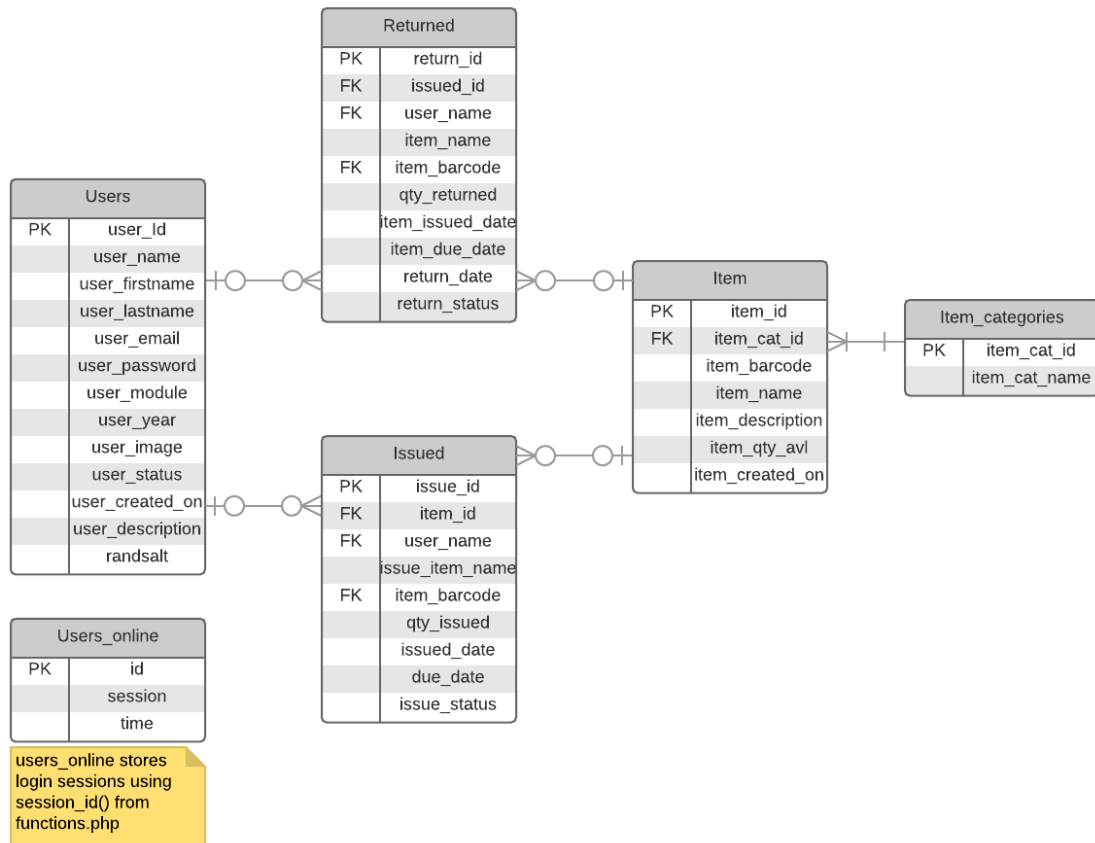


Figure 6: Entity Relationship Diagram (ERD)

### 4.7 Summary

This chapter has outlined the design choices for the proposed website 'IMS | Inventory Manager'. It then describes the different elements of the architectural design and their relationship. The chapter draw a conclusion by discussing the detail structure of the tables providing the different relationship through the Entity Relationship Diagram (ERD).

# Implementation

## 5.1 Introduction

This chapter details the implementation of the proposed application and briefly discussed the software methodology used. The implementation process makes use of the storyboards to create an efficient blueprint for the application development and learning gained from the development of the prototype. It discusses further the platform used and provide an in-depth discussion of the core functionalities of the system.

## 5.2 The Prototype

Progressing from the use of wireframe in capturing the system behaviour and interaction with the user requires creating a prototype. This was designed using the software requirements gathered from the project analysis phase acknowledging the outcome of research into similar software, focus group and discussions with the main stakeholder. The first feature considered was the login/sign up options for users. This was a requirement which allows authorise access into the system main features. It also allows for an interaction with the database previously created (Back-End) and the Login/Sign up UI (Front-End).

The content within this page and throughout the entire application is easily accessible and clearly outlined. Figure 7 below illustrates the validation carried out using the login and sign up from the prototype.

**Login**

• You need to enter a username and password

**Username**  
Enter University ID

**Password**  
Enter password

☒ Remember me

[Login](#)

Not Registered? [Sign Up](#)  
Forgot Username or Password?

**Sign Up**

• Fields marked with asterisk(\*) are required  
• Your password must be at least 6 characters.  
• A valid email address is required.

**Username\***  
University ID

**Password\***  
Password

**Confirm password\***  
Confirm password

By using empty field, the system performs a validation check represented by the red message text before interacting with the database

Figure 7: Prototype of UI Validation

There were a number of important aspect of developing the prototype, the most significant is to understand the system behaviour through its interaction with the database. For example, one of the Admin requirement is the ability to “list all items” from the database. By developing a prototype, it was easier to understand how the pages’ work together with database and what SQL query is required for this purpose. Figure 8 illustrates the Admin page from the prototype and a query performed through the Admin UI by selecting “View All Items” which using the SQL SELECT syntax interact with the database providing the information to be displayed in Admin UI.

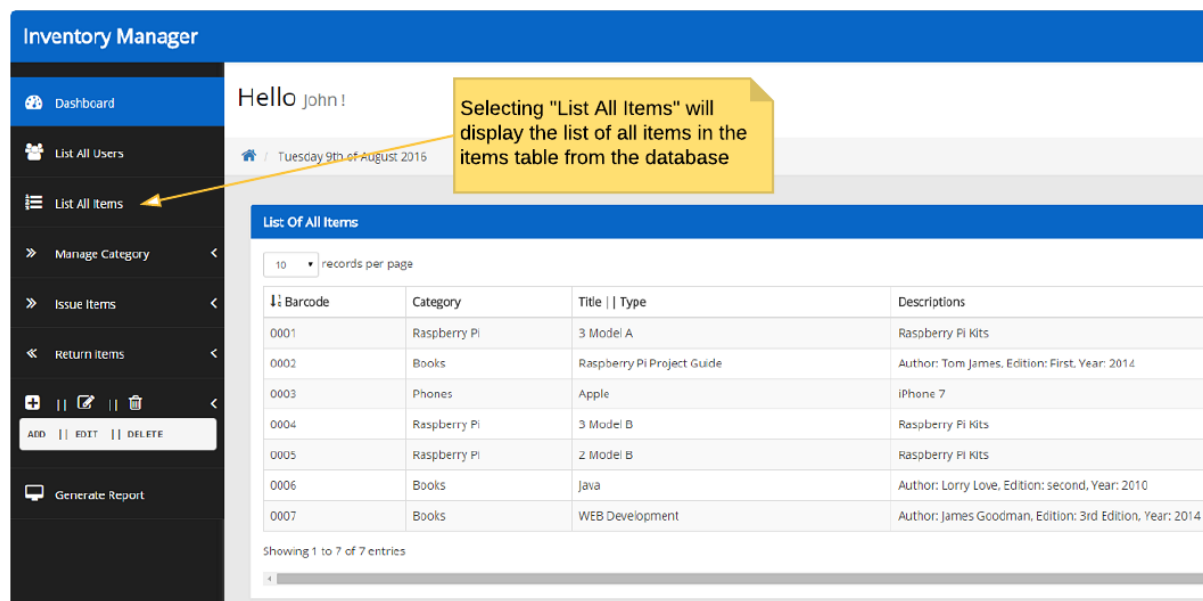


Figure 8: Prototype Admin example - "All Items"

The key findings from building the prototype was instrumental in the development of the real application. Throughout the process, there have been continuous learning of requirement completed without issues and as well improving from development failures.

### 5.3 Development Methodology

From the initial software development phase, series of SDLC model were considered which includes waterfall and Agile. However, due to the flexibility required in the development process and the level of information available at the start, it was decided that Agile Model will be more suitable. The reason being the agile model’s ability to quickly respond to unpredictable changes as a result of which requirements and solution evolve throughout continuous improvement.

Adapting the agile model allows for a better working environment with the stakeholder and project supervisors and the following steps were established in order to achieve a successful outcome.

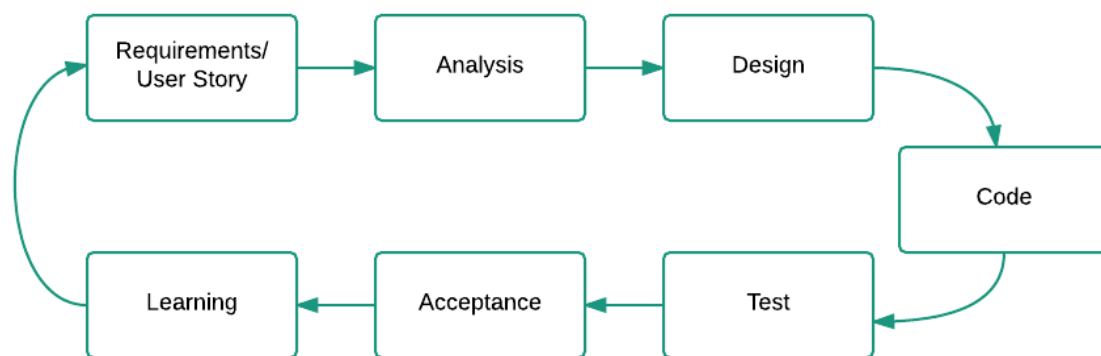
**Step 1:** Each user story was broken into several iterations/sprints in respect to user's hierarchy e.g. Admin, Academics and Students.

**Step 2:** Each iteration was given between 1-2 weeks' time duration for completion.

**Step 3:** Prioritising each iteration according to how important the requirements/features are in arriving at a functional software.

**Step 4:** At the end of each iteration, a working feature is developed and manual testing performed checking for unforeseen errors not picked up during the development.

**Step 5:** The cycle moves unto the next iteration following the same process and any features not completed is either re-prioritised or carried unto the next iteration depending on its complexity.



*Figure 9: Personal Development process used (Agile Model)*

The Agile methodology provides opportunities to assess the direction of the project through meeting with supervisors and stakeholders. For example, the initial key players were identified to be Admin and the Students. After further consultation with the stakeholder, there were changes in requirements in which another key player was created as Academics. The flexible nature of agile allows for this new request to be prioritised and dealt with accordingly within the product backlog.

Product Backlog /Requirements	Sprint Backlog	Features	In progress	Review	In Test	Completed	Issues
Home Page	As a user, I want to be able to navigate the website easily so I can choose which feature I want to use. {HOME PAGE}	<ul style="list-style-type: none"> <li>• Sign Up</li> <li>• Login</li> <li>• Links to other sites</li> </ul>	01/06/2016	07/06/2016	08/06/2016	10/06/2016	
Users Account Admin and Other Users (User type)	As a user, I want to have a user account so that I can manage my actions on the website. {ADMIN} / {Other USER}	# Admin # Other Users <ul style="list-style-type: none"> <li>• Settings</li> <li>• Change password</li> </ul> # Database <ul style="list-style-type: none"> <li>• Normalisation</li> <li>• Implementation</li> <li>• Integrity</li> </ul>	06/06/2016	14/06/2016	15/06/2016	17/06/2016	
Access to list of Items and Users as Admin	As Admin, I want to be able to access inventory & user catalogue to ensure accuracy. {ADMIN}	<ul style="list-style-type: none"> <li>• List All Items</li> <li>• List All Users</li> </ul>	20/06/2016	23/06/2016	24/06/2016	25/06/2016	

Figure 10: Example of Sprint Backlog used in application development

## 5.4 Platform Implementation

A range of potential implementation platforms were considered which includes Adobe Dreamweaver, Bootstrap and WordPress for the front-end development. Editors such as Notepad++ and Sublime Text editor were the free source code editors considered as they support several languages and can run in most environment. For the back-end development, Laravel and Yii were the two PHP frameworks considered for the purpose of this project.

Among the front-end, Bootstrap was found to be more easily accessible, free and open-source framework which contains all required components and learning resources for beginners available on YouTube and Udemy.com. Both Notepad++ and Sublime Text editor were found to be very useful for the source code editing

Laravel was considered to be the best PHP framework for the back-end development with video tutorials on Laracast and good reviews from developers in the field. However, having difficulties with the installation process on personal PC, a decision was made to insert PHP into the HTML documents. This is due to the limited time scale of the project and therefore important to ensure all deadlines are met and all issues dealt with accordingly.

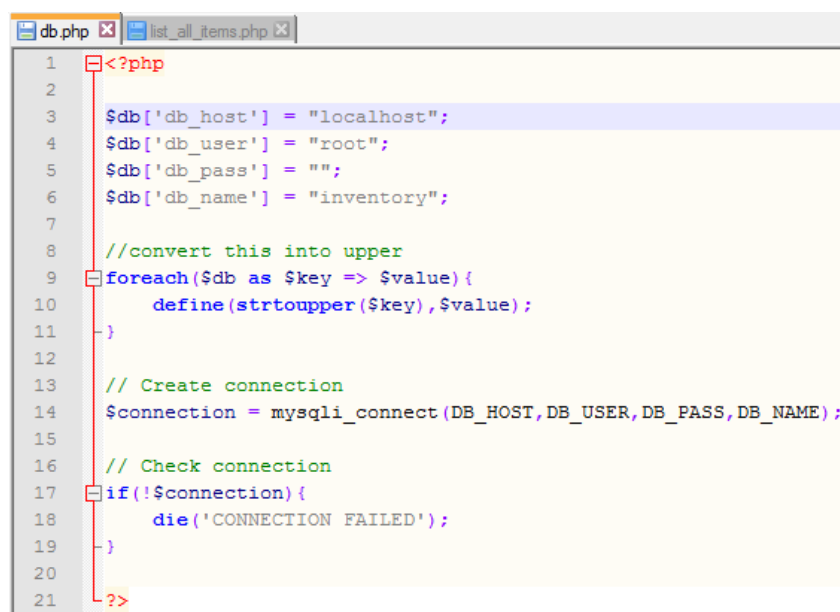
### 5.4.1 Resource Implementation

XAMPP was used as the web server which bring together both the front-end and the back-end of the development. Some of its primary components includes Apache, MySQL, PHP, phpMyAdmin and Perl used to create full-fledged web server. The XAMPP web server provides access to the resource folder located in its root directory by accessing <http://localhost> or 127.0.0.1 from browser.

The project resource folder allows for values to be stored and resources to be maintained within the appropriate folders. For example, “includes” folder contains the files which allow for code to be re-used in creating multiple pages with consistency. The management of the resource folder is an extremely important element of the project development as it allows for values and properties to be referenced accordingly whilst ensuring that the project structure is understandable and as efficient as possible.

### 5.5 Database Implementation

As mentioned in previous section, XAMPP web server was used for the back-end and therefore access to phpMyAdmin facilitated the creation of a database. Once MySQL database have been created, connection was established using php syntax to be able to retrieve data from the database and display the result on the web page.

A screenshot of a code editor with two tabs: 'db.php' and 'list\_all\_items.php'. The 'db.php' tab is active, showing PHP code for database connection. The code includes comments and variable assignments for database host, user, password, and name, followed by a foreach loop to convert keys to uppercase, and finally a mysqli\_connect call and a connection check.

```
1 <?php
2
3 $db['db_host'] = "localhost";
4 $db['db_user'] = "root";
5 $db['db_pass'] = "";
6 $db['db_name'] = "inventory";
7
8 //convert this into upper
9 foreach($db as $key => $value){
10     define(strtoupper($key), $value);
11 }
12
13 // Create connection
14 $connection = mysqli_connect(DB_HOST,DB_USER,DB_PASS,DB_NAME);
15
16 // Check connection
17 if(!$connection){
18     die('CONNECTION FAILED');
19 }
20
21 ?>
```

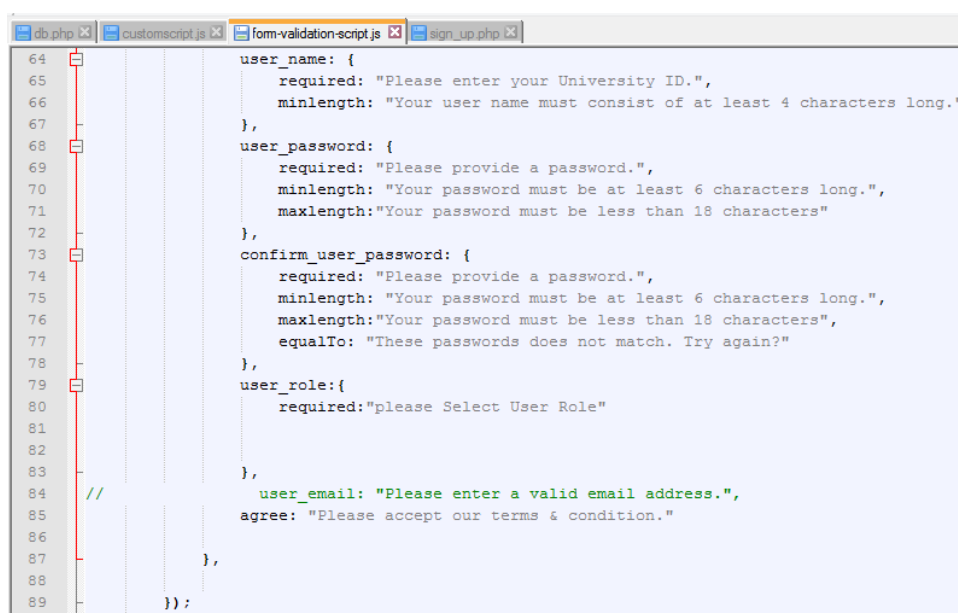
Figure 11: Connecting to MySQL database (db.php)

## 5.6 Implementation of Front-End (HTML, CSS and JS)

One of the main goal of the project is to ensure the proposed web site is responsive and look good on all devices, from small phones to large desktop. By using Bootstrap, it was easier to design and create a website which automatically adjust itself on any devices. Also, with various templates and example available on [getbootstrap.com](https://getbootstrap.com), it was easy to customise and adapt some of the template used in developing the application. However, customising the templates requires a good knowledge of HTML, CSS and JS. These was acquired by completing short web development courses on [Udemy.com](https://www.udemy.com), watching web development tutorials on YouTube and making use of the free tutorials on [w3schools.com](https://www.w3schools.com).

It is important to note that even though example templates were modified, majority of the scripts used in the Human Computer Interaction were written from scratch. For example, the structure on the template were built as basic template and does not have any script in term of input validation. By creating `form-validation-script.js` enable the user input to be validated accordingly in line with Client-Side validation standard. For better user experience, JavaScript language was used to validate user's input as they type to create more responsiveness and visually rich validation.

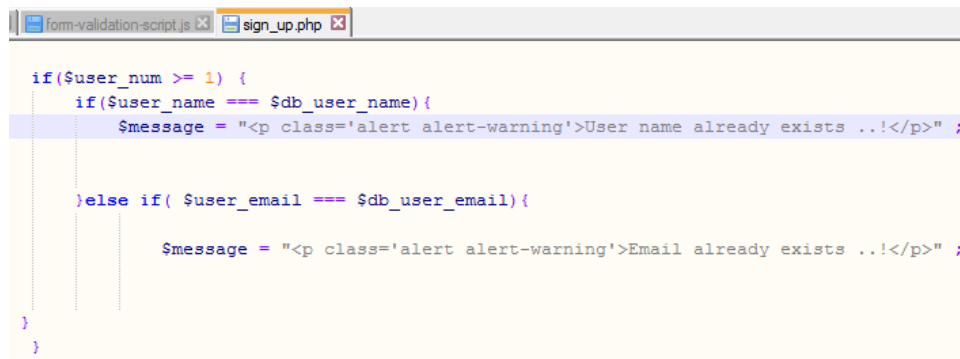
For example, the "Register" form to create a new account validate each input individually starting from Username input field. When a user input a username (University ID) which has already been registered, a warning message such as "Username Not Available" is displayed immediately to the user. With client-side validation, form never gets submitted if validation fails. Also to prevent the main drawback such as when JavaScript is turned off on user's device, php syntax handles the server-side validation to ensure better user experience.

A screenshot of a code editor with four tabs: 'db.php', 'customscript.js', 'form-validation-script.js', and 'sign\_up.php'. The 'form-validation-script.js' tab is active, showing a JavaScript object with validation rules. The code is as follows:

```
64 user_name: {
65   required: "Please enter your University ID.",
66   minlength: "Your user name must consist of at least 4 characters long."
67 },
68 user_password: {
69   required: "Please provide a password.",
70   minlength: "Your password must be at least 6 characters long.",
71   maxlength: "Your password must be less than 18 characters"
72 },
73 confirm_user_password: {
74   required: "Please provide a password.",
75   minlength: "Your password must be at least 6 characters long.",
76   maxlength: "Your password must be less than 18 characters",
77   equalTo: "These passwords does not match. Try again?"
78 },
79 user_role:{
80   required:"please Select User Role"
81 },
82 //
83 user_email: "Please enter a valid email address.",
84 agree: "Please accept our terms & condition."
85 },
86 },
87 },
88 },
89 );
```

Figure 12: JavaScript form validation (client-side)





```

if($user_num >= 1) {
    if($user_name === $db_user_name){
        $message = "<p class='alert alert-warning'>User name already exists ..!</p>" ;

    }else if( $user_email === $db_user_email){

        $message = "<p class='alert alert-warning'>Email already exists ..!</p>" ;

    }
}

```

Figure 13: Example php form validation (server-side)

## 5.7 Core Functionalities

Most of the functionalities implemented in the development are considered to be very important as each contributes to the development of a functional system. However, few functions were identified as being essential to ensure the website is robust. Each of these core functions have been outlined and detailed below.

### 5.7.1 Issue/Borrow Item

The issue/ borrow item allows transaction between a user and item to be established. This transaction could either be implemented by Admin or Academic within the user's hierarchy. To issue/borrow an item, "issue item" or "borrow item" is selected from the UI. This selection processes a call to the issue\_item.php file and the following actions are performed:

- ✓ The item selected is identified using the item\_id from the database
- ✓ SQL query is performed to SELECT from the "items table" WHERE the item\_id selected is the same as the row in the table.
- ✓ Once the row is identified using the item\_id, associated information is collected from the row and user name is selected from the "users table" already populated in the UI.
- ✓ The issue date is automatically set to the current date using the syntax \$issue\_date = ("y/m/d h: i: s a") and the item\_due date is selected by the user from the calendar already populated using the jquery.datetimepicker.full.js
- ✓ To complete the transaction, another SQL query is performed which is to INSERT all the details of the transaction into the "issued table" where the loan record is kept.

After the transaction is completed, it is necessary to ensure that items on loan could not be deleted by any user from the database. To achieve this, a conditional statement was used in the `list_all_item.php`. The function of this condition is to **disable** the delete button once an item is issued or borrowed. This ensures that items on loan are at all time safe from any human error and creates a good software development practice. (See Appendix Six for script example)

### 5.7.2 Import CSV file

Importing records from external source to the database is one of the most important features in the development. This function provides a solution to inventory management by saving time taken to process items or users data individually through a bulk import. The Admin user creates a file `format.csv` which is meant to be a way of moving data from one source to another.

For this process to be effective, the file format is required to be in CSV and each row of data in the file must follow similar arrangement in the items or users table within the database. To import a supported CSV file, the user ensures that:

- Add Item or Add User is selected from the side navigation menu upon which `add_item.php` or `add_user.php` is called.
- The item category or user role is selected from the drop down. This is achieved by SQL SELECT statement requesting ALL categories or roles from either the `item_category` table or the `user_type` table.
- The user then browses the csv file location, select the file and click the “upload” button.
- The `$file = $_FILES ['file'] ['tmp_name']` get the csv file from its location.
- SQL SELECT query is used to select data from the database and compared with the csv data.
- If data already exists, the import failed and a warning message is presented to user such as “Import Failed. Reason: Duplicate Entry in CSV file”. Else, the CSV file data import is successfully processed into the database. (See Appendix Seven for script example)

## 5.8 Summary

This chapter has detailed the implementation of the prototype and the learning achieved in progressing the development through agile model. It also detailed both the Front-End and Back-End including some of the core functionality implemented and the Error handling process put in place to ensure good software practice.

## Testing and Evaluation

### 6.1 Introduction

This chapter outlines the details of testing process and the testing environment in which the application was developed. There has being continuous testing performed during the development life cycle to ensure that the software is efficient and reliable. The testing environment involved non-functional testing, functional testing and User Defined Testing through some members of the focus group. This process has been fundamental in the implementation of the application and has allowed for features to be reconsidered and evaluated. The testing environment has ensured that the objectives of the application are met and the stakeholder's/ end-users requirements are more refined.

### 6.2 Functional Testing

The functional testing of an application enhances the credibility, usability, and effectiveness of the application. Testing is an important part of development life cycle as it ensures the value of the system is measured against the system requirements. It also validates the correctness of the application as issue raised are resolved and improved upon. Appendix Eight illustrates the test cases used, the outcome of the test performed and any issues discovered during the testing. The "Add User" functionality failed when test was performed using special characters among input. The current implementation failed to register a user with last name such as **O'Neil**. After researching the issue, it was discovered that MySQLi real escape string function should be used to escape special characters in a string for an SQL statements. The reason for this is to sanitize user input and protect against accidental or intentional SQL injection which may corrupt the database. The problem was resolved by ensuring all user input are sanitized using the function.

Appendix 8.2 illustrates the overall functional testing result for the application. The result illustrate that 30 test cases were created out of which 29 passed and 1 failed. This represents that 97% of the test performed were successful and only 3% failed. The test result confirm the functional testing performed on the application was successful.

### 6.3 Non-Functional Testing

Non-Functional Testing was also carried out to check compatibility of the hardware using different devices such as Desktop, Laptop, Tablet and a smart phone. Further test was also performed to test compatibility with different browsers such as IE, Chrome, Firefox, and Safari. The last testing performed was to check compatibility on the Operating Systems such as Windows 10, Windows Vista, MAC, and iOS. (See Appendix Ten)

## 6.4 Focus Group Testing

To develop a system which is acceptable to all users, it was necessary to evaluate the final product using members of the focus group. A second questionnaire was created and administered to users who had responded to the initial research questionnaire and agreed to test the application once completed. The questionnaire was designed using scenario based approach where users were required to complete certain task and rate their experience in terms of difficulty on a scale of one to five with one being very difficult and five being very easy. In addition to this questions, visual step by step instructions was provided as a guide in completing the task. From the response received, results were gathered and illustrated in Table 3 with one of the response from the questionnaire in Appendix 8.3.

Each task within the questionnaire has been outlined and detailed below:

- **Homepage:** this task was designed to test compatibility as each user uses different browsers and operating systems. It also tests the first impression of users in regards to the layout presentation of the UI and page navigation.
- **Creating a user account:** This task required the users to interact with the system by providing user-input. It also tests the performance of the application in terms of efficiency and response time.
- **Admin Login:** This task required users to examine the Admin UI reliability and accuracy. As user is logged in as Admin, a summary of the current state of the application is available to view from the dashboard.
- **Approve a user:** This task is required to test the functionality by approving the user account previously created. This task also test the security of the site as registered users required approval before their account becomes active.
- **View Item:** This task is to test consistency of the screen layout and colour content. By selecting a button which link to another view, the user's impression is tested in terms of how appealing they find the colour scheme.
- **Issue Item:** Users were required to issue an item to the user account created. This is to test the capability of the core function of the application and its resource usage.
- **New User Login:** Users were required to login using the user account created at the start of the questionnaire. This is to test the reliability and consistency of the application through interacting with back-pend.

The result obtained from User Defined Testing have been outlined below in Table 3.

Task	Description	Average Rating
1	Homepage presentation	5
2	Creating a user account	5
3	Admin login and functions	4.8
3.1	Approve a user	4.8
3.2	View Item	4.8
4	Issue Item	4.4
5	New User login	4.4
<b>Total Average</b>		<b>4.7</b>

*Table 3: User Defined Testing Results*

From the result presented and feedback from focus group, it was concluded that each user found the system fairly easy to use and navigate throughout. Most of the users find the links to be very straightforward and clearly signposted. Overall, each of the individual who took part in this evaluation felt the application was successful in that they would be happy to use the application once available. This is a positive outcome and essentially support the idea that the application meets all objectives outlined at the beginning of the application development.

## 6.5 Summary

This chapter detailed the testing performed in-house by the developer and the presented the results in an organised format. These results showed that the functionalities were performing as intended and that implementation was successful by achieving both cross browser and cross platform capability. In addition to this this chapter detailed the results gained from the testing scenarios/tasks administered to users in the focus group. The results gained from respondents further reinforced the conclusion that a successful implementation had been achieved and feedback received was welcomed and acted upon by the developer.

## Conclusion and Recommendations

### 7.1 Introduction

The final chapter of the report includes a summary of the project life cycle from conception to the development of the application. It also looked at all the objectives proposed at the beginning and a discussion on the development of proposed solutions. The chapter also assessed the result from the experience and a brief discussion of experience gained and issues encountered during the development process. Any revision made during the project development will also be outlined and any reason why the changes were implemented.

### 7.2 Summary of the Project Objectives

The advancement in technology has led to both public and private sectors awareness and promoting the study of computer science in various institutions through the use of small computers such as Raspberry Pi. In line with this, a number of Raspberry Pi development kits were acquired by the University to be used in the laboratory sessions and borrowed by both staffs and students. However, the management of these assets has become labour-intensive and difficult to manage.

The existing solution developed in 2014/15 as a mobile application to manage the assets failed to provide adequate solution as it lacks major functionality due to its limitation to as a single user app. As a result, the existing solution does not support multiple users and the process of managing inventory of the assets creates issues such as inability of the system to give real time information about the status of the inventory assets. As other users are not restricted access to the inventory assets, information about its status is difficult to establish and making tracking the items a difficult task. To fully understand the existing solution to the inventory management and the core functionality it lacks, series of discussions were held with the primary stakeholder to have more insight on the problems and how to provide a lasting solution to these problems. These discussions were summarised into basic objectives which includes:

- To conduct background research on the problem and demonstrate why this website constitutes an adequate solution.
- To examine and investigate the current market and analyse available inventory management system websites. Content of these applications will be evaluated as well as the overall layout and design. These details provide a good basis on which to design a prototype and further develop the final application.
- To compose and administer a research questionnaire to a group of lecturers and students in order to establish the key elements users would expect in terms of design, features, and content. The feedback from questionnaires will be fundamental in the overall design of the

website, both in terms of understanding essential content and potentially introducing new ideas to enhance its functionality.

- To develop a simple user interface, containing clearly defined sections which were easy to navigate, usually attractive and provided clear, concise information. This will consider the content and layout of available applications and build on these strengths to develop an effective website.
- To evaluate the website prior to testing and as a result provide recommendations for additional features or modifications.
- To create and administer a scenario based testing questionnaire to a group of individuals who have expressed agreement to test the system in the research questionnaire. This will ensure efficiency is of a high level and the project aim is being achieved.

In order to bring about a lasting solution to the current problems, a research was conducted into possible existing solutions readily available in the market deemed at solving inventory management problems. Five of these software's were examined by analysing their features and how it could be configured to provide a solution to the inventory issues within the School of Computer and Mathematics. Research within this area provide some outcomes as recommendations to be considered and used in the initial research questionnaire administered to a focus group consisting of both lecturers and students within the University environment. The respondents of the questionnaire provided essential feedback and recommendations which were instrumental in the development of the application. The combination of the discussions with the primary stakeholder, research into existing solutions and response from the research questionnaire was useful in gathering a list of functional and non-functional requirements to be incorporated in the design of the application.

The design of the software was also influenced by usability and accessibility guidelines from the W3C which allowed a consistent structure to be outlined for the project. The User Interface was designed with a focus on the end-user in order to provide them with easy to use application that provides a lasting solution to inventory management problems, efficient system and a positive user experience.

The use of wireframes allows the developer to define the layout and presentation of the website and ensure that navigational properties of the website were properly managed. Storyboards were very instrumental in the understanding of the navigational architectures and structure of the website. The use of sitemap also helps to format a detailed architecture of the website design layout and its relationship with the relational database specifications.

The client server relationship was defined and a combination of programming language to be used in the development process. The client side which is the front-end includes language such as HTML, CSS, and JavaScript while the back-end (Server-side) which is php allows a connection with the database, retrieval and manipulation through MySQL queries. The combination of both the front-end and back-end ensures a resolute client server relationship in formed as well ensuring data communication is rapid, accurate and responsive.

By using the storyboards and design specifications as guidelines, implementing the structure and presentation of the website was more detailed resulting in the development of prototype used to test how the front-end interacts with the back-end. Developing the prototype provides a proper interpretation of the system requirements and functionality and helps to ensure that the proposed solution does what it is supposed to do. In addition, the Agile SDLC model implemented allows for a constant and continuous analysis, development and testing of the software which allows for any issues discovered to be resolved accordingly.

A thorough testing was performed to ensure that the application perceived value is measured against the requirements. Test cases were developed and implemented while further User Defined Testing was carried out using members of the focus group. This allows the developer to explore ideas and exchange feedback with the stakeholder and end-user with the aim of understanding if the developed solution is fit for purpose, does what it need to do, and does it well.

### 7.3 Results Assessment

The functional and non-functional requirements compiled were prioritised and implemented according to its significance. However, it must be noted that some focus group recommendations were discarded after further discussions with the primary stakeholder. This is to ensure that the required attention is given to core functionality, and that the application is developed efficiently and completed on time. Result gained from both the functional and User Defined Testing confirmed that user requirements had been met as the outcome was positive indicating a successful implementation. A total of 30 test cases were implemented with a 97% pass rate and 3% fail which has since been resolved. Also, the average satisfaction from the five member of the focus group was 4.7% which indicate that the users enjoyed using the application.



## 7.4 Experience gained and Issues encountered

### 7.4.1 Experience Gained

The entire process of the project from the initial research, through the stages of analysis, design, development, testing and evaluation, has been complex with a steep learning curve. Most importantly, experience gained throughout the project can be outlined as follows:

- The importance of project and time management
- Experience in Agile model of SDLC
- Experience of front-end and back-end languages
- Experience in developing test cases

### 7.4.2 Issues Encountered

The main challenges were associated with complexity of the core functionality and finding an up-to-date tutorial and resources to use in the development process. Most of the tutorial available on YouTube were outdated and only cover older versions of programming languages especially on MySQL implementation. Another challenge is the level of knowledge required to be able to develop a robust application within a limited time.

## 7.5 Recommendations

Throughout the development of this application, potential features which could be implemented for future development includes not only new ideas but ideas discarded during the development of the final application. These recommendations include the following:

- The website could be further developed to provide a more interactive environment by allowing users to discuss projects or laboratory activities in a forum.
- As the system was custom created for lectures and students of the School of Computer and Mathematics, minimal consideration was given to user role. The system could be further developed to be more flexible if more roles were to be created.
- The barcode system could be further developed by implementing a scanner device which could be integrated with the barcode system within the application.
- Due to time constraint and the fact that stakeholder is the super-Admin user, email configuration was embedded into the HTML body readily configured. The stakeholder is notified of all user registrations, issues and returns activities at any period of time. The system could be developed further by providing the configuration within the UI to create more flexibility.

- When accessing the database, the system uses MySQLi which supports MySQL only. The system could be further developed using PDO which supports twelve different drivers making switching to another database easy and transparent.

## 7.6 Summary

The project was aimed at providing a solution to the existing inventory management problems within the school of computer and Mathematics. A suitable design was created based on stakeholder's requirements, research into end-user and existing solutions in the market. The application was successfully implemented and efficiently tested by member of the focus group. Also using the agile model encourages continuous testing of each requirements and a final functional testing conducted using test cases.

All the set objectives were achieved along with the core functionality required by the stakeholder and some recommendations from focus group have been excluded for good reason. The successful completion of this project has created a robust inventory management system which not only support multiple users, multiple browsers, and multiple operating systems which can adjust to different size of screen device. It has created an easy to use monitoring and tracking application for the stakeholder/end-user through the configured email notification facility and issue/return functionality. Finally, some notable recommendations which would further enhance the effectiveness and flexibility of future versions of the application have been made.

## References

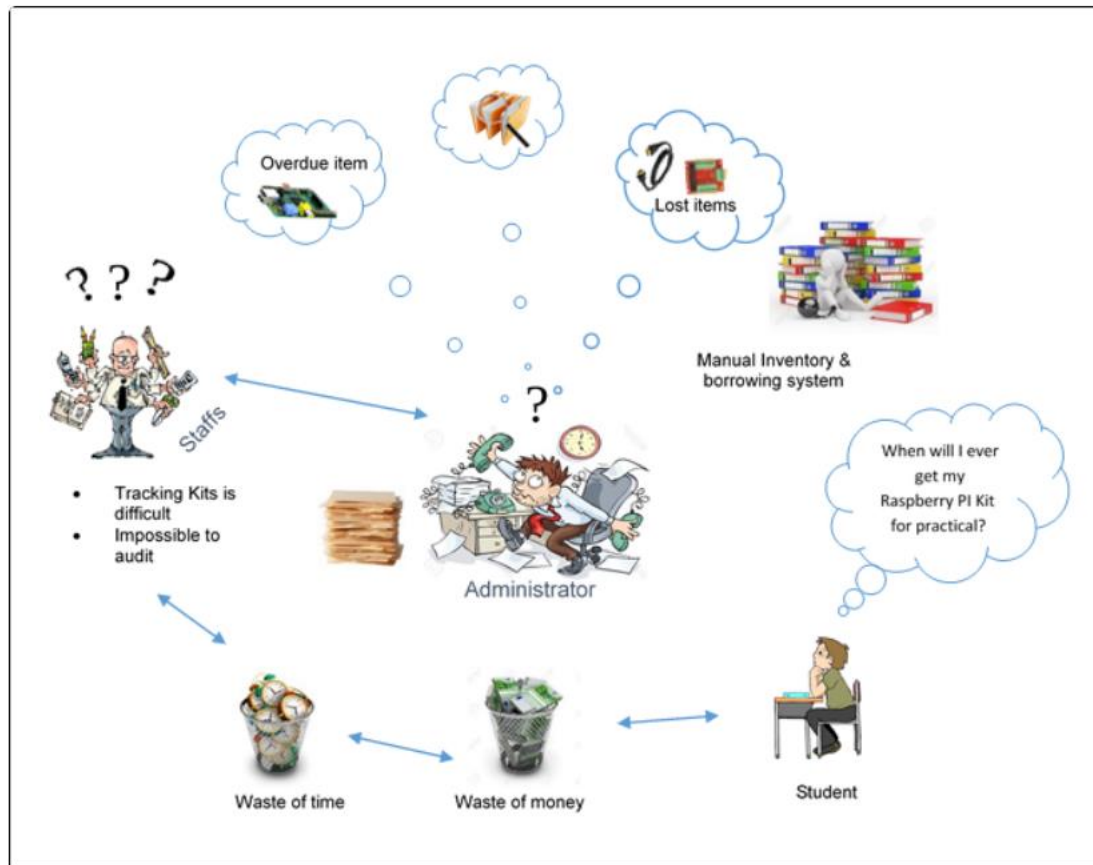
- BBC (2016) *Bitesize*. Available at: <http://www.bbc.co.uk/education/guides/zdsbwmn/revision> (Accessed: 12 January 2016).
- Dix, A., Finlay, J.E. and Abowd, G. (2003) *Human-computer interaction*. New York: Prentice Hall
- *Features* (2015) Available at: <http://www.webhostingbest10.com/best-websitehosting> (Accessed: 6 January 2016).
- Heath, N. (2013) '*We thought we'd sell 1, 000': The inside story of the raspberry pi*'. Available at: <http://www.zdnet.com/article/we-thought-wed-sell-1000-the-insidestory-of-the-raspberry-pi/> (Accessed: 29 December 2015).
- James, M. (2008) *Agile Methodology*. Available at: <http://agilemethodology.org/> (Accessed: 8 January 2016).
- James, M. (2012) *Introduction to Scrum and agile for knowledge creating companies*. Available at: [http://scrumtrainingseries.com/Intro\\_to\\_Scrum/Intro\\_to\\_Scrum.htm](http://scrumtrainingseries.com/Intro_to_Scrum/Intro_to_Scrum.htm) (Accessed: 8 January 2016).
- Kay, R. (2002) *System development life cycle*. Available at: <http://www.computerworld.com/article/2576450/app-development/appdevelopment-system-development-life-cycle.html> (Accessed: 8 January 2016).
- Lazarte, M. (2012) *W3C web content accessibility guidelines become ISO/IEC international standard*. Available at: [http://www.iso.org/iso/home/news\\_index/news\\_archive/news.htm?refid=Ref1670](http://www.iso.org/iso/home/news_index/news_archive/news.htm?refid=Ref1670) (Accessed: 29 December 2015).
- Nielsen, J. (2003) *Ten good deeds in web design (Alertbox Oct. 1999)*. Available at: <http://www.arraydev.com/commerce/delibadmin/TenGoodDeeds.htm> (Accessed: 8 January 2016).
- Palani, G. (2011) *Summary of web application testing methodologies and tools*. Available at: <http://www.ibm.com/developerworks/library/wa-webapptesting/> (Accessed: 12 January 2016).
- Purch.com (2015) *Best inventory management software of 2016*. Available at: <http://inventory-software-review.toptenreviews.com/> (Accessed: 5 January 2016). Inline Citation: (Purch.com, 2015)

- Rogers, Y., Sharp, H.M. and Preece, J. (2011) *Interaction design: Beyond human - computer interaction*. 3rd edn. Oxford, United Kingdom: John Wiley & Sons.
- Shneiderman, B., Plaisant, C., Cohen, M. and Jacobs, S. (2009) *Designing the user interface: Strategies for effective human-computer interaction*. 5th edn. Boston: Addison-Wesley Educational Publishers.
- Software Advice (2016) *Best inventory management software of 2016*. Available at: <http://inventory-software-review.toptenreviews.com/> (Accessed: 29 December 2015).
- Top Ten Reviews (2016) Available at: <http://inventory-software-review.toptenreviews.com/> (Accessed: 29 December 2015).
- *User interface design basics* (2014) Available at: <http://www.usability.gov/what-andwhy/user-interface-design.html> (Accessed: 8 January 2016).
- Ward, M. (2013) *Raspberry pi and the rise of small computers*. Available at: <http://www.bbc.co.uk/news/technology-21588419> (Accessed: 18 June 2016).
- w3shool (2016) *W3Schools online web tutorials*. Available at: <http://www.w3schools.com/default.asp> (Accessed: 18 June 2016).

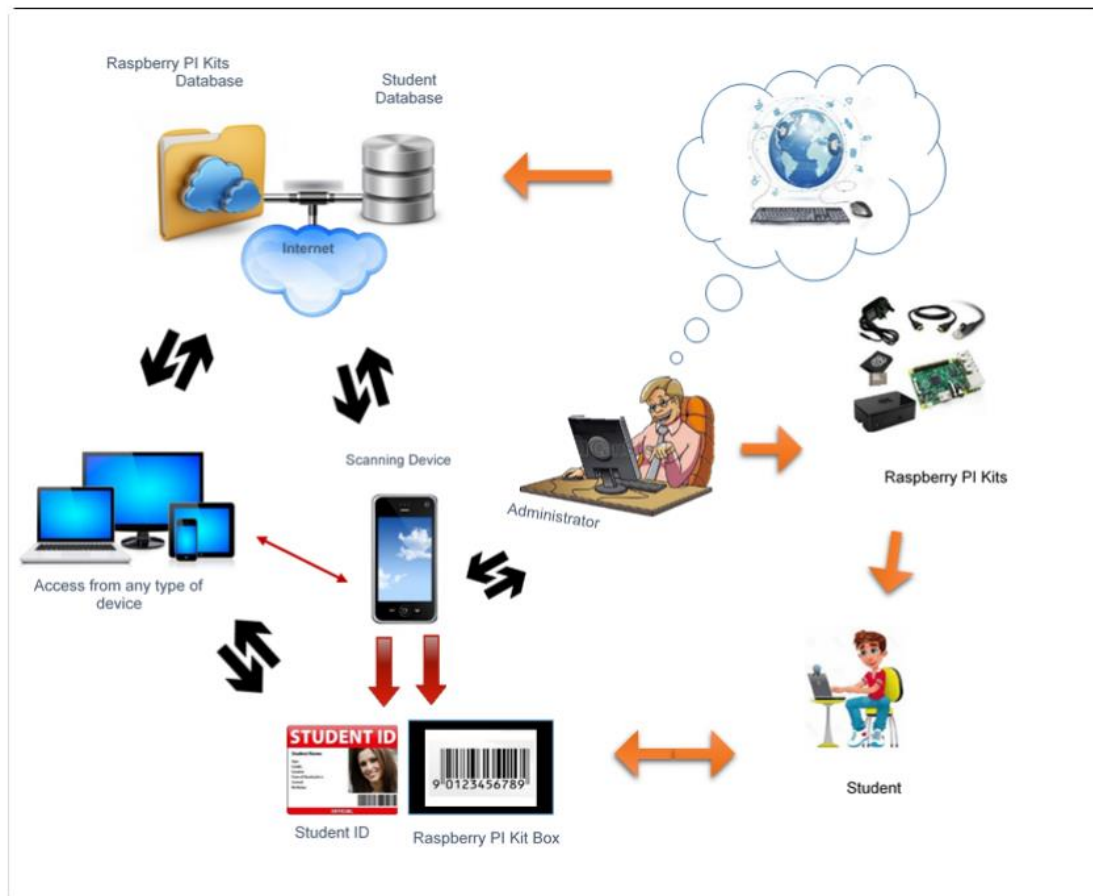
## Appendices

### Appendix One: Rich Pictures Outlining Current Problem and Proposed Solution

#### 1.1: Current Problem




## 1.2: Proposed Solution



## Appendix Two: Example of Completed Research Questionnaire

### 2.1: Example Questionnaire



**Information sheet**

Project Student: Abiodun Oke  
Email: Oke-A@email.ulster.ac.uk  
Tel: 07758851376

Project Supervisor: Dr Giuseppe Trombino  
Email: g.trombino@ulster.ac.uk  
Tel: +44 (0) 28 9036 6505

I would like to invite you to take part in a questionnaire aimed at the design, implementation and testing of a new website. The Ulster University School of Computing and Mathematics has recently acquired a number of Raspberry Pi development Kits, which are intended to be borrowed by staffs and students to use in laboratory sessions or projects. This website shall provide Inventory management for the School as a way of tracking the usage and to keep current inventory of these kits. The website will be accessible from any device which include desktop, laptop, tablets and mobile phones.

This questionnaire is being carried out in partial fulfilment of my MSc Professional Software Development course project at University of Ulster School of Computing & Information Engineering. Your involvement would help greatly and would be much appreciated.

This study involves questions relating to the preferred design of this new website and more specifically:

1. The website content.
2. The features offered by the website.
3. The presentation of the website.

Questionnaires are provided to a selection of individuals and it contains 10 questions which should take approximately not more than 15 minutes of your time to complete.

Please be assured that if you agree to partake in this study, under the Freedom of Information Act, you will have the right of access to data which you have contributed. Every effort will be taken to ensure that all personal data, which may identify participants in this study, will be removed from information that will be gathered for the purposes of this research.

Hard copy data will be stored securely only until the successful completion of this programme of study and thereafter safely destroyed. Confidentiality is a vital part of this research. In the final write-up any references or quotations used from questionnaires will be anonymous.

It is emphasised that this research is for the sole purpose of the course of academic study being undertaken by the student at University of Ulster.

If you are interested in taking part please complete the attached questionnaire and return by email to Oke-A@email.ulster.ac.uk. Should you have any additional queries at any point in the process, please contact my project supervisor Dr Giuseppe Trombino at University of Ulster School of Computing & Mathematics (contact details above).

Please be aware that participation in this research is NOT compulsory and even if you consent to take part you have the right to withdraw from the study at any time, without prejudice, and request that any information you have contributed be removed from the record.

I would like to thank you for your time in reading this information sheet and would be very grateful if you decide to participate in this study.

Many thanks,  
Abiodun Oke

### Introduction

Q1 Please indicate which of the following best describes your role:

**Please tick all that apply**

☒ Teaching staff      ☐ Non-teaching staff      ☐ Student      ☐ None

Q2

- a) Please indicate which of the following Inventory Management software are you aware of?  
b) Please indicate which of the following Inventory Management software you have used before?

**Please tick all that apply**

	Aware of	Not aware of	Have used	Never used
tradegecko (www.tradegecko.com)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
inFlow (www.inflowinventory.com)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
iMagic Inventory (www.imagicinventorysoftware.com)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Inventory Tracker Plus (www.inventory-tracker.com)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
stitchlabs (www.stitchlabs.com)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Others (please specify) <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Q3 Please indicate the extent to which you agree or disagree with the following statements:

**Please choose an item from the drop down menu**

- Teaching staff would benefit from using an inventory website: Agree
- Students would benefit from using the inventory website Agree

### Website Content and Features

Q4 Please rate the suitability of the following features for a Raspberry Pi Kit Inventory website:

**Please choose an item from the drop down menu**

- 🚩 Low inventory warning (Very suitable)
- 🚩 View item history (Neither suitable nor unsuitable)
- 🚩 Barcode scan adding (Very suitable)
- 🚩 Email reminder or notification (Very suitable)

🚩 Are there any other features you would like to suggest:



Q5 Please rate the importance of the following features for the Raspberry PI Inventory website:  
*Please choose an item from the drop down menu*

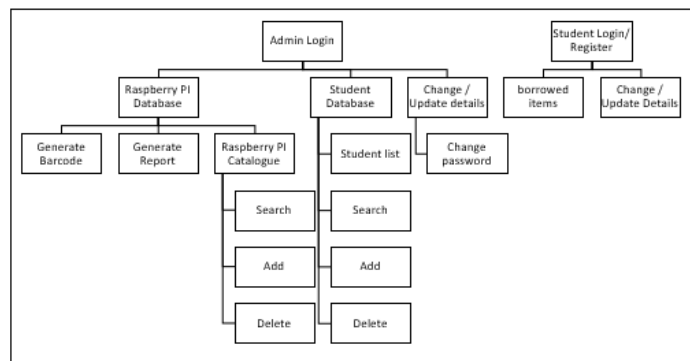
**1. Standard Features**

- a) Admin Login (Very important)
- b) Student Login (Very important)
- c) Staff Login (Very important)

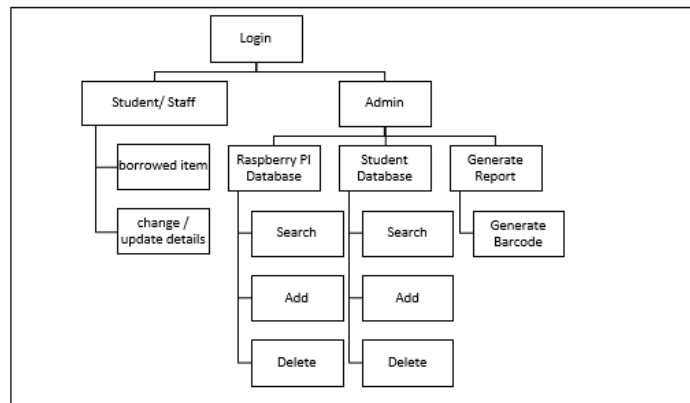
**2. Management Features**

- A. Search / Filter (Neither important nor unimportant)
- B. Tracking (Quite important)
- C. Barcode /QR generator (Quite important)

Q6 Please rate the suitability of the following content structures for the website:  
*Please choose an item from the drop down menu*



**Structure A:** Very suitable



**Structure B:** Quite suitable

Q7 Please rate the suitability of the following features for the Raspberry Kit Inventory website:  
*Please choose an item from the drop down menu*



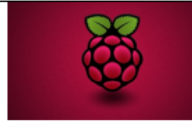

- 1) User Login (with password reset) (Very suitable)
- 2) Backup Database (Neither suitable nor unsuitable)
- 3) Raspberry Kit Catalogue (Very suitable)
- 4) Website Designer and Maintenance Contact (Neither suitable nor unsuitable)
- 5) Review Section (for users to post reviews) (Neither suitable nor unsuitable)
- 6) Discussion Board (for users to discuss project) (Neither suitable nor unsuitable)
- 7) Are there any other features you would like to suggest:

### Website Presentation

Q8 Please indicate which of the following text font properties is suitable for the website:  
*Please select an option from the radio button*

Font	Description	Example	
Century Gothic	A design based on Monotype 20th Century, Useful for headlines and general display work and for small quantities of text, particularly in advertising	F	<input type="radio"/> Option 1
Times New Roman	This remarkable typeface first appeared in 1932 in The Times of London newspaper, for which it was designed. Widely used in books and magazines, for reports, office documents and also for display and advertising	F	<input checked="" type="radio"/> Option 2
Comic Sans MS	Designed by Microsoft's Vincent Connare, this is a face based on the lettering from comic magazines.	F	<input type="radio"/> Option 3
Are there any other font you would like to suggest:		A range/mix of fonts might be	

Q9 Please indicate the background image for the website:  
*Please tick all that apply*

			
<input type="checkbox"/> Pattern	<input checked="" type="checkbox"/> Bright Colour	<input type="checkbox"/> Raspberry Pi	<input checked="" type="checkbox"/> Raspberry Pi Kit

### Website Testing

Q10 Please indicate if you would be willing to test the website once available?

*Please tick one box*

I am willing to test the website when available	<input type="checkbox"/> YES <input checked="" type="checkbox"/> NO
Please include your name and email for contact purposes when the website is available	Name: <input type="text"/>  Email: <input type="text"/>

\* Please note that any name provided will not be transcribed anywhere else. It will only allow me to contact you once the website is available in order to test it. Also all completed forms will be destroyed once the project is completed.

#### Free Comments

Please include below any comments you wish to make related to the design of this new website:

Re Qn 6: I feel some refinement (option C) may be preferable. You might explore further such architecture-related considerations with stakeholder(s).  
  
Re Qn 9: several of the illustrative proposals seem very dominant, likely to hinder usage.

## Appendix Three: Wireframe Design

### 3.1: Login page

The wireframe shows a browser window titled "IMS | Inventory Manager". The navigation bar includes links for "inventory Manager", "Home", "Uni Portal", "C & E Portal", "Help", "Login", and "Sign up". The main content area features a login form with a "Logo" placeholder, "University ID" and "Password" input fields, a "Remember me" checkbox, a "Forgot Password?" link, and "Login" and "Sign up" buttons.

### 3.2: User's Profile page

The wireframe shows a browser window titled "IMS | Inventory Manager". The top bar includes "IMS ADMIN" and "ADMIN" with a user icon. A left sidebar contains links for "Dashboard", "Users", "Items", "Manage Items", and "Report". The main content area displays a user profile for "User". It includes a profile picture placeholder with "Change Photo" and "Ok" buttons, a "Hello User" greeting, a "USER" role badge, a "Status: Active" indicator, and a "Member Since: 01-01-2016" date. An "ABOUT" section shows "First Name: User", "Last Name: User", "User Name: Admin", and "Email: admin@mail.com" with an "Edit" button.

### 3.3: Manage Items Page

IMS | Inventory Manager

IMS ADMIN

ADMIN

Dashboard

Users

Items

Manage Items

Report

Export CSV

search

Username	Barcode ID	Item Title	Qty Issued	Issued Date	Due Date	Issue Status	Issued By	Return Item
e00112233	IID001032	14" Inspiron	5	25-07-2016 11:37:47	31-07-2016 12:00:00	Confirmed	Admin	Return
e00112233	IID001032	14" Inspiron	5	25-07-2016 11:37:47	31-07-2016 12:00:00	Confirmed	Admin	Return
e00112233	IID001032	14" Inspiron	5	25-07-2016 11:37:47	31-07-2016 12:00:00	Confirmed	Admin	Return

### 3.4: Report Page

IMS | Inventory Manager

IMS ADMIN

ADMIN

Dashboard

Users

Items

Manage Items

Report

Export CSV

search

Username	Barcode ID	Item Title	Qty Returned	Issued Date	Due Date	Return Date
e00112233	IID001032	14" Inspiron	5	25-07-2016 11:37:47	31-07-2016 12:00:00	30-07-2016 11:50:16
e00112233	IID001032	14" Inspiron	5	25-07-2016 11:37:47	31-07-2016 12:00:00	30-07-2016 11:50:16
e00112233	IID001032	14" Inspiron	5	25-07-2016 11:37:47	31-07-2016 12:00:00	30-07-2016 11:50:16

3.5: Manage Users Page

IMS | Inventory Manager

IMS ADMIN

ADMIN

Dashboard

Users

Items

Manage Items

Report

Select option

Delete selected

Apply

ADD User

ADD Admin

search

ID	First Name	Last Name	Email	Module	Year	Role	Status	Edit	Delete
e00112233	Janet	Doe	janet@mail.com	N/A	N/A	Academic	Approved	Edit	Delete
b00115588	John	Doe	john@mail.com	COM 101	2014-2015	Student	Approved	Edit	Delete
e00112233	Janet	Doe	janet@mail.com	N/A	N/A	Academic	Approved	Edit	Delete
b00115588	John	Doe	john@mail.com	COM 101	2014-2015	Student	Approved	Edit	Delete
e00112233	Janet	Doe	janet@mail.com	N/A	N/A	Academic	Approved	Edit	Delete
b00115588	John	Doe	john@mail.com	COM 101	2014-2015	Student	Approved	Edit	Delete

3.6: Change Password Page

IMS | Inventory Manager

IMS ADMIN

ADMIN

Dashboard

Users

Items

Manage Items

Report

Change Password

Old Password

New Password

Confirm new Password

Change Password

Cancel

### 3.7: Forgotten Password Page

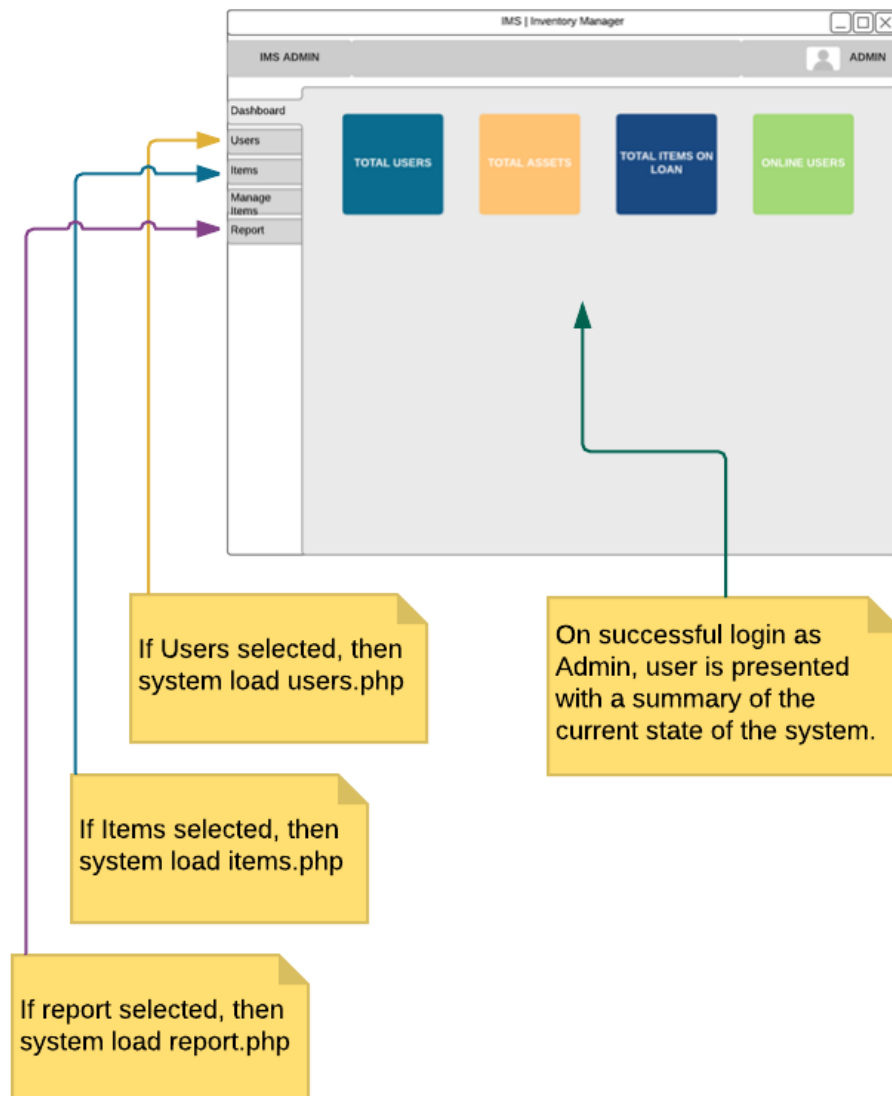
The screenshot shows a web browser window titled "IMS | Inventory Manager". The navigation bar at the top includes links for "inventory Manager", "Home", "Uni Portal", "C & E Portal", "Help", "Login", and "Sign up". The main content area features a central gray box with the heading "Enter your email to recover password". Inside this box, there is a text input field labeled "Your email", a blue button labeled "Get password", and a light blue button labeled "Login" preceded by the word "Or".

### 3.8: Sign up Page

The screenshot shows a web browser window titled "IMS | Inventory Manager". The navigation bar at the top includes links for "inventory Manager", "Home", "Uni Portal", "C & E Portal", "Help", "Login", and "Sign up". The main content area features a central gray box with a "Logo" icon in the top right corner. The box contains several form fields: "Select User Role" (a dropdown menu), "University ID", "First Name", "Last Name", "Select Year option" (a dropdown menu), "Module", "Your Email", "Password", and "Confirm Password". At the bottom of the box are two buttons: a blue "Sign up" button and a light blue "Login" button.

## Appendix Four: Storyboard

### 4.1: Example storyboard of Admin dashboard





## Appendix Five: Database Tables

### 5.1: Users Table

**Table: Users**

Column name	Key type	Data type	Comment
user_id {PK}	Primary	INT (11)	PRIMARY KEY (user_id)
user_name	No	VARCHAR (255)	University ID
User_firstname	No	VARCHAR (255)	User first name
User_lastname	No	VARCHAR (255)	User last name
user_email	No	VARCHAR (255)	email address
User_password	No	VARCHAR (255)	Login password
User_module	No	VARCHAR (255)	Module which user require asset
User_year	No	VARCHAR (255)	Year of entry into the institution
User_image	No	TEXT	Passport photo size image
User_status	No	TINYINT (1)	Active or inactive
User_created_on	No	DATE	Date time of creation
User_description	No	TEXT	A summary about the user

### 5.2: Item Categories Table

**Table: Item\_categories**

Column name	Key type	Data type	Comment
Item_cat_id {PK}	Primary	INT (11)	PRIMARY KEY
Item_cat_name	No	VARCHAR (255)	Name of the item

### 5.3: Items Table

**Table: Items**

Column name	Key type	Data type	Comment
Item_id {PK}	Primary	INT (11)	PRIMARY KEY (item_id)
Item_cat_id {FK}	Foreign	INT (11)	FOREIGN KEY (item_cat_id) REFERENCES Item_category(item_cat_id)
Item_barcode	Composite	VARCHAR (255)	Barcode for item
Item_name	No	VARCHAR (255)	Name of item
Item_description	No	TEXT	Description of the item

Item_qty_avl	No	INT (11)	Quantity available
Item_created_on	No	DATETIME	Date time of creation

#### 5.4: Issued Table

**Table: Issued**

Column name	Key type	Data type	Comment
issue_id {PK}	Primary	INT (11)	PRIMARY KEY (issue_id)
item_id {FK}	Foreign	INT (11)	FOREIGN KEY (item_id) REFERENCES Items(item_id)
user_id {FK}	Foreign	INT (11)	FOREIGN KEY (user_id) REFERENCES Users(user_id)
Issue_item_name	No	VARCHAR (255)	Name of issued item
Issued_item_barcode	No	VARCHAR (255)	Barcode of issued item
Qty_issued	No	INT (11)	Quantity of item issued to a user
Issued_date	No	DATETIME	Date time item borrowed/issued
Due_date	No	DATETIME	Due date of item borrowed
Issue_status	No	TINYINT (1)	If issued 1 otherwise 0

#### 5.5: Users online Table

**Table: Users\_online**

Column name	Key type	Data type	Comment
id {PK}	Primary	INT (11)	PRIMARY KEY
sessions	No	VARCHAR (255)	User session from session_id () in functions.php
time	No	INT (11)	Login time represented in integer

#### 5.6: Returned Table

**Table: Returned**

Column name	Key type	Data type	Comment
return_id {PK}	Primary	INT (11)	PRIMARY KEY (return_id)
issued_id {FK}	Foreign	INT (11)	FOREIGN KEY (issue_id) REFERENCES Issued(issue_id)
user_id {FK}	Foreign	INT (11)	FOREIGN KEY (user_id) REFERENCES Users(user_id)
item_name	No	VARCHAR (255)	Name of item returned
item_barcode	No	VARCHAR (255)	Barcode of item returned

Qty_returned	No	INT (11)	Quantity of item returned by user
Item_issued_date	No	DATETIME	Date time item borrowed/issued
Item_due_date	No	DATETIME	Due date of item borrowed
Return_date		DATETIME	Date item returned
return_status	No	TINYINT (1)	If returned 1 otherwise 0

## Appendix Six: Script to prevent item on loan from being deleted.

Status	Issue Item	Edit	Delete
Available   156	<a href="#">Issue item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Available   10	<a href="#">Issue item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Available   5	<a href="#">Issue item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Available   5	<a href="#">Issue item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Available   1	<a href="#">Issue item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Available   1	<a href="#">Issue item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Available   1	<a href="#">Issue item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

Previous 1 Next

Delete button is disabled once item is on loan (\$loan\_count > 0) using the source code below

```

if(!$loan_count>0){
    echo "<td>
    <a class='btn btn-warning' href='items.php?source=edit_item&it_id=$item_id' <span class='fa fa-edit'></span></a>
    </td>" ;

    echo "<td>
    <a class='btn btn-danger' href='items.php?delete=$item_id' onclick=\"javascript:return confirm('Do you want delete ?')\">
    <span class='fa fa-trash-o'></span></a>
    </td>" ;

    }else {
        echo "<td>
        <a class='btn btn-warning' href='items.php?source=edit_item&it_id=$item_id' <span class='fa fa-edit'></span></a>
        </td>" ;

        echo "<td>
        <a class='btn btn-danger disabled' href='items.php?delete=$item_id' onclick=\"javascript:return confirm('Do you want dele
        <span class='fa fa-trash-o'></span></a>
        </td>" ;
    }

```

## Appendix Seven: Script to prevent duplicate item import into database

CSV Import Failed,Reason:Duplicate Entry in CSV files [View all Items](#)

```
53 <?php
54 if(isset($_POST['import_csv'])){
55     global $connection;
56     $file = $_FILES['file']['tmp_name'];
57
58     $handle = fopen($file,"r");
59     $item_cat_id = $_POST['item_category'];
60     $item_created_on = date('Y-m-d\TH:sP');
61
62     $query = "SELECT MAX(item_id) FROM items ";
63     $result = mysqli_query($connection,$query);
64
65
66
67     $row = mysqli_fetch_row($result);
68     $highest_id = $row[0];
69
70
71     while(($fileop = fgetcsv($handle,1000,"")) != false){
72
73
74         $item_name      = $fileop[0];
75         $item_description = $fileop[1];
76         $item_qty_avl    = $fileop[2];
77         $item_barcode     = 'IID'.sprintf('%04d',$item_cat_id).('++$highest_id);
78
79         $select_item     = "SELECT * FROM items WHERE item_name = '$item_name' ";
80         $items_result     = mysqli_query($connection,$select_item);
81         confirmQuery($items_result);
82
83         if(mysqli_num_rows($items_result)){
84
85             $query = "INSERT INTO items(item_category_id,item_name ,item_barcode,item_description,item_qty_avl,item_created_on)
86             VALUE('$item_cat_id','$item_name','$item_barcode','$item_description','$item_qty_avl','$item_created_on') ";
87             $result = mysqli_query($connection,$query);
88
89
90         }else {
91             $message = "<p class='alert alert-danger text-centre'>
92             CSV Import Failed,Reason:Duplicate Entry in CSV files <a href='items.php' class='link' > View all items</a></p>";
93         }
94
95
96         if(!$result){
97             die("QUERY FAILED");
98
99         }else{
100             echo "Data uploaded successfully" ;
101         }
```

php source code to ensure that multiple instances of same data are not duplicated

Error message display if user try to import a csv containing data that are already in database

## Appendix Eight: Testing

### 8.1: Example Test Case (See Appendix Ten for ALL the Test cases)

#### Example 1: PASS

##### Project Name: IMS | Inventory Manager

Test Case ID	IMS_001	Test Case Description	Test the Login functionality in IMS		
Created By	Abi	Test Priority (Low/Medium/High)	High	Version	2.0

Tester's Name	Abi	Date Tested	15-August-2016	Test Case (Pass/Fail/Not Executed)	PASS
---------------	-----	-------------	----------------	------------------------------------	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	Username = admin
2	Password = password
3	
4	

Test Scenario	Verify on entering valid username and password, the user can login
---------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to login link from homepage	Login page should open, credential can be entered and User is logged in.	User is logged in.	Pass
2	Provide valid username			Pass
3	Provide valid password			Pass
4	Click on login button			Pass

#### Example 2: FAIL

Test Case ID	IMS_010	Test Case Description	Test USERS functionality		
Created By	Abi	Test Priority (Low/Medium/High)	High	Version	2.0

Tester's Name	Abi	Date Tested	15-August-2016	Test Case (Pass/Fail/Not Executed)	FAIL
---------------	-----	-------------	----------------	------------------------------------	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	As required on form
2	Last Name = O'Neil
3	
4	

Test Scenario	Verify "ADD User / Admin" functionality to Manage Users work with special characters
---------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to ADD User	Add user page should open and new user details should be saved.	Error message displayed on output	Pass
2	Click the Add user button			Pass
3	Provide user details			Fail
4	Click save button			Fail

### 8.2: Functional Testing Results

#### Test Results

Status	Total	%
PASS	29	97
FAIL	1	3
NOT EXECUTED	0	0
SUSPENDED	0	0
TOTAL TEST CASES	30	100

### 8.3: Example User Testing Questionnaire



#### Information sheet



Project Student: Abiodun Oke  
Email: Oke-A@email.ulster.ac.uk  
Tel: 07758851376

Project Supervisor: Dr Gaye Lightbody  
Email: g.lightbody@ulster.ac.uk  
Tel: +44 (0) 28 903 66574

I would like to invite you to take part in a questionnaire aimed at the design, implementation and testing of a new website. The Ulster University School of Computing and Mathematics has recently acquired a number of Raspberry Pi development Kits, which are intended to be borrowed by staffs and students to use in laboratory sessions or projects. This website shall provide Inventory management for the School as a way of tracking the usage and to keep current inventory of these kits and other assets. The website will be accessible from any device which include desktop, laptop, tablets and mobile phones.

This questionnaire is being carried out in partial fulfilment of my MSc Professional Software Development course project at University of Ulster School of Computing & Information Engineering. Your involvement would help greatly and would be much appreciated.

This study involves questions relating to the usability testing of this new website and more specifically:

1. The website content.
2. The features offered by the website.
3. The presentation of the website.

Questionnaires are provided to a selection of individuals and it contains 5 Tasks which should take approximately not more than 15 minutes of your time to complete.

Please be assured that if you agree to partake in this study, under the Freedom of Information Act, you will have the right of access to data which you have contributed. Every effort will be taken to ensure that all personal data, which may identify participants in this study, will be removed from information that will be gathered for the purposes of this research.

Hard copy data will be stored securely only until the successful completion of this programme of study and thereafter safely destroyed. Confidentiality is a vital part of this research. In the final write-up any references or quotations used from questionnaires will be anonymous.

It is emphasised that this research is for the sole purpose of the course of academic study being undertaken by the student at University of Ulster.

If you are interested in taking part please complete the attached questionnaire and return by email to Oke-A@email.ulster.ac.uk. Should you have any additional queries at any point in the process, please contact my project supervisor Dr Gaye Lightbody at University of Ulster School of Computing & Mathematics (contact details above).

Please be aware that participation in this research is NOT compulsory and even if you consent to take part you have the right to withdraw from the study at any time, without prejudice, and request that any information you have contributed be removed from the record.

I would like to thank you for your time in reading this information sheet and would be very grateful if you decide to participate in this study.

Many thanks,

Abiodun Oke



<b>Name:</b>		
<b>Email</b>		
<b>Role</b>	Academic <input type="checkbox"/>	Student <input checked="" type="checkbox"/>

The following tasks focus on testing the website in various areas of functionality with each task in segments. As each task is completed, please rate the task in terms of difficulty by selecting the appropriate answer. Please, follow each step carefully and refer to the annotated screenshots for guidance.

### Task 1: Homepage

To complete the following task, please follow the steps provided.

Step 1: Copy the link below into your browser. The page should appear identical to Figure 1 below.

<a href="https://dunluce.infoc.ulst.ac.uk/B00668298/COM810/ims/index.php">https://dunluce.infoc.ulst.ac.uk/B00668298/COM810/ims/index.php</a>
---

**Please NOTE:** The server hosting the website is the Ulster University server (student domain area) and you might be prompted with a message *"There is a problem with this website's security certificate"* OR *"Your connection is not private"*. Be rest assured there is no threat to your system. Proceed to *"Continue to this website"* OR *"Advance, then Proceeds to dunluce.infoc.ulst.ac.uk"*.

Thank you for your understanding



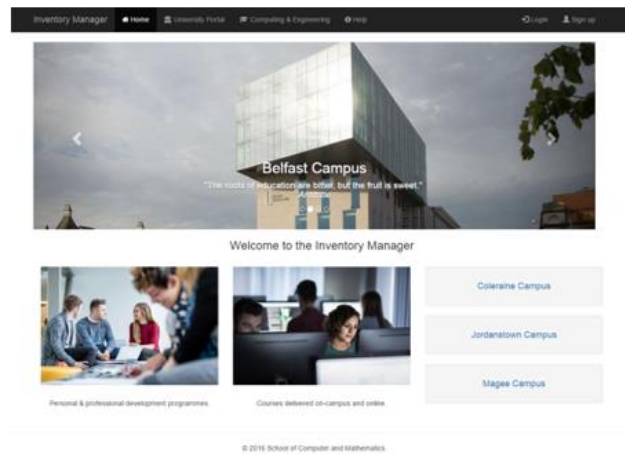


Figure 1: Screenshot of the Homepage

Step 2: Click any of the links and browse through the homepage

What is your first impression of the site?

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Comments/Feedback

Very professional user interface, features are not cluttered together. Aesthetically appealing, and all features are clearly visible to user.

## Task 2: Creating a user account

To complete the following task, please follow the steps provided.

Step 1: Select Sign up from the top right corner from the navigation bar. The page should appear identical to Figure 2 below.

A screenshot of a web application's 'Register' page. The page has a dark header with navigation links: 'Inventory Manager', 'Home', 'University Portal', 'Computing & Engineering', 'Help', 'Login', and 'Sign up'. The main content area is a light gray box with the title 'Register' and the 'Uttar University' logo. The form includes fields for 'Student' (a dropdown menu), 'University ID', 'First Name', 'Last Name', 'Select option' (a dropdown menu), 'Module', 'Your Email', 'Password', and 'Confirm Password'. At the bottom of the form are two buttons: 'Sign up' and 'Login'.

Figure 2: Screenshot of the Sign up page

Step 2: Complete the registration form using any choice from the following details:	
Role: Academic	Student
University ID: use the word TEST and 4 random numbers	University ID: use the word TEST and 4 random numbers
First Name: Your first name	First Name: Your first name
Last Name: Your last name	Last Name: Your last name
Year: N/A	Year: Select any year
Module: NOT APPLICABLE	Module: COM 101
Email: your email	Email: your email
Create and confirm password	Create and confirm password

Step 3: Click on Signup button

Rate your experience in completing the registration form.

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Comments/Feedback

All details that are required are clearly explained to the user, and the layout of the screen is both clear and aesthetically appealing. Very useful error dialogs used, for example if passwords don't match, or username already taken. Stops errors occurring before submit details. Very impressive.

### Task 3: Admin Login

To complete the following task, please follow the steps provided.

Step 1: Select Login from the top right corner from the navigation bar. The page should appear identical to Figure 3 below.

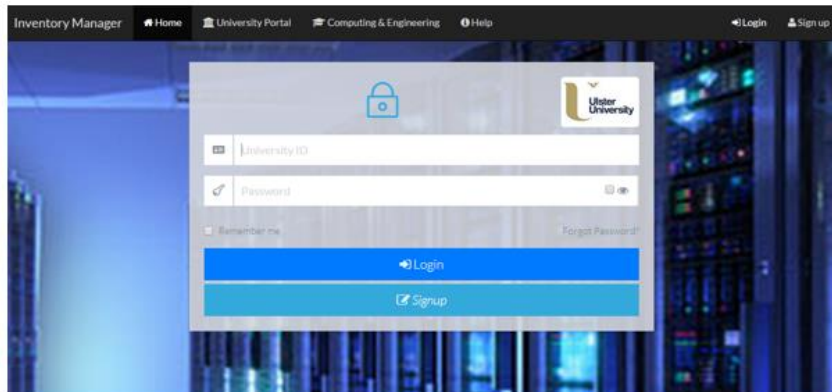


Figure 3: Screenshot of the Login page

Step 2: Complete the Login form using the following details:

University ID	subadmin
Password	password

Step 3: Click the Login button. The page should appear identical to Figure 4 below.

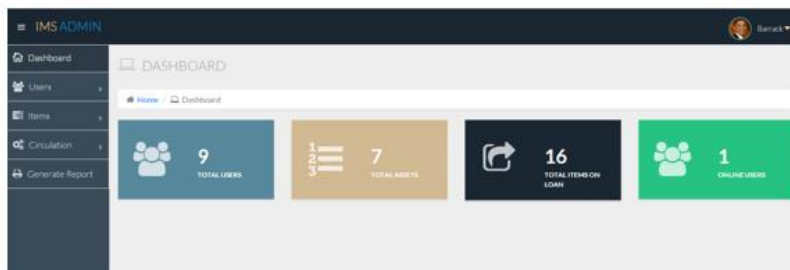


Figure 4: Screenshot of the subadmin dashboard

Rate your experience in completing the Login form.

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Comments/Feedback

Sections in the Admin login section (eg Total Users, Total Asset etc) clearly differentiated from all other sections. The counter of total users, total assets etc a nice touch, informing user of totals without having to enter the sections themselves, which is much more convenient for the user.

### 3.1: Approve a user

To complete the following task, please follow the steps provided.

Step 1: Click "TOTAL USERS" from the dashboard. The button should appear identical to Figure 5 below.

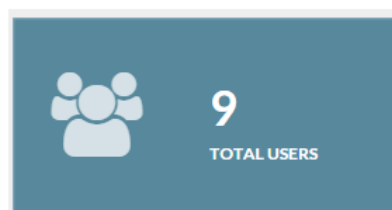


Figure 5: Screenshot of the TOTAL USERS

Step 2: Click "Approve" from the Status column for the account you created earlier. The page should appear identical to Figure 6 below.


	Image	University ID	First Name	Last Name	Email	Module	Year	User Role	Status	Edit	Delete
		test1023	John	Doe	j@mail.com	NOT APPLICABLE	NOT APPLICABLE	Academic	Approve	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure 6: Screenshot of the new account

Rate your experience.

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Comments/Feedback

The layout of this screen was particularly usable, both in terms of screen layout and functionality. The various colours used for the buttons are visually appealing, while all options available to the user are clearly highlighted immediately. Easy to use.

### 3.2: [view Item](#)

To complete the following task, please follow the steps provided.

Step 1: Click "TOTAL ASSETS" from the dashboard. The button should appear identical to Figure 7 below.

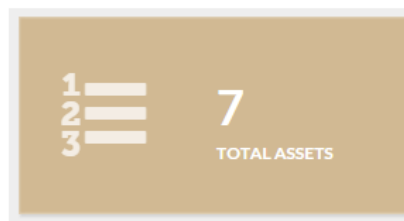


Figure 7: Screenshot of the TOTAL ASSETS

Rate your experience.

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

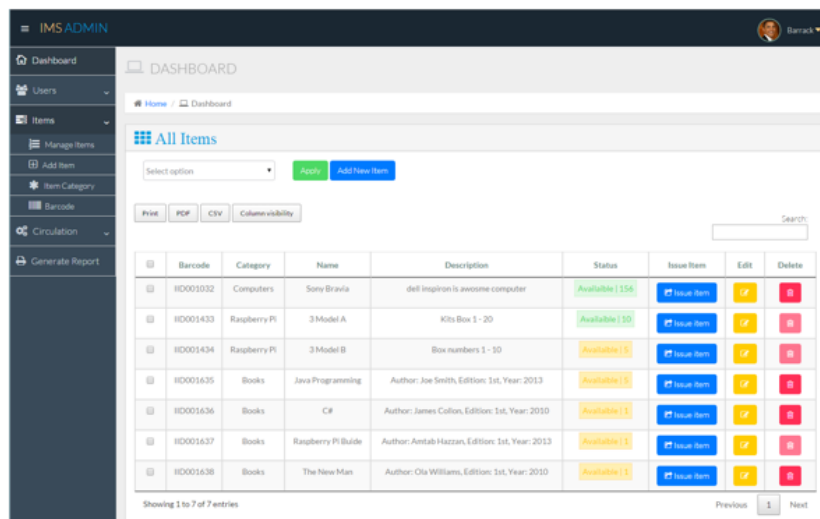
## Comments/Feedback

The drop-down menu along the top of the screen made returning to the dashboard and accessing this section very straightforward.

## Task 4: Issue Item

To complete the following task, please follow the steps provided.

Step 1: From the side navigation, select Items, then Manage Items. The page should appear identical to Figure 8 below.



Barcode	Category	Name	Description	Status	Issue Item	Edit	Delete
RD001032	Computers	Sony Bravia	dell inspiron is awesome computer	Available   156	<a href="#">Issue Item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
RD001433	Raspberry Pi	3 Model A	Kits Box 1 - 20	Available   10	<a href="#">Issue Item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
RD001434	Raspberry Pi	3 Model B	Box numbers 1 - 10	Available   5	<a href="#">Issue Item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
RD001635	Books	Java Programming	Author: Joe Smith, Edition: 1st, Year: 2013	Available   5	<a href="#">Issue Item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
RD001636	Books	C#	Author: James Coffin, Edition: 1st, Year: 2010	Available   1	<a href="#">Issue Item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
RD001637	Books	Raspberry Pi Guide	Author: Antab Hazzan, Edition: 1st, Year: 2013	Available   1	<a href="#">Issue Item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
RD001638	Books	The New Man	Author: Oia Williams, Edition: 1st, Year: 2010	Available   1	<a href="#">Issue Item</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

Figure 8: Screenshot of the Manage Items

Step 2: From the Issue Item column, select any Items by clicking the “issue item” button.  
The page should appear identical to Figure 9 below.

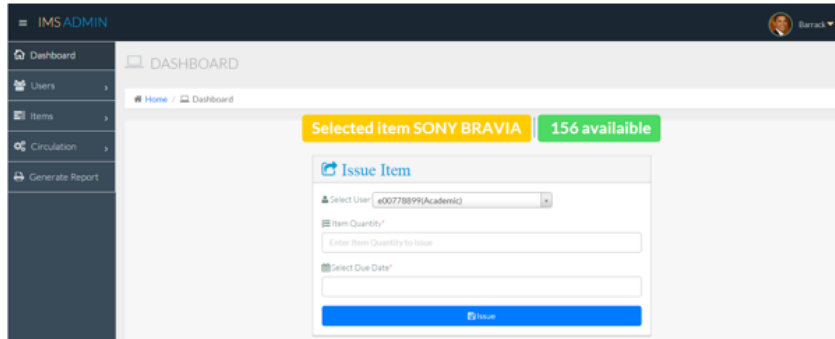


Figure 9: Screenshot of the Issue Item page

Step 3: From Select User drop down, select the user created earlier from the list of users.

Step 4: Enter Item Quantity.

Step 5: Select due date and time.

Step 6: Click “Issue” button.

Step 7: From the top right corner, select the dropdown arrow beside account name. Click “LOG OUT”

Rate your experience.

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>



#### Comments/Feedback

The different colours used for the various remaining quantities of an item (orange for low quantity, green for higher quantity) is a good idea. Being able to select the due date and the quantity is straightforward, and displaying the quantity of the remaining items after this transaction is very useful. Impressive.

#### Task 5: New User Login

To complete the following task, please follow the steps provided.

Step 1: From the top navigation, select Login.

Step 2: Enter the University ID/password created earlier.

Step 3: From the top navigation, select Login.

Step 4: From the side navigation, select Issue History.

Step 5: From the top right corner, select the dropdown arrow beside account name. Click "LOG OUT"

Rate your experience.

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

#### Comments/Feedback

Very straightforward. The menu along the top allowed easy access to the login screen, and the Issue History section was clearly signposted. The whole website is very user friendly.



Figure 14: User Guide starts

## Installation

### Installation:

1. **Download** .zip package
2. **Extract** the package
3. **Upload** it to your server
4. **Create** a database in your MySQL (suggested name: inventory)
5. **Open** "db.php" located in includes and edit the following:  
`$db ['db_host'] = Your MySQL host name`  
`$db ['db_user'] = Your MySQL login name`  
`$db ['db_pass'] = Your MySQL login password`  
`$db ['db_name'] = Your MySQL database name`
6. **Browse** the package url via browser (<http://yourdomain.com/ims/index.php>)

## Demo

### ✦ Login as ADMIN

Username = admin

Password = password

### ✦ Login as ACADEMIC

Username = academic

Password = password

### ✦ Login as STUDENT

Username = student

Password = password

## Screenshots & Description

### Homepage

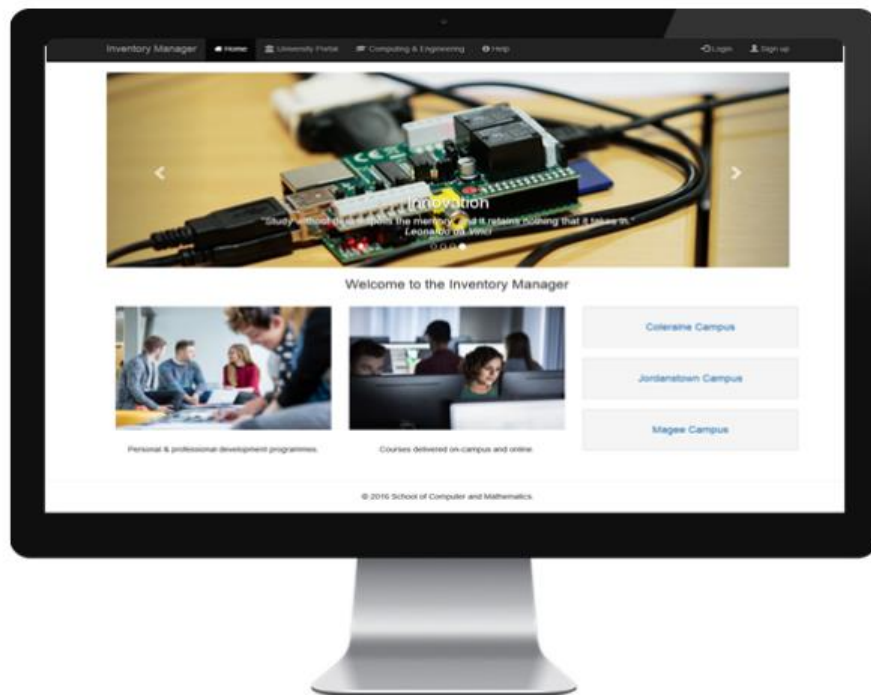


Figure 15: Homepage

### Login Page



Figure 16: Login Page

## Admin Dashboard

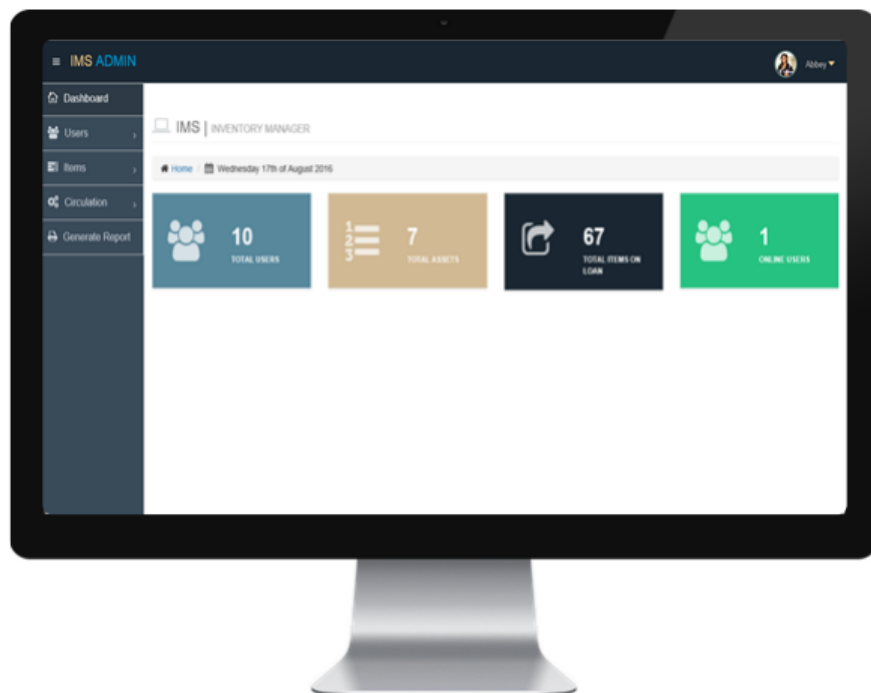


Figure 17: Admin dashboard

SIDE NAV	DASHBOARD	TOP NAV
<b>USERS</b> Manage Users Add User	<b>TOTAL USERS</b> Total number of approved users excluding Admin/sub-admin	<b>MY PROFILE</b>
<b>ITEMS</b> Manage Items Add Item Item Category Barcode	<b>TOTAL ASSETS</b> Total number of assets in the database	<b>CHANGE PASSWORD</b>
<b>CIRCULATION</b> Issue History Return History	<b>TOTAL ITEMS ON LOAN</b> Total number of items in circulation	<b>LOGOUT</b>
<b>GENERATE REPORT</b>	<b>ONLINE USERS</b> Users currently logged in	

Table 4: Admin Menu

## Manage Users

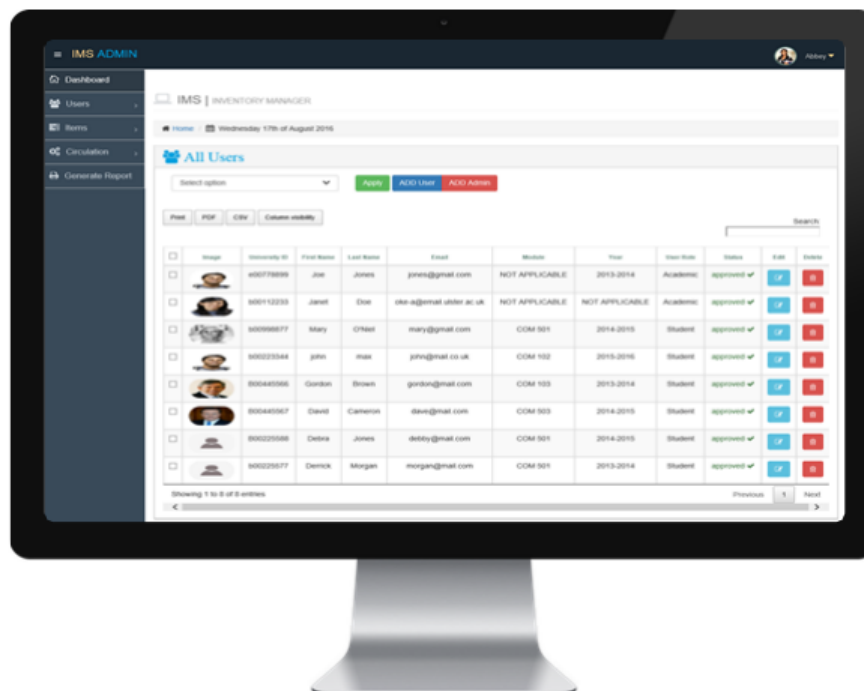


Figure 18: Manage Users

<ul style="list-style-type: none"> <li>View All Members</li> <li>Add New Member</li> <li>Add Admin</li> <li>Approve new user</li> <li>Edit member</li> </ul>	<ul style="list-style-type: none"> <li>Delete member</li> <li>Import member list (CSV)</li> <li>Export member list (CSV, PDF)</li> <li>Print member list</li> <li>Column visibility</li> </ul>
--	--

## Manage Items

<ul style="list-style-type: none"> <li>Add Item</li> <li>Issue Item</li> <li>Search Item</li> </ul>	<ul style="list-style-type: none"> <li>Edit Item</li> <li>Delete Item</li> <li>Item Status</li> </ul>	<ul style="list-style-type: none"> <li>Import Item list (CSV)</li> <li>Export Item list (CSV, PDF)</li> <li>Print Item list</li> </ul>
---	---	--

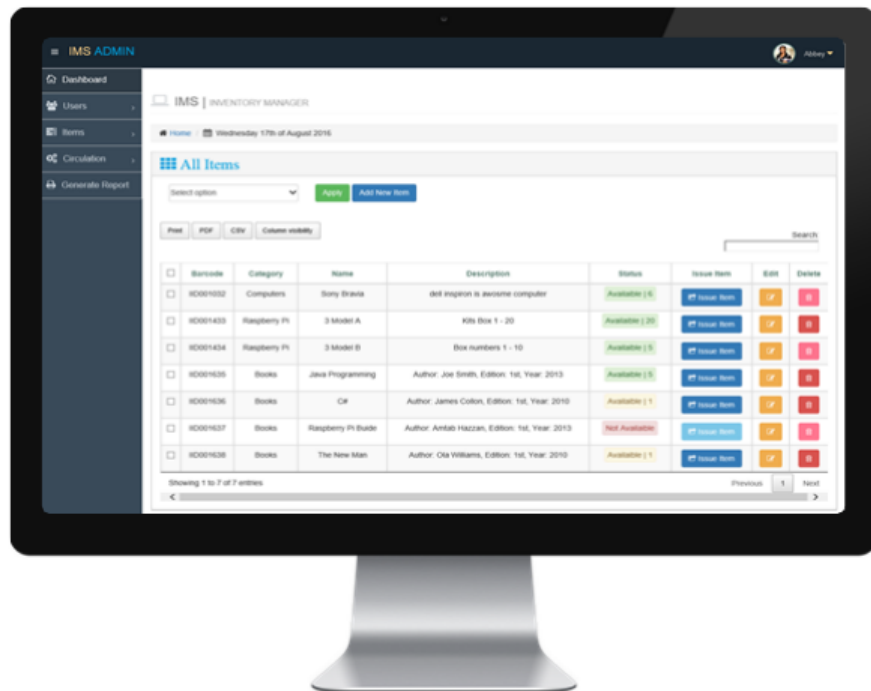


Figure 19: Manage Items

## Issue Item

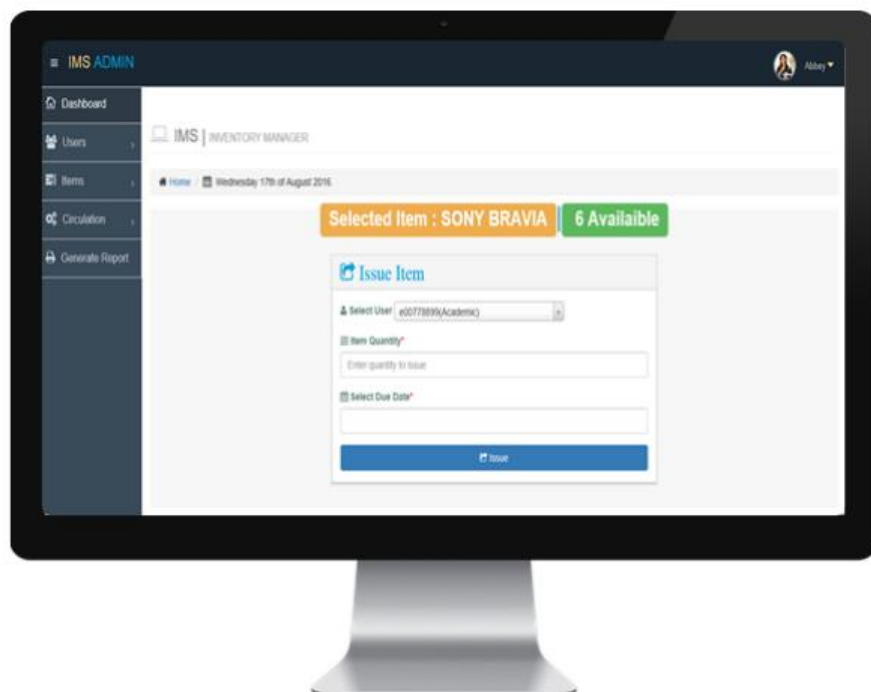


Figure 20: Issue Item

Item Category

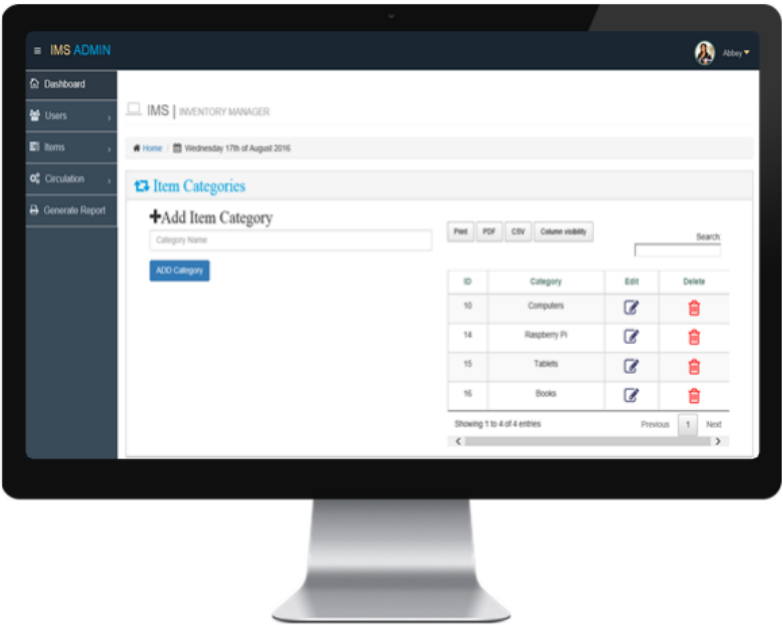


Figure 21: Item Category

Create Item Category	Delete Item Category	Print Item Category
Edit Item Category	Search Item Category	Export list (PDF, CSV)

Print Barcode

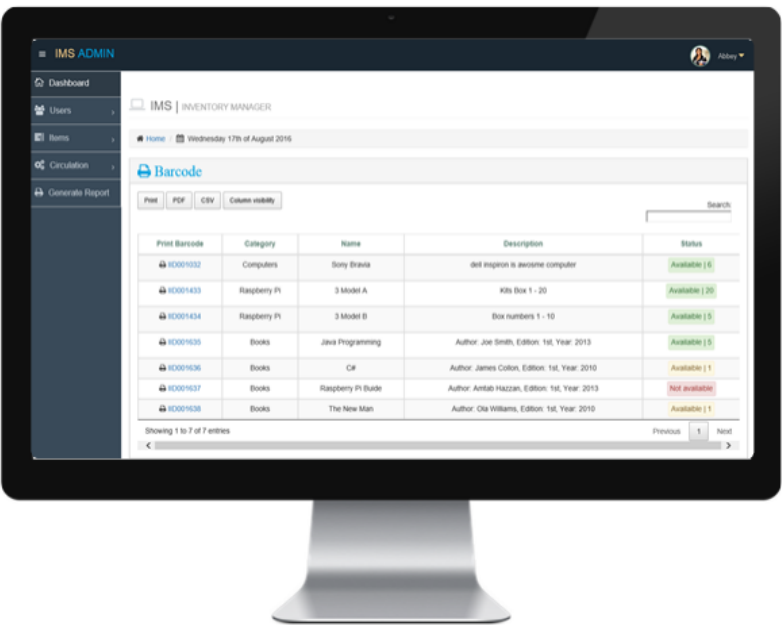


Figure 22: Print Barcode



## Circulation

➤ Issue History	➤ Return History
-----------------	------------------

## Issue History

Username	Barcode	Item Name	Quantity Issued	Issued Date	Due Date	Issue Status	Issued By	Return Item
80070089	IC001032	Sony Bravia	2	2016-08-11 09:04:06	2016-08-13 20:08:00	Confirmed ✓	subadmin	<a href="#">Return</a>
test234	IC001032	Sony Bravia	1	2016-08-10 12:04:14	2016-08-01 16:00:00	Confirmed ✓	subadmin	<a href="#">Return</a>
test075	IC001032	Sony Bravia	1	2016-08-10 11:58:27	2016-08-01 16:00:00	Confirmed ✓	subadmin	<a href="#">Return</a>
test492	IC001637	Raspberry Pi Bude	1	2016-08-10 01:36:28	2016-08-13 10:00:00	Confirmed ✓	subadmin	<a href="#">Return</a>
80012233	IC001032	Sony Bravia	56	2016-08-05 02:45:41	2016-08-19 17:00:00	Confirmed ✓	admin	<a href="#">Return</a>
40077899	IC001434	3 Model B	5	2016-07-29 11:37:47	2016-07-29 19:00:00	Confirmed ✓	subadmin	<a href="#">Return</a>
40077899	IC001637	Raspberry Pi Bude	1	2016-07-29 11:37:14	2016-08-05 18:00:00	Confirmed ✓	subadmin	<a href="#">Return</a>

Showing 1 to 7 of 7 entries

Figure 23: Issue History

## Return History

University ID	Barcode	Item Name	Quantity Returned	Issued Date	Due Date	Returned Date	Status
Maria	IC001433	3 Model A	1	2016-08-05 03:10:03	2016-08-12 15:00:00	2016-08-16 10:45:49	Confirmed ✓
80012233	IC001032	Sony Bravia	50	2016-08-03 02:13:27	2016-08-12 14:00:00	2016-08-03 12:41:13	Confirmed ✓
Maria	IC001032	Sony Bravia	10	2016-08-03 04:12:08	2016-08-06 03:18:00	2016-08-03 12:41:01	Confirmed ✓
80012233	IC001433	3 Model A	10	2016-07-30 02:06:01	2016-08-05 02:00:00	2016-08-03 01:52:47	Confirmed ✓
Maria	IC001032	Sony Bravia	50	2016-07-29 11:38:09	2016-07-31 22:38:00	2016-07-30 02:07:52	Confirmed ✓
80012233	IC001635	Java Programming	1	2016-07-30 01:52:08	2016-07-31 04:00:00	2016-07-30 02:05:41	Confirmed ✓
Sir_Abbey	IC001032	Sony Bravia	22	2016-07-26 10:27:43	2016-07-30 01:57:00	2016-07-26 10:28:14	Confirmed ✓

Showing 1 to 7 of 7 entries

Figure 24: Return History

Generate Report

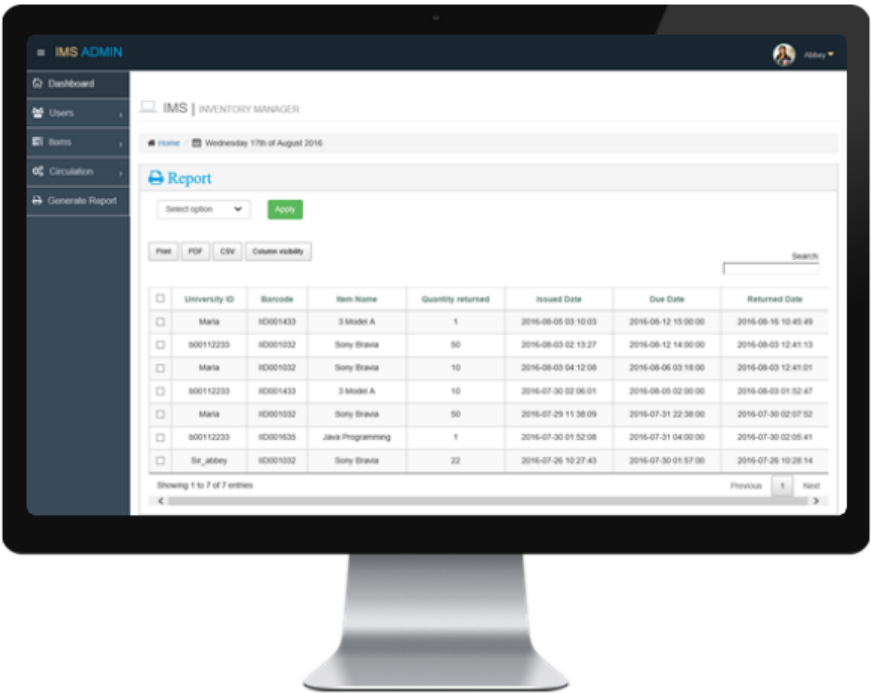


Figure 25: Generate Report

Member Profile

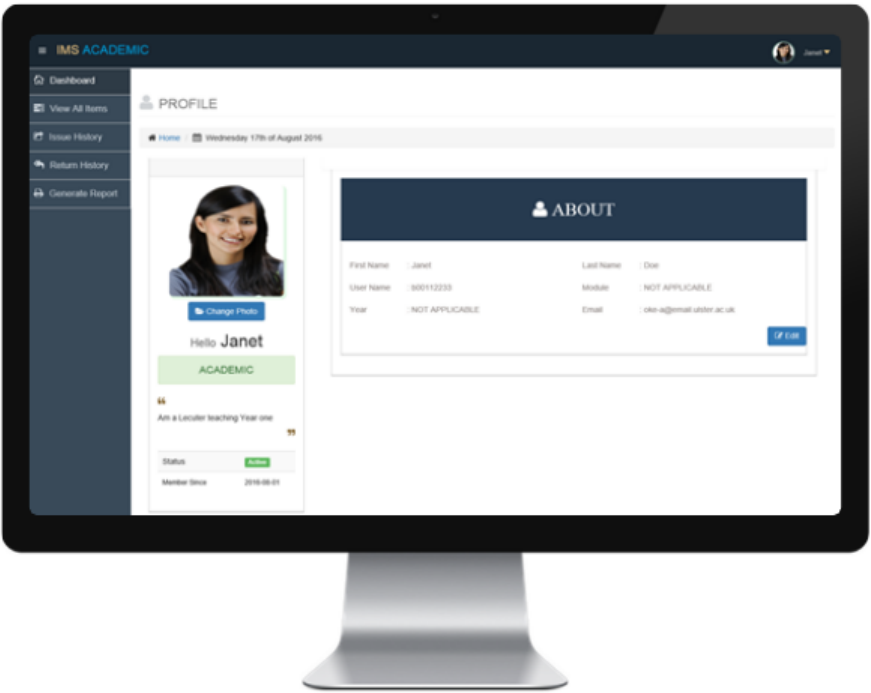


Figure 26: Profile

## Support multiple device screens

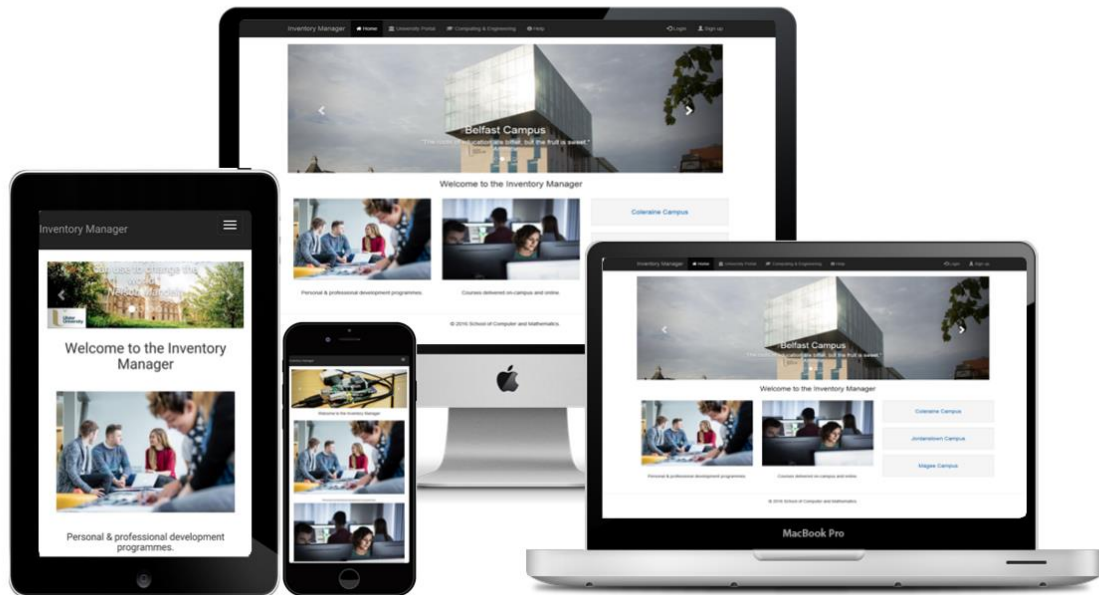


Figure 27: Support multiple device screen

# Test Cases

## IMS | Inventory Manager V 2.0

**Abiodun Oke [B00668298]**

**Supervisor: Dr Gaye Lightbody**

**Second Marker: Dr Pat Corr**



<b>Test Case ID</b>	IMS_001	<b>Test Case Description</b>	Test the Login functionality in IMS		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Username = admin
2	Password = password
3	

<b>Test Scenario</b>	Verify on entering valid username and password, the user can login
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to login link from homepage	Login page should open, credential can be entered and User is logged in.	User is logged in.	Pass
2	Provide valid username			Pass
3	Provide valid password			Pass
4	Click on login button			Pass

Table 5: Test Case ID: IMS\_001

<b>Test Case ID</b>	IMS_002	<b>Test Case Description</b>	Test the Login functionality error handling in IMS		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	Username = admin
2	Password = password01
3	Username = admin01
4	Password = password

<b>Test Scenario</b>	Verify unsuccessful login with invalid username/valid password and valid username/invalid password
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to login link from homepage	Login page should open, credential can be entered and user is not logged in.	User is presented with error messages. "user does not exist Sign up now" "invalid username or password"	Pass
2	Provide username			Pass
3	Provide password			Pass
4	Click on login button			Pass

Table 6: Test Case ID: IMS\_002

<b>Test Case ID</b>	IMS_003	<b>Test Case Description</b>	Test the Sign up functionality in IMS		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	As required on form
2	
3	
4	

<b>Test Scenario</b>	Verify successful sign up using valid input
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to Sign up link from homepage	Sign up page should open, Credential can be entered and User is signed up.	User is signed up and presented with confirmation message. "Your account will be active soon, details sent to Admin for approval. Thanks"	Pass
2	Provide required data			Pass
3	Click on Sign up button			Pass
4				

Table 7: Test Case ID: IMS\_003

<b>Test Case ID</b>	IMS_004	<b>Test Case Description</b>	Test the Sign up functionality error handling in IMS		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	As required on form
2	
3	
4	

<b>Test Scenario</b>	Verify successful sign up using invalid input
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to Sign up link from homepage	Sign up page should open, Credential can be entered and error message displayed in respect of field requirement	User unable to sign up and error messages displayed for each field as expected	Pass
2	Provide required data			Pass
3	Click on Sign up button			Pass
4				

Table 8: Test Case ID: IMS\_004



**Permission Test: ADMIN**

<b>Test Case ID</b>	IMS_005	<b>Test Case Description</b>	Test the Dashboard functionality in Admin User		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Username = admin
2	Password = password
3	

<b>Test Scenario</b>	Verify all information on ADMIN dashboard is accurate and up to date.
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to login link from homepage	Login page should open, Credential can be entered and User is logged in and directed to ADMIN dashboard.	User is logged in and all information on dashboard are accurate and up to date	Pass
2	Provide username			Pass
3	Provide password			Pass
4	Click on login button			Pass

Table 9: Test Case ID: IMS\_005

<b>Test Case ID</b>	IMS_006	<b>Test Case Description</b>	Test USERS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	None
2	
3	
4	

<b>Test Scenario</b>	Verify all sub-menu links in Users work as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to side navigation menu	Users menu should open sub-menu and "All Users" page should open	As expected	Pass
2	Click on users			Pass
3	Select Manage Users			Pass
4				Pass

Table 10: Test Case ID: IMS\_006

<b>Test Case ID</b>	IMS_007	<b>Test Case Description</b>	Test USERS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	None
2	
3	
4	

<b>Test Scenario</b>	Verify "Approve" functionality to Manage Users work as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to status column	Users status should change to "Approved"	As expected	Pass
2	Click on Approve button			Pass
3				
4				

Table 11: Test Case ID: IMS\_007

<b>Test Case ID</b>	IMS_008	<b>Test Case Description</b>	Test USERS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	As required on form
2	
3	
4	

<b>Test Scenario</b>	Verify "Edit" functionality to Manage Users work as expected
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Select user in row 1	Edit page should open pre-populated with user's current information. Once save button is clicked, new details should be updated.	As expected	Pass
2	Navigate to edit column			Pass
3	Click edit button			Pass
4	Change some data			Pass
5	Click save button			Pass

Table 12: Test Case ID: IMS\_008

<b>Test Case ID</b>	IMS_009	<b>Test Case Description</b>	Test USERS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	As required on form
2	
3	
4	

<b>Test Scenario</b>	Verify "Delete" functionality to Manage Users work as expected
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Select user in row 1	An alert box should appear to confirm delete action	As expected	Pass
2	Navigate to delete column			Pass
3	Click delete button			Pass
4				

Table 13: Test Case ID: IMS\_009

<b>Test Case ID</b>	IMS_010	<b>Test Case Description</b>	Test USERS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>FAIL</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	As required on form
2	Last Name = O'Neil
3	
4	

<b>Test Scenario</b>	Verify "ADD User / Admin" functionality to Manage Users work with special characters
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to ADD User	Add user page should open and new user details should be saved.	Error message displayed on output	Pass
2	Click the Add user button			Pass
3	Provide user details			Fail
4	Click save button			Fail

Table 14: Test Case ID: IMS\_010

<b>Test Case ID</b>	IMS_011	<b>Test Case Description</b>	Test USERS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Users.csv file
2	Populated with users details as required.
3	

<b>Test Scenario</b>	Verify import csv functionality to Add users work as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to the bottom of Add user page	File should upload and confirmation message displayed	As expected	Pass
2	Select user role			Pass
3	Browse csv file on device			Pass
4	Select csv file			Pass
5	Click the upload button			Pass

Table 15: Test Case ID: IMS\_011

<b>Test Case ID</b>	IMS_012	<b>Test Case Description</b>	Test USERS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Non-csv file (word.doc)
2	Populated with users details as required.
3	

<b>Test Scenario</b>	Verify import csv functionality to Add users does not work with non-csv files
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to the bottom of Add user page	File should not upload and error message displayed	As expected	Pass
2	Select user role			Pass
3	Browse non-csv file on device			Pass
4	Select non-csv file			Pass
5	Click the upload button			Pass

Table 16: Test Case ID: IMS\_012



<b>Test Case ID</b>	IMS_013	<b>Test Case Description</b>	Test ITEMS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	As required on form
2	
3	

<b>Test Scenario</b>	Verify "ADD item" functionality to Manage item works as expected
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to ADD item	Add item page should open and new item details should be saved.	As expected	Pass
2	Click the Add item button			Pass
3	Select item category			Pass
4	Provide item details			Pass
5	Click save button			Pass

Table 17: Test Case ID: IMS\_013

<b>Test Case ID</b>	IMS_014	<b>Test Case Description</b>	Test ITEMS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	Items.csv file
2	Populated with items details as required.
3	
4	

<b>Test Scenario</b>	Verify import csv functionality to Add items work as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to the bottom of Add item page	File should upload and confirmation message displayed	As expected	Pass
2	Select item category			Pass
3	Browse csv file on device			Pass
4	Select csv file			Pass
5	Click the upload button			Pass

Table 18: Test Case ID: IMS\_014

<b>Test Case ID</b>	IMS_015	<b>Test Case Description</b>	Test ITEMS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	As required on form
2	
3	

<b>Test Scenario</b>	Verify "Issue Item" functionality to issue item work as expected
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Select item in row 1	Issue item page should open and item issued to user with a confirmation message display	As expected	Pass
2	Click issue item button			Pass
3	Select user			Pass
4	Provide quantity and due date			Pass
5	Click the issue button			Pass

Table 19: Test Case ID: IMS\_015

<b>Test Case ID</b>	IMS_016	<b>Test Case Description</b>	Test ITEMS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	As required on form
2	
3	

<b>Test Scenario</b>	Verify "Edit" functionality to Manage Items work as expected
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Select item in row 1	Edit page should open pre-populated with item's current data. Once update button is clicked, new details should be updated.	As expected	Pass
2	Navigate to edit column			Pass
3	Click edit button			Pass
4	Change some data			Pass
5	Click update button			Pass

Table 20: Test Case ID: IMS\_016

<b>Test Case ID</b>	IMS_017	<b>Test Case Description</b>	Test ITEMS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	None
2	
3	
4	

<b>Test Scenario</b>	Verify "Delete" functionality to Manage Item's work as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Select item in row 1	An alert box should appear to confirm delete action	As expected	Pass
2	Navigate to delete column			Pass
3	Click delete button			Pass
4				

Table 21: Test Case ID: IMS\_017

<b>Test Case ID</b>	IMS_018	<b>Test Case Description</b>	Test ITEMS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	None
2	
3	
4	

<b>Test Scenario</b>	Verify "Delete" functionality to Manage Item's does not work if item is on loan
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Select an item on loan	Delete button should be disabled	As expected	Pass
2	Navigate to delete column			Pass
3	Click delete button			Pass
4				

Table 22: Test Case ID: IMS\_018

<b>Test Case ID</b>	IMS_019	<b>Test Case Description</b>	Test CIRCULATION functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	None
2	
3	
4	

<b>Test Scenario</b>	Verify "Return" functionality to Manage circulation works as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to circulation	Issued item history page should open and item returned once return button is clicked	As expected	Pass
2	Click issue history			Pass
3	Navigate to return item column			Pass
4	Click return item button row 1			Pass

Table 23: Test Case ID: IMS\_019

<b>Test Case ID</b>	IMS_020	<b>Test Case Description</b>	Test CIRCULATION functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	None
2	
3	
4	

<b>Test Scenario</b>	Verify return information is accurate and up to date.
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to circulation	Return item history page should open and list of all item history should be available and accurate	As expected	Pass
2	Click return history			Pass
3				
4				

Table 24: Test Case ID: IMS\_020



<b>Test Case ID</b>	IMS_021	<b>Test Case Description</b>	Test GENERATE REPORT functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	None
2	
3	

<b>Test Scenario</b>	Verify report functionality works as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to Generate Report	Report page should open and each button should generate required file once clicked. Visibility button should hide column once clicked.	As expected	Pass
2	Click Print button			Pass
3	Click PDF button			Pass
4	Click CSV button			Pass
5	Click column visibility button			Pass

Table 25: Test Case ID: IMS\_021

<b>Test Case ID</b>	IMS_022	<b>Test Case Description</b>	Test EDIT PROFILE functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	As required on form
2	
3	

<b>Test Scenario</b>	Verify edit profile functionality works as expected
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to user profile	Edit Profile page should open and fields pre-populated with current information. New data should save on correctly.	As expected	Pass
2	Click My Profile			Pass
3	Click Edit button			Pass
4	Provide data			Pass
5	Click save button			Pass

Table 26: Test Case ID: IMS\_022

<b>Test Case ID</b>	IMS_023	<b>Test Case Description</b>	Test CHANGE PASSWORD functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Old Password = password
2	New password = password01
3	

<b>Test Scenario</b>	Verify change password functionality works as expected
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to user profile	Change password page should open and password changed once correct data provided.	As expected	Pass
2	Click Change password			Pass
3	Provide Old password			Pass
4	Provide new PW & confirm			Pass
5	Click change password button			Pass

Table 27: Test Case ID: IMS\_023

<b>Test Case ID</b>	IMS_024	<b>Test Case Description</b>	Test CHANGE PASSWORD functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Password = pass
2	New password = password
3	

<b>Test Scenario</b>	Verify password does not change if incorrect old password is provided
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to user profile	Change password page should open and error message display once change password button clicked	As expected	Pass
2	Click Change password			Pass
3	Provide incorrect Old password			Pass
4	Provide new password			Pass
5	Confirm new password			Pass
6	Click change password button			Pass

Table 28: Test Case ID: IMS\_024

<b>Test Case ID</b>	IMS_025	<b>Test Case Description</b>	Test LOG OUT functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	None
2	
3	
4	

<b>Test Scenario</b>	Verify that user is able to logout from the system
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to user profile	User should be logged out and redirected to the homepage	As expected	Pass
2	Click Log out			Pass
3				
4				

Table 29: Test Case ID: IMS\_025

<b>Test Case ID</b>	IMS_026	<b>Test Case Description</b>	Test DELETE functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	None
2	
3	

<b>Test Scenario</b>	Verify multiple delete functionality works as expected
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to circulation	Return item history page should open and item history selected deleted once apply button is clicked	As expected	Pass
2	Click return history			Pass
3	Select checkbox row 1-3			Pass
4	Select "delete selected"			Pass
5	Click Apply button			Pass

Table 30: Test Case ID: IMS\_026

<b>Test Case ID</b>	IMS_027	<b>Test Case Description</b>	Test CHANGE PASSWORD functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Password = password
2	New password = password01
3	Confirm new password = pass01

<b>Test Scenario</b>	Verify password does not change if new password mismatch is provided
----------------------	--

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to user profile	Change password page should open and error message display once change password button clicked	As expected	Pass
2	Click Change password			Pass
3	Provide correct Old password			Pass
4	Provide new password			Pass
5	Provide an incorrect new PW			Pass
6	Click change password button			Pass

Table 31: Test Case ID: IMS\_027

<b>Test Case ID</b>	IMS_028	<b>Test Case Description</b>	Test ITEMS functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser

S#	Test Data
1	Items.csv file
2	Populated with items details as required.
3	

<b>Test Scenario</b>	Verify import csv functionality to Add items does not work if item already exists
----------------------	---

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to the bottom of Add item page	File should not upload and error message should be displayed with a <b>“failed upload: reason, duplicate item in csv file”</b>	As expected	Pass
2	Select item category			Pass
3	Browse csv file on device			Pass
4	Select csv file			Pass
5	Click the upload button			Pass

Table 32: Test Case ID: IMS\_028



<b>Test Case ID</b>	IMS_029	<b>Test Case Description</b>	Test USER PERMISSION functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	PASS
----------------------	-----	--------------------	----------------	---	------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	Username: academics
2	Password: password
3	
4	

<b>Test Scenario</b>	Verify Academic user is able to login
----------------------	---------------------------------------

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to login link from homepage	Login page should open, Credential can be entered and User is logged in as Academic.	User is logged in as Academic.	Pass
2	Provide valid username			Pass
3	Provide valid password			Pass
4	Click on login button			Pass

Table 33: Test Case ID: IMS\_029

<b>Test Case ID</b>	IMS_030	<b>Test Case Description</b>	Test USER PERMISSION functionality		
<b>Created By</b>	Abi	<b>Test Priority (Low/Medium/High)</b>	High	<b>Version</b>	2.0

<b>Tester's Name</b>	Abi	<b>Date Tested</b>	15-August-2016	<b>Test Case (Pass/Fail/Not Executed)</b>	<b>PASS</b>
----------------------	-----	--------------------	----------------	---	-------------

S#	Prerequisites
1	Access to Chrome Browser
2	Access to IE Browser
3	Access to Firefox Browser
4	

S#	Test Data
1	Username: student
2	Password: password
3	
4	

<b>Test Scenario</b>	Verify Student user is able to login
----------------------	--------------------------------------

Step#	Step Details	Expected Results	Actual Results	Pass/ Fail /Not executed / Suspended
1	Navigate to login link from homepage	Login page should open, Credential can be entered and User is logged in as Student.	User is logged in as Student.	Pass
2	Provide valid username			Pass
3	Provide valid password			Pass
4	Click on login button			Pass

Table 34: Test Case ID: IMS\_030

## Functional Testing (Test Results)

Status	Total	%
PASS	29	97
FAIL	1	3
NOT EXECUTED	0	0
SUSPENDED	0	0
TOTAL TEST CASES	30	100

Table 35: Functional Testing Results

## NON-FUNCTIONAL TESTING

Testing Criteria	Desktop	Laptop	Tablet	Smart Phone
Compatibility Testing (Hardware)	PASS	PASS	PASS	PASS

Table 36: Compatibility Testing (Hardware)

Testing Criteria	IE	Google Chrome	Firefox	Safari
Compatibility Testing (Browser)	PASS	PASS	PASS	PASS

Table 37: Compatibility Testing (Browser)

Testing Criteria	Windows 10	Windows Vista	MAC	iOS
Compatibility Testing (Operating Systems)	PASS	PASS	PASS	PASS

Table 38: Compatibility Testing (Operating Systems)