

B₃L

Project Design

Project Name: Information Literacy Sessions Form Redesign

Project Management Group: B3L

Prepared By:

Team Members
Ben Barclay
Ben Cornish
Ben Williams
Lance Baker

Table of Contents

1 Software Requirements specification	3
1.1 Introduction	3
1.1.1 Purpose	3
1.1.2 Intended Audience and Reading Suggestions	3
1.2 Project Scope.....	3
1.2.1 References	3
1.3 Overall Description	4
1.3.1 Product Perspective	4
1.3.2 Product Features	4
1.3.3 User Classes and Characteristics	4
1.3.4 Operating Environment	5
1.3.5 Design and Implementation Constraints	5
1.3.6 User Documentation	5
1.4 System Features	6
1.4.1 User Authentication	6
1.4.2 Dynamically Changeable Field Values	7
1.4.3 Submit Statistics	8
1.4.4 View Statistics	9
1.4.5 Dynamic Form Design.....	10
1.5 External Interface Requirements	11
1.5.1 User Interfaces	11
1.5.2 Software Interfaces	11
1.5.3 Communication Interfaces.....	12
1.6 Other Non-functional Requirements	12
1.6.1 Performance Requirements.....	12
1.6.2 Security Requirements.....	12
1.6.3 Software Quality Attributes	12

2 Use Case Diagram	13
3 Use Case Descriptions	14
3.1 Create Report	14
3.2 System Administration	15
3.3 Editing Form	16
3.4 View Reports	17
3.5 New Form Entry	18
3.6 Validate user access	19
4 Class Diagram.....	20
5 Entity Relationship Diagram	21
6 Relational Database Design	22
7 Screen Design	23
7.1 Introduction	23
7.2 Login page	23
7.3 Navigation Menu	24
7.4 Forms page	24
7.5 Form submitted page	26
7.6 Reporting page.....	27
7.7 Administration menu	28
7.8 Administration – Edit user	29
7.9 Administration – Edit form	30
8 User Testing Procedures 31	
8.1 Testing Approach.....	31
8.2 Testing Activities	32
8.2.1 Create Report.....	32
8.2.2 System Administration	33
8.2.3 Editing Form.....	34
8.2.4 View Reports	34
8.2.5 New Form Entry	35
8.2.6 Validate User Access	35

1 Software Requirements specification

1.1 Introduction

1.1.1 Purpose

The purpose of this document is to further emphasize the requirements of the information literacy sessions system redesign. The new features in the project will be further detailed, and the overall feel of the new system will be illustrated. The document will cover every aspect of the new system, and the system will be built according to the project scope mentioned throughout this document.

1.1.2 Intended Audience and Reading Suggestions

The document will be intended for people with some technical knowledge; however, there will also be some people with no technical knowledge, so therefore the document will be written to keep this in mind. The readers with some technical knowledge should start from the beginning of the document, and proceed through every section. For people whom may not have technical knowledge; it might be preferable to start from the project scope, secondly the systems features, and then go through the remaining document.

1.2 Project Scope

1. The new system will implement security measures using the existing authentication system used by the University. The staff accounts can be setup based on the security groups assigned in active directory, which will therefore allow consistent usage of staff user accounts; also providing them with the ability to use the existing campus support for ongoing ease of user administration.
2. The new system will encompass the functionality to customise the form field values for each campus location. It would have the authentication implemented in such a way, that it would detect the campus location of the user account, and only show the field values applicable to their campus location. Therefore, this method would dramatically reduce the selectable fields - making the form easier to navigate, with an increase of relevancy to their campus.
3. The form layout will comprise the ability to dynamically change based on the session type selected, therefore making the form more relevant by only displaying the appropriate form fields. The form will also be redesigned to increase the usability in other areas; such as implementing a date picker (instead of individual fields), providing ways to submit multiple sessions, whilst also improving the form navigation.

1.2.1 References

- The document refers to the existing system in some instances, which can be found on the Newcastle University website using the following web address:
http://www.altarama.com.au/ASAPinfo/ref100.aspx?key=Information_Literacy

1.3 Overall Description

1.3.1 Product Perspective

The system being developed is a replacement for the current library class statistics system. This system is a subsystem of the larger university statistics monitoring software. As such this will integrate into that larger system. In addition it will provide statistics to the ASAP library statistic system. These statistics are used to determine further funding and class support for the individual libraries that use the system.

1.3.2 Product Features

This system is focused on four key features that will differentiate it from the system it will replace. These features are: easy data entry, security, usability of statistics and modification by the end users.

- The current system has a very inefficient data entry form, with lots of duplication of data and irrelevant fields. We intend to address this by offering a form customized for each library.
- The current system has no authentication or verification of users or data. Our system will address this security issue by integrating with the University LDAP authentication system. This will allow for roles to be specified and users assigned permissions to only the resources they should be able to access.
- The current system generates statistics based on the data that the library staff input, but there is no way for them to access these statistics due to system limitations. In addition the statistics can only be accessed by one person, who is currently away on extended leave, leaving everyone unable to access the data. We plan to address this shortcoming by showing user and library statistics on a separate page available to logged-in users.
- Finally the current system is extremely hard to modify and when modifications are made there are often mistakes that remain on the page for months. We propose a system that allows modification of classes and other items through an administration page available to a subset of all users.

1.3.3 User Classes and Characteristics

We propose two levels of users: standard users and administrative users. Standard users will be able to enter data into the system and be able to see their personal statistics. Administrative users will have all of the features of standard users and in addition they will be able to see library wide statistics and be able to access the administrative page to manipulate the data being presented on the form.

1.3.4 Operating Environment

The system will hopefully be hosted on University servers; however, it is uncertain at this stage whether their infrastructure can support a PHP5/ MySQL environment. The system will be set up using Apache2 on either a Windows or Linux operating system. If the server is using IIS, then the Apache installation would need to be configured on a different http port than the default port 80. It is almost certain that the University would be able to support another httpd environment on top of their existing infrastructure, so therefore the essential hosting for the system should not be a problem. Clients will access the system through a web browser. The system will be platform independent for end users.

1.3.5 Design and Implementation Constraints

At this stage in the project we have identified several constraints:

- The university hosting facilities.
- The statistic requirements of various parties.
- The time required to complete the project.
- The course requirements.
- Final acceptance from Newcastle officials.

1.3.6 User Documentation

We plan to provide several forms of user documentation to assist with the user acceptance of the system. These documents will primarily consist of the following:

- Tutorial – a brief overview of the system capabilities and how to use them.
- Manual – a detailed overview of system capabilities and how to use them.
- Help file – a look up reference to find out how to use specific features.

1.4 System Features

The system features section will continue on from the proposal document.

The following is an outline of the system features that will be incorporated into the new system.

1.4.1 User Authentication

Description and Priority

The user authentication is classified as a High priority feature; since system security is essential, as without security measures implemented, the system will be prone to misuse and inaccurate statistics - leading to the system being unreliable for the determination of facility usage. The user authentication however should be ideally using the University authentication services, therefore providing the ability for staff to use their existing logon credentials.

The feature however, may not be able to use the University authentication services in this release (which will be determined after consultation with a technical representative); so therefore the system may have to be designed with its own login system implemented, which could then be replaced at a later date. The only drawback in doing it this way would be that there may be some back-tracking in development; but since security is essentially the most important feature associated with the system - it cannot go without being implemented in the release.

Stimulus/ Response Sequences

1. The user is prompted with an authentication page to input their credentials.
2. The user enters their credentials and submits the form.
3. The data entered is cross-referenced with the credentials stored in the system.
 - a. The user is logged-in (allowed entry); with access to their local campus form display – with the ability to see an administration panel (if they have permission).
 - b. The user account does not logged-in and is displayed an error message indicating the failure.

Functional Requirements

The system should hopefully interface with the existing University infrastructure, and enable the ability for the users to use their existing user accounts. The authentication system should have permission grouping on the university's end; enabling them to add additional users to the security groups, thus allowing them to also use the existing support services offered on campus once the system is rolled out.

- The user accounts will be setup based on *locality*; allowing for the system to know which campus they are a member of dynamically once logged-in. This could theoretically 'look-up' their campus relating details by querying the existing user account information stored in Active Directory, by either selecting for users based on the information stored within a field – or finding the campus group the user belongs to. If there is no data stored representing the location of the user, then we could (if permitted) setup the particular user accounts in permission groups relating to the campuses.
- The user accounts will have to be filtered, since their will only be a select few whom will have access to the service; therefore, the accounts will have to belong to a security group relating to system usage. The same will also apply for *user access types*, since we will not want all users who have access to the system have administration access - so we may need two security groups setup to meet this requirement.

1.4.2 Dynamically Changeable Field Values

Description and Priority

The system should provide the ability to change the field values relating to the campus location of the user account. The system will need to have a facility in the administration panel to *customize* the fields to their own specifications based on the campus location. Therefore, the campus can have their own specific fields applying directly to their campus location (training rooms, and others) which will only be used to that campus.

Stimulus/ Response Sequences

1. The user selects 'Change Fields' option available in the administration panel.
2. The system lists the available fields (with the option to enable and disable).
3. The user selects a field.
4. The system provides the field values relating to the field selected.
5. The user has the option to add, or remove a field value.

Functional Requirements

- The system should enable for the campuses to customize the form field values relating to their needs. The actual fields themselves will be static in nature; however the campus can disable fields that don't exactly correspond with their needs. The static fields (containing the drop downs) would have the option to be customized – which enables the administration staff to specify the field values.
- The user once logged-in will have their campus location automatically determined, and their user account information will be stored for that session. The form once loaded, will only show the fields enabled for that campus location (as specified in the administration panel) and the field values will be set only for the select dropdowns.

1.4.3 Submit Statistics

Description and Priority

The statistics will be stored in the system database; however, since the statistics will be used by multiple campuses - it will have to be designed to accommodate the issue of having the fields' campus changeable. The foreseeable solution would be to have all the form fields within the database table, and once the record is submitted - for any fields that aren't shown for that campus, it will just submit a null value. The statistics table will be designed to link up with the field values table, therefore eliminating data redundancy issues.

Stimulus/ Response Sequences

1. The user logins with their user credentials.
2. The system loads the form showing the fields applicable to that campus, with the select dropdowns populated with the field values setup.
3. The user fills out the form, and submits it.
4. The system stores the submitted data in the statistics table.

Functional Requirements

- The submitted statistic data should be everything visible on the form; which would need to correctly link to the field values stored separately. The table layout should be designed keeping data redundancy concerns in mind, therefore allowing ease of updating field values; in case they decide to change the name, or value of something to something else later on.
- The statistics will be queried, and on some instances by multiple campuses; therefore the data should be stored in such a way, which will facilitate this need.

1.4.4 View Statistics

Description and Priority

The viewing statistics feature will be incorporated in the administration panel, and will essentially consist of a couple ways to view the statistics data:

- The data will be displayed in a grid layout form, which will basically list the records submitted corresponding to that campus location; with an option to delete records (which may be ones that weren't submitted correctly). This allows the Librarians to submit statistics, review the statistic records, and also remove any statistic records that weren't entered correctly. The viewing facility will not however provide a functionality to edit the records, since if they are correctly entered, they should stay in that state – providing the option to also remove the record in-case it was accidentally submitted incorrectly.
- The data will be also displayed in the form of a line graph. It should ideally be a graph showing room usage based on date ranges; however, the fields which the graph is based on should have options to be filtered. Therefore, there will be a graph capable of displaying more than one type of data.

Stimulus/ Response Sequences

1. The administrative user is logged in.
2. The user selects the ‘Statistics’ option located in the administration panel, which dynamically expands revealing the ‘View Graph’, and ‘View Data’.
 - a. The user selects ‘View Data’.
 - a.i. The view data page is loaded by the system; the data is selected from the database, and outputted in a tabled form by the system.
 - a.ii. The user selects ‘Delete’ next to a record. The view data page is then reloaded with the delete request and the system then removes the statistics record from the database.
 - b. The user selects ‘View Graph’
 - b.i. The graph page is loaded with the default graphed form selected (the room usage).
 - b.ii. The database will be queried, and the system will generate a graph based on that data received.

Functional Requirements

- The system will need to have the functionality to enable users to view statistics; which will be a section within the administration panel.
- The statistics viewing facility will have two viewing displays; the first display will enable the statistic data to be listed in a record table layout – with the other as a graphed form.
- The graph should provide the facility to be filtered; so it displays more than one set of fields, therefore allowing the same graph to display different sets of statistics.

1.4.5 Dynamic Form Design

Description and Priority

- The new system will be designed to make the form more ‘dynamic’ and improve usability by incorporating dynamic aspects – such as an interactive date picker.
- The new system will have the capability to submit multiple statistic records (with relative ease) which will be done by submitting the form using an asynchronous JavaScript request – and once the form has been submitted correctly, it will be cleared, and ready for immediate input.
- The form layout will be determined based on the session type selected; since there are some differences in field usage regarding a Library Tour (for example) in comparison to a normal training session.

1.5 External Interface Requirements

1.5.1 User Interfaces

We plan to have a series of user interface elements that will remain consistent throughout the application. They will be implemented as a separate entity that will appear on each of the individual pages of the application.

Elements common to all pages:

- Header
- Footer
- Menu

Individual Pages:

- User Logon/ Authentication
- Statistics Form
- View Statistics
 - View Data
 - View Graph
- Administration screen
 - Manage Form Fields
 - Manage Users
 - Manage Reports

1.5.2 Software Interfaces

The system will require specific software on the server so that it can interface with its various components as well as having the ability to interface with the required statistic systems that are required.

Software requirements for implementation:

- Apache2
- PHP5
- MySQL
- LDAP

Software requirements for output:

- Generate statistic records (from the table) as CSV; which will be used to submit to ASAP.

1.5.3 Communication Interfaces

The application will be delivered over a university wide intranet. As such it will require standard web browser compliance. Internally the application will require LDAP to communicate with and utilize the university's current authentication system. In addition the project will utilize SQL to perform database transactions. How the output is to be used will be determined by the end parties that require the data.

1.6 Other Non-functional Requirements

1.6.1 Performance Requirements

The system ideally should be used by multiple campuses, and therefore would have concurrent connections. The performance requirements of the hosting should be substantial enough to withstand the required system load to fulfill that requirement. Also, the existing university infrastructure is using Microsoft based services (such as ASP.NET) – therefore since we would preferably want to use PHP/MySQL development environment, it may be best to use a dedicated machine. However, this will need to be further discussed with the technical representative. The actual hardware requirements required by the system would be minimal – being enough to support at least twenty concurrent connections, which can be easily accomplished with an older retired machine with a LAMP environment.

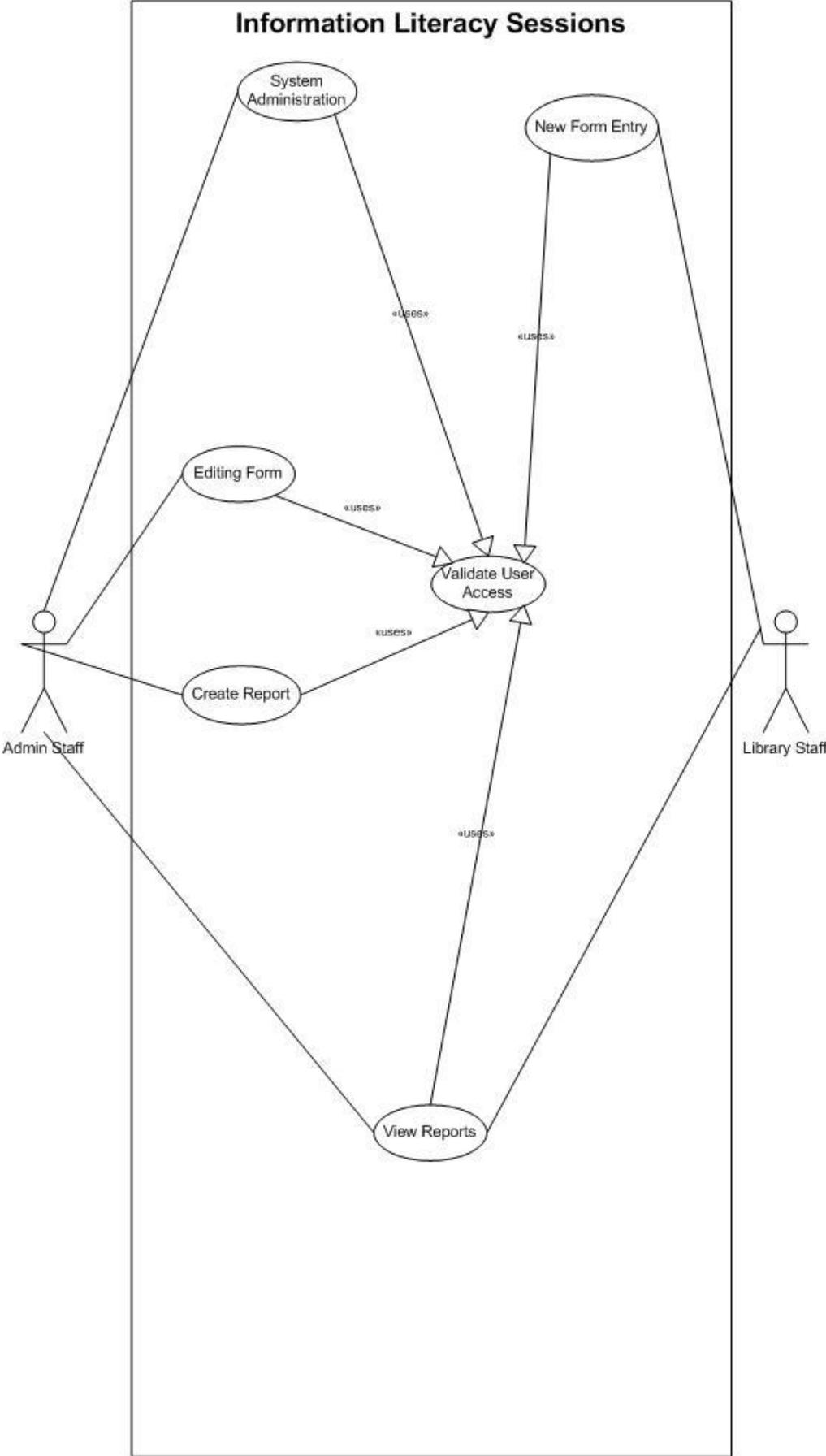
1.6.2 Security Requirements

The system should provide authentication capabilities (as outlined before), and should also operate on a secured server environment. The communication should also ideally be operating on an encrypted tunnel (such as a secured socket layered connection); which should be recommended in the final release version, however, since of drawbacks with the inability to talk to technical representatives from the university - it will not be included within the scope.

1.6.3 Software Quality Attributes

The system will be designed to keep standardized coding conventions in mind; which with PHP scripting (will be the Zend coding conventions) and relating to the user interface aspects (HTML/CSS) it will follow the standards advised by the World Wide Web Consortium (W3C).

2 Use Case Diagram



3 Use Case Descriptions

3.1 Create Report

Use Case name:	Create Report	
Scenario:	Admin staff create a new report according to specification	
Triggered Event:	A new report is required to be generated	
Brief Description:	When a custom report is required, admin staff may create this report referencing certain fields in the database. Relevant users can then access this report.	
Actors:	Admin Staff	
Related Use Cases:	Validate user access	
Stakeholders:	Admin Staff, Library staff	
Preconditions:	User must have required access	
Postconditions:	Report is created	
Flow of events:	Actor	System
	<ol style="list-style-type: none"> 1. New report is requested 2. Admin staff log on to system 3. Admin staff initiates report creation 4. Admin staff selects relevant fields 5. Admin staff formats report 6. Admin staff saves report 	<ol style="list-style-type: none"> 2.1 Verify user access 3.1 Create new report 3.2 Display available database fields 6.1 Validate report data 6.2 Saves report changes
Exception conditions:	<ol style="list-style-type: none"> 2.1 If user does not have the correct access the report creation will not be available 6.1 If report is not valid, user will be prompted to resolve issue(s) 	

3.2 System Administration

Use Case name:	System Administration	
Scenario:	Admin staff edit user access rights	
Triggered Event:	A users access rights need to be edited	
Brief Description:	A users access rights may need to be edited if they are a new user, have changed job positions or left the organisation	
Actors:	Admin Staff	
Related Use Cases:	Validate user access	
Stakeholders:	Admin Staff, Library staff	
Preconditions:	Admin staff have sufficient privileges to edit user access	
Postconditions:	User account has been modified, created or deleted	
Flow of events:	Actor	System
	<ol style="list-style-type: none"> 1. User requests modification or creation of access 2. Access rights are approved 3. Admin staff log on to system 4. Admin staff initiates system administration console 5. Admin staff edit user access rights as required 6. Admin staff save changes 	<ol style="list-style-type: none"> 3.1 Verify user access 4.1 Retrieve current user access 6.1 Validate user rights 6.2 Saves changes
Exception conditions:	<ol style="list-style-type: none"> 3.1 If user does not have the correct access the system administration console will not be available 6.1 If user rights are not valid, user will be prompted to resolve issue(s) 	

3.3 Editing Form

Use Case name:	Editing Form	
Scenario:	Admin staff update field values in form	
Triggered Event:	New values need to be added to the form	
Brief Description:	Over time, field values will become out of date and will need to be maintained	
Actors:	Admin Staff	
Related Use Cases:	Validate user access	
Stakeholders:	Admin Staff, Library staff	
Preconditions:	User must have required access	
Postconditions:	Form is successfully modified	
Flow of events:	Actor	System
	<ul style="list-style-type: none"> • Request submitted for form modification • Modifications are approved • Admin staff log on to system • Admin staff initiates form modification console • Admin staff edit form as required • Admin staff save changes 	3.1 Verify user access 4.1 Retrieve current form details 6.1 Validate form data 6.2 Saves form changes
Exception conditions:	2.1 If user does not have the correct access the form modification console will not be available 6.1 If form is not valid, user will be prompted to resolve issue(s)	

3.4 View Reports

Use Case name:	View Reports	
Scenario:	Admin or Library staff need to view a report	
Triggered Event:	A report needs to be viewed	
Brief Description:	Admin staff and Library staff can view a selection of pre-designed reports	
Actors:	Admin Staff, Library Staff	
Related Use Cases:	Validate user access	
Stakeholders:	Admin Staff, Library staff, system stakeholders	
Preconditions:	User must have required access	
Postconditions:	Report is successfully produced.	
Flow of events:	Actor	System
	<ol style="list-style-type: none"> 1. Request for a report is made 2. User logs on to system 3. User opens reporting console 4. User selects required report 5. User views the report 6. User exports report (if required) 	<ol style="list-style-type: none"> 2.1 Verify user access 3.1 Verify users report access 3.2 Retrieve available reports 4.1 Generate report 6.1 Generate exported report
Exception conditions:	<ol style="list-style-type: none"> 2.1 If user does not have the correct access the reporting console will not be available 3.1 Library staff may not have access to available reports 	

3.5 New Form Entry

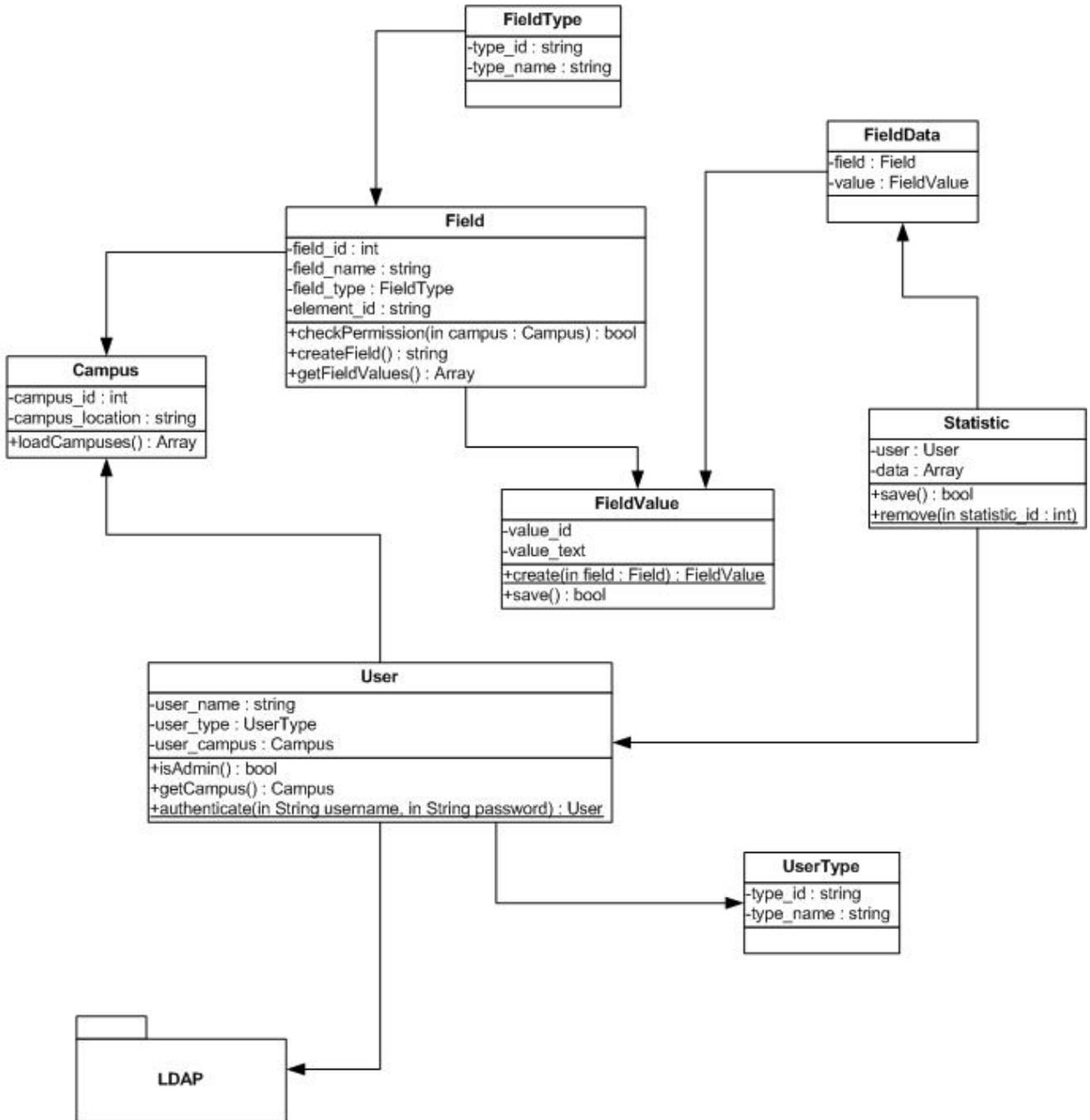
Use Case name:	New form entry	
Scenario:	Library Staff enter new form	
Triggered Event:	Information literacy session has been conducted and needs to be recorded	
Brief Description:	Information literacy session has been conducted Details are entered into the form and submitted	
Actors:	Library Staff	
Related Use Cases:	Validate user access	
Stakeholders:	Admin Staff, Library staff	
Preconditions:	User must have required access Information literacy session has been conducted	
Postconditions:	Form is successfully submitted	
Flow of events:	Actor	System
	<ol style="list-style-type: none"> 1. Information literacy session has been conducted 2. Library staff log on to system 3. Library staff opens form entry console 4. Library staff enters details of Information literacy session 5. Library staff submit form 	<ol style="list-style-type: none"> 2.1 Verify user access 3.1 Display form 4.1 Validate data input 4.2 Display relevant field values according to inputted data 6.1 Validate data input 6.2 Save data to database 6.3 Display confirmation
Exception conditions:	<ol style="list-style-type: none"> 2.1 If user does not have the correct access the Information literacy session console will not be available 4.1 If field value is not valid, user will be prompted to resolve issue(s) 6.1 If field value is not valid, user will be prompted to resolve issue(s) 	

3.6 Validate user access

Use Case name:	Validate user access	
Scenario:	User access needs to be validated	
Triggered Event:	User logs into the system	
Brief Description:	Admin staff and Library staff can view a selection of pre-designed reports	
Actors:	N/A	
Related Use Cases:	Editing form New form entry Create report System administration View Reports	
Stakeholders:	Admin Staff, Library staff	
Preconditions:	User must try to access system	
Postconditions:	User is either granted or denied access to the system	
Flow of events:	Actor	System
	1. Use case requests access to restricted system component	1.1 Grants or denies access according to access control list
Exception conditions:		

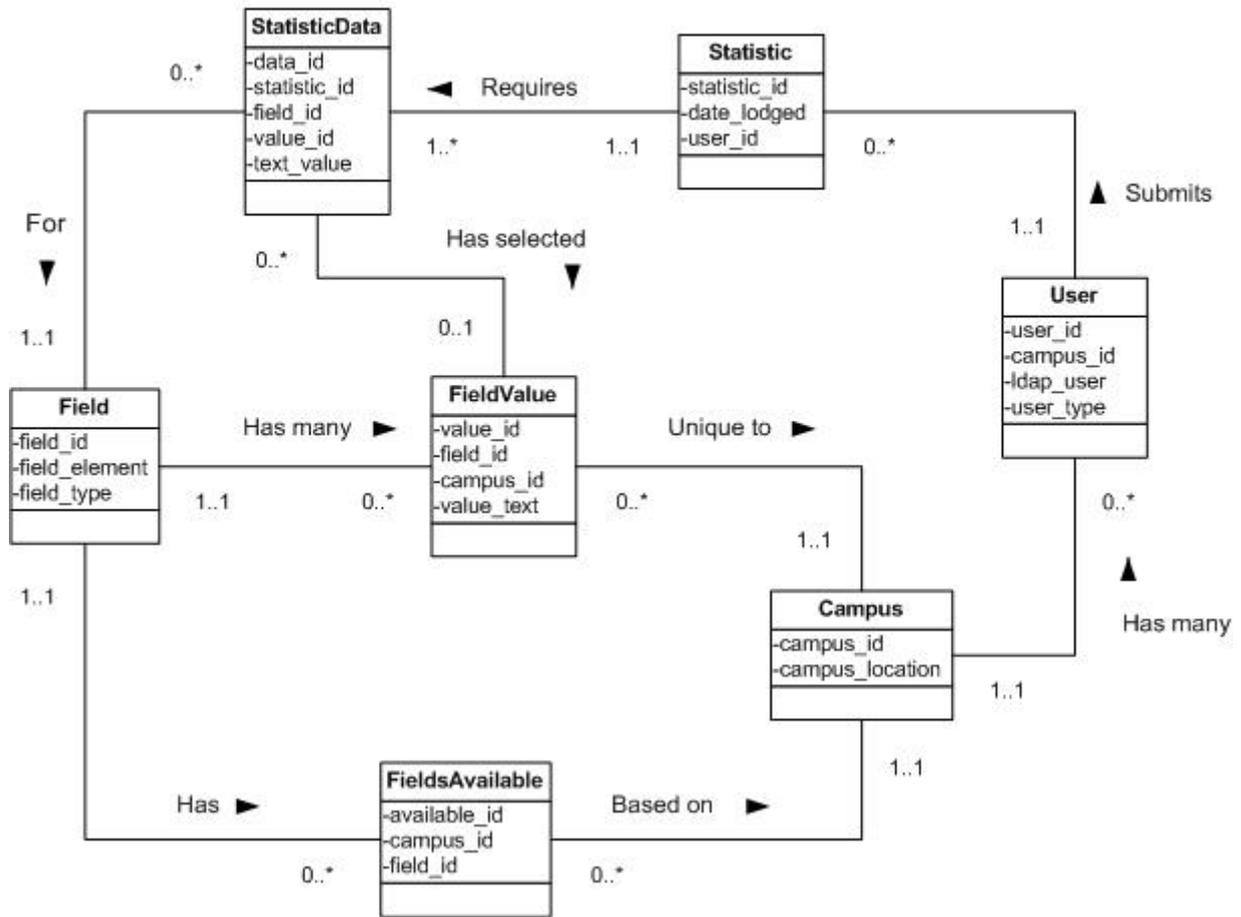
4 Class Diagram

Array



5 Entity Relationship Diagram

Entity Relationship Diagram



6 Relational Database Design

Campus (Campus_ID, Campus_Name)

Primary Key Campus_ID

Alternate Key Campus_Name

Foreign Key

ON UPDATE CASCADE, ON DELETE CASCADE

Field (Field_ID, Field_Element)

Primary Key Field_ID

Alternate Key

Foreign Key

ON UPDATE CASCADE, ON DELETE CASCADE

FieldsAvailable (Available_ID, Field_ID, Campus_ID)

Primary Key Available_ID

Alternate Key

Foreign Key Field_ID **references** Field (Field_ID)

Foreign Key Campus_ID **references** Campus (CampusID)

ON UPDATE CASCADE, ON DELETE CASCADE

FieldData (Data_ID, Statistic_ID, Field_ID, Field_Data_ID, Value_ID)

Primary Key Data_ID

Alternate Key

Foreign Key Statistic_ID **references** Statistics (Statistic_ID)

Foreign Key Field_ID **references** Field (Field_ID)

Foreign Key Field_Data_ID **references** FieldData (Data_ID)

Foreign Key Value_ID **references** FieldValues (Value_ID)

ON UPDATE CASCADE, ON DELETE CASCADE

FieldValues (Value_ID, Field_Text, Field_ID)

Primary Key Value_ID

Alternate Key

Foreign Key Field_ID **references** Field (Field_ID)

ON UPDATE CASCADE, ON DELETE CASCADE

Statistics (Statistic_ID, Date_Submitted, User_ID)

Primary Key Statistic_ID

Alternate Key

Foreign Key User_ID **references** User (User_ID)

ON UPDATE CASCADE, ON DELETE CASCADE

User (Username, First_Name, Last_Name, User_Type, User_State, Campus_ID)

Primary Key Username

Alternate Key First_Name, Last_Name, User_State, Campus_ID

Foreign Key Campus_ID **references** Campus (Campus_ID)

ON UPDATE CASCADE, ON DELETE CASCADE

7 Screen Design

7.1 Introduction

The Information Literacy Sessions Form system will be a web based application. It will be available through any web browser; however it will only be extensively tested in recommended browsers.

The system will follow web conventions, it will contain:

- Small client side files, for access speed and performance.
- A resizable interface, with high contrast colours and easy to read monitor friendly fonts to suit the wide variety of hardware it will be accessed from.
- A familiar navigation menu and layout. Major menu links will remain static and be placed along the top of each page. Other conventions will be adhered to, including the logout button being placed on the top right of the page and the save or submit button on the bottom right.
- Use familiar colours, blue font for hyperlinks, greyed out boxes for inactive fields and red for errors.
- The forms will be keyboard user friendly. Users will be able to tab through each option, increasing speed of data entry.

7.2 Login page

A user will access the system using a web browser. They are first presented with a login screen. Users can not access any functionality without logging in.

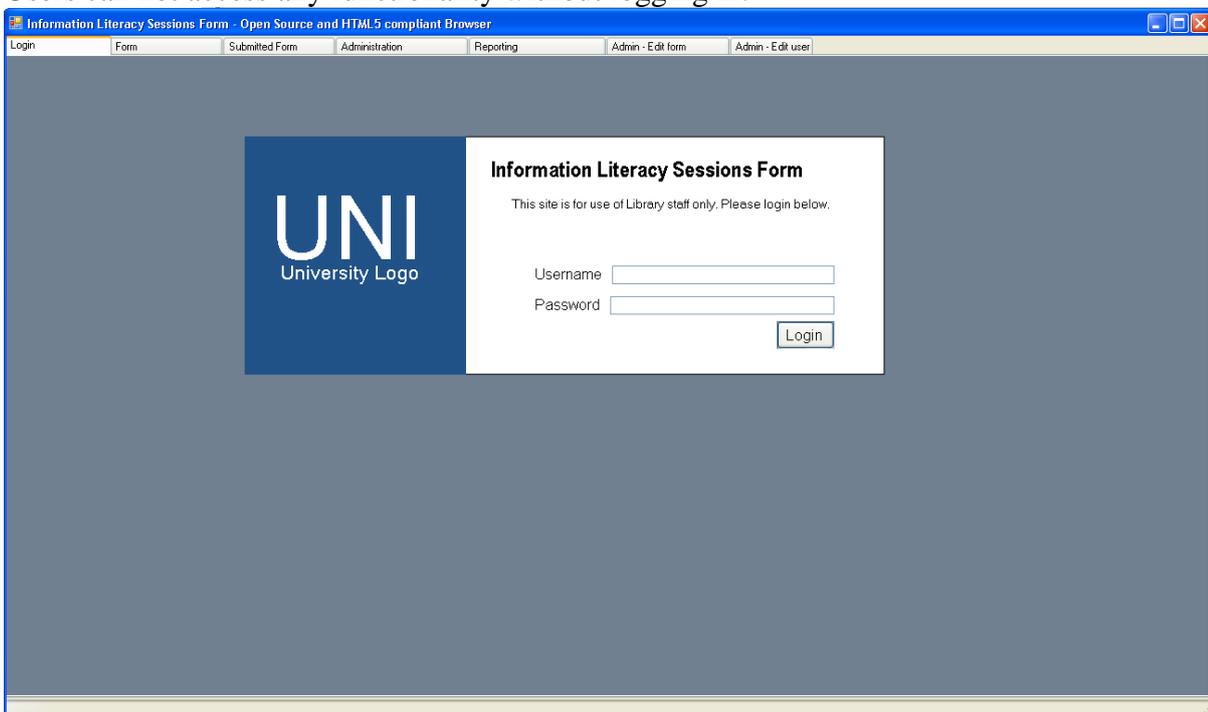


Figure 1 - Login Page

7.3 Navigation Menu

The forms page functions as the home screen, as it is the most common task for users. It contains the navigation menu, which is common to all forms. The navigation menu allows easy access to the different functionality of the system and provides reference so users know where they are. Most links will be available to all users, with the exception of the administration link, which will only be visible to users with admin privileges.

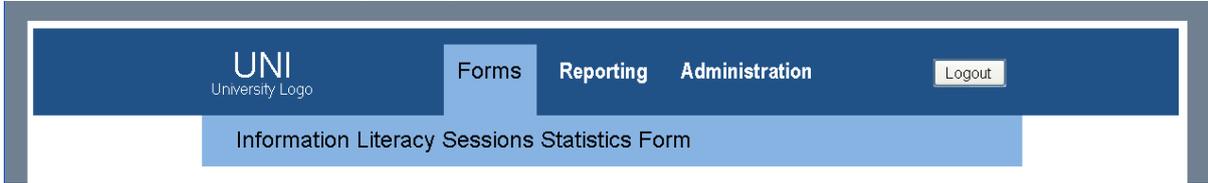


Figure 2 - Navigation Menu

7.4 Forms page

The forms page allows users to enter the details for a new session. The first question asks which campus and institution the session was for. The rest of the form will then dynamically change so only relevant questions are asked based on the campus selected. For example, there are a different set of training rooms available for Ourimbah and Newcastle campuses. If Ourimbah is selected, only Ourimbah rooms will be available to select.

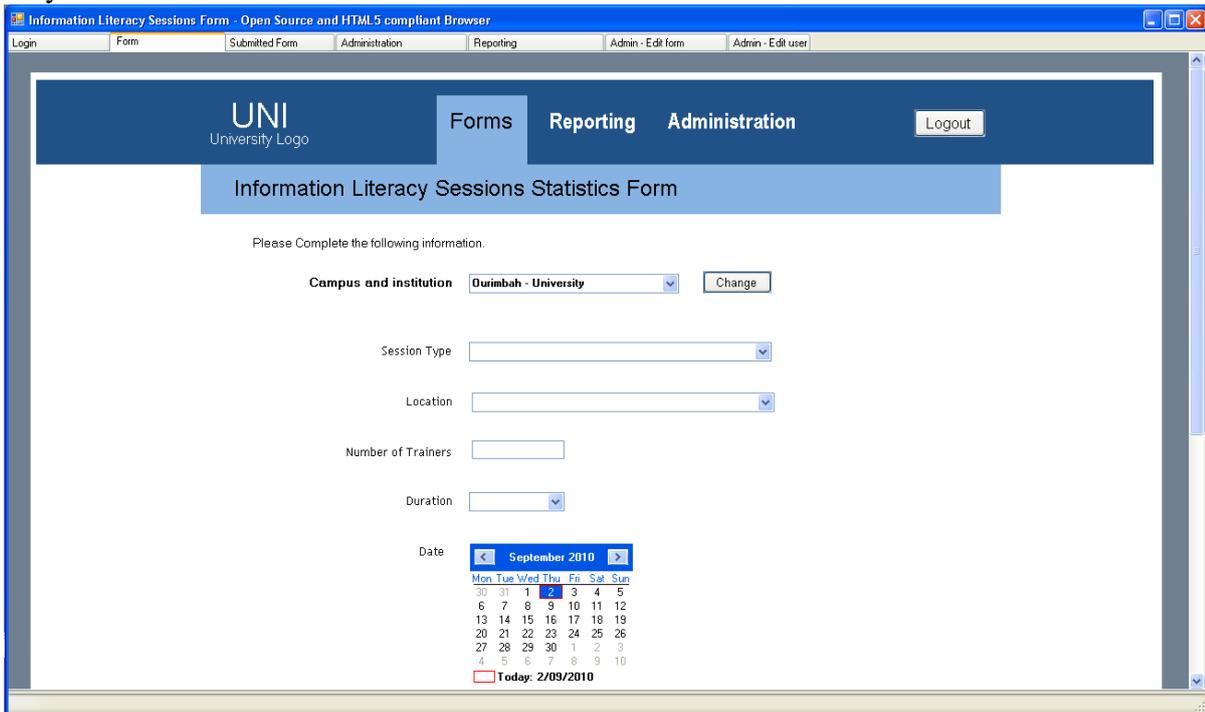


Figure 3 - Forms Page (Top Half)

The date selection is in a simple calendar allowing users to quickly select the date of the session. Other fields are either text boxes allowing manual input or drop down boxes, allowing users to select the relevant data. At the bottom of the page is a submit button and below that a small reset form link in case all fields needs to be reset.

The screenshot shows a web browser window titled "Information Literacy Sessions Form - Open Source and HTML5 compliant Browser". The browser's address bar and tabs are visible at the top. The main content area contains the following form fields:

- Duration:** A dropdown menu set to "1 hour 30 mins".
- Date:** A calendar for "September 2010". The date "2" (Tuesday, September 7th) is selected and highlighted with a red box. Below the calendar, it says "Today: 2/09/2010".
- Number of attendees:** A text input field containing the number "54".
- Reason for attending:** A dropdown menu with "Part of a class or course" selected.
- Preparation time:** A text input field containing the number "2", followed by the text "hours".
- Comments (optional):** A large empty text area.

At the bottom right of the form is a "Submit" button. Below the form, there are three links: "Home", "Change Campus and institution", and "Reset form". At the very bottom, there is a small copyright notice: "Copyright 2010 B3L Group for the use of U. of L. University only".

Figure 4 - Forms Page (bottom half)

7.5 Form submitted page

Once the form is submitted, the interface displays a confirmation page that allows users to review the information entered. If the information is correct, users can either create a new form using the link, navigate elsewhere using the top navigation menu or logout. If the user discovers errors in the data, they can click edit details and be returned to the enter form page where they can make changes.

Information Literacy Sessions Form - Open Source and HTML5 compliant Browser

UNI University Logo

Forms Reporting Administration Logout

Information Literacy Sessions Statistics Form

Your form has been successfully submitted [New Form](#)

Please review the data entered below and edit details if any information is incorrect

Session Type	Orientation Tour
Location	Tour - around campus
Number of Trainers	1
Duration	1 hour 30 mins
Date	3rd July 2011
Number of attendees	37
Reason for attending	Voluntary
Preperation time	1 hour
Comments (optional)	

[Edit Details](#)

Figure 5 - Form Submitted Page

7.6 Reporting page

The reporting page allows users to view and download reports. Reports available will vary due to users access levels. Users can either select to view the report in browser or export it into a variety of formats, ready for import into other systems. Initially a few reports will be built into the system; custom reports can be created and modified in the administration menu.

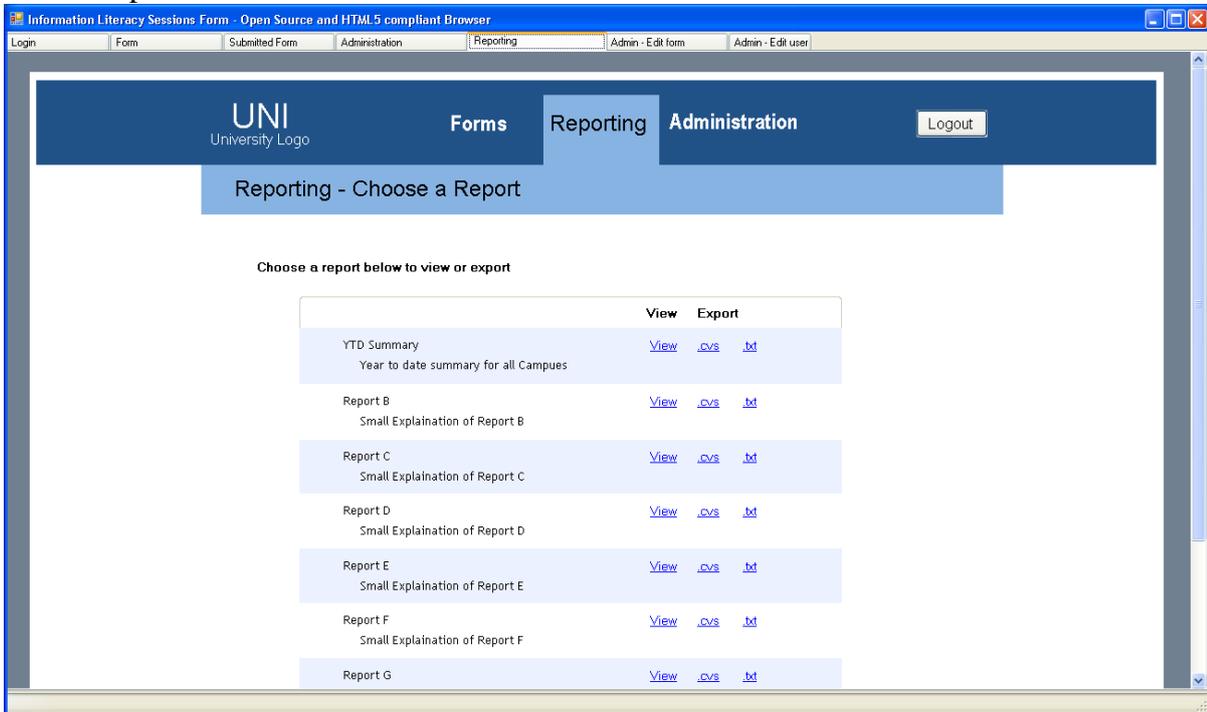


Figure 6 - Reporting Page

7.7 Administration menu

This menu will only be available to users with appropriate access. It allows admin staff to create and edit users, edit forms and edit reports.

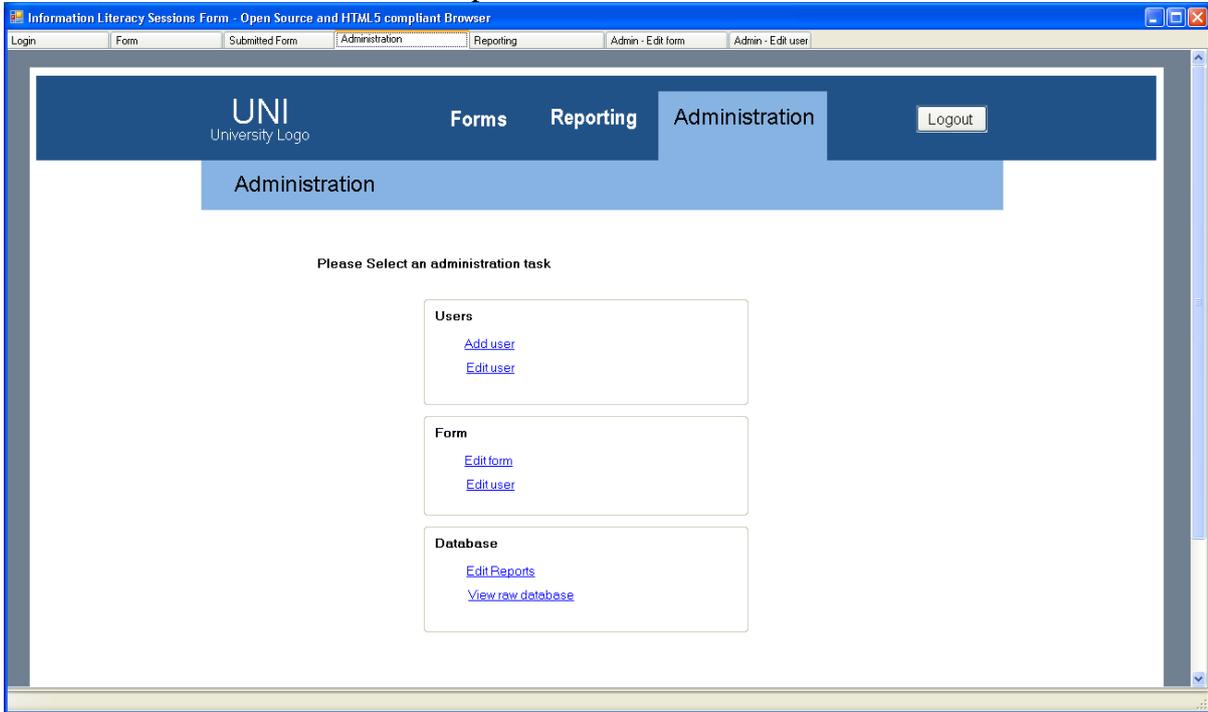
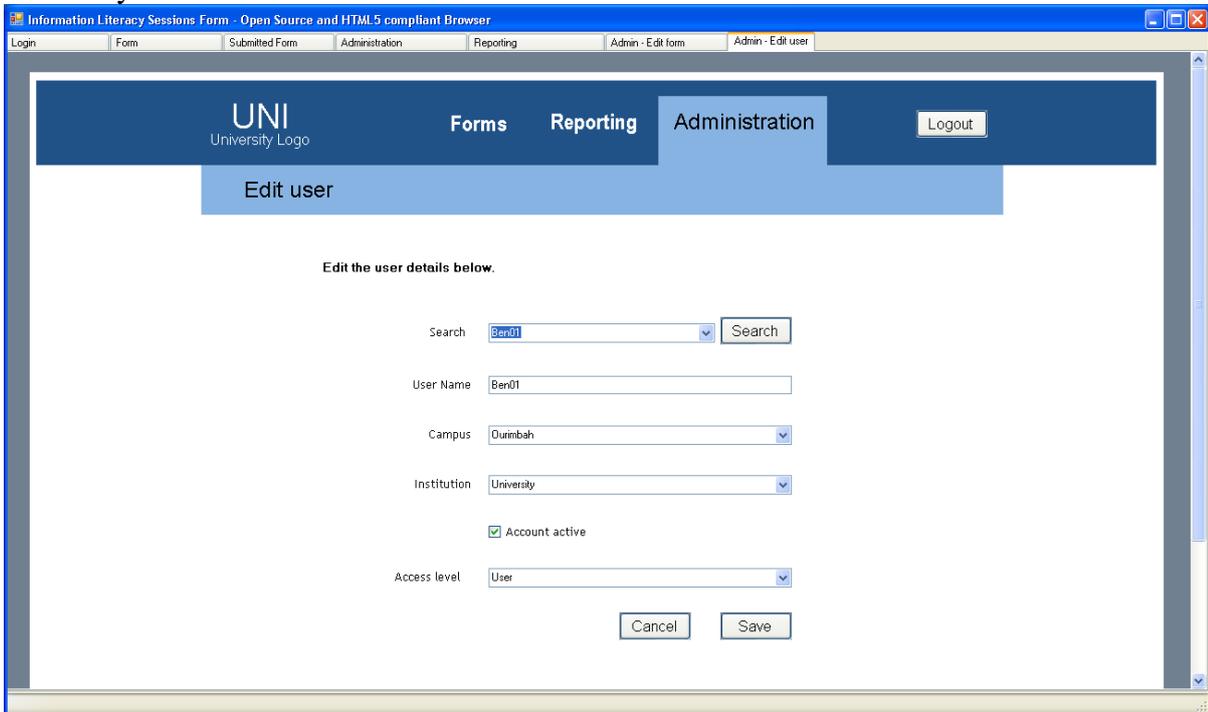


Figure 7 - Administration Menu

7.8 Administration – Edit user

Admin staff can edit a user by first searching for a user. Once the user is selected, their details can be edited. Note there is no new password field as authentication is handled by the LDAP server externally.



The screenshot shows a web browser window titled "Information Literacy Sessions Form - Open Source and HTML5 compliant Browser". The browser has several tabs open: "Login", "Form", "Submitted Form", "Administration", "Reporting", "Admin - Edit form", and "Admin - Edit user". The "Administration" tab is active, and the "Edit user" page is displayed. The page has a dark blue header with the "UNI University Logo" on the left, navigation links for "Forms", "Reporting", and "Administration" in the center, and a "Logout" button on the right. Below the header, there is a light blue banner with the text "Edit user". The main content area is white and contains the following form elements:

- A heading: "Edit the user details below."
- A search field with a dropdown menu containing "Ber01" and a "Search" button.
- A "User Name" text input field containing "Ber01".
- A "Campus" dropdown menu with "Durimbah" selected.
- An "Institution" dropdown menu with "University" selected.
- A checked checkbox labeled "Account active".
- An "Access level" dropdown menu with "User" selected.
- "Cancel" and "Save" buttons at the bottom.

Figure 8 - Edit User

7.9 Administration – Edit form

Admin staff will be able to edit the details of the form. The field type, name options and visibility can easily be changed.

The screenshot shows a web browser window titled "Information Literacy Sessions Form - Open Source and HTML5 compliant Browser". The browser's address bar shows "Admin - Edit form" and "Admin - Edit user". The page contains a form with the following fields and options:

Field Name	Type	Options	Used for:
Session Type	Combo Box	Information Literacy Tutorial, Orientation Tour, EndNote, NEWCAT, Blackboard	<input checked="" type="checkbox"/> Quimbah - University, <input checked="" type="checkbox"/> Callaghan - Auchmuty, <input checked="" type="checkbox"/> Quimbah - TAFE, <input checked="" type="checkbox"/> Callaghan - Huxley
Location	Combo Box	Training Lab 1 (TL1), Training Lab 2 (TL2), Tour - around campus, Computer Lab 1	<input checked="" type="checkbox"/> Quimbah - University, <input type="checkbox"/> Callaghan - Auchmuty, <input checked="" type="checkbox"/> Quimbah - TAFE, <input type="checkbox"/> Callaghan - Huxley
Number of Trainers	Text Box		<input checked="" type="checkbox"/> Quimbah - University, <input checked="" type="checkbox"/> Callaghan - Auchmuty, <input checked="" type="checkbox"/> Quimbah - TAFE, <input checked="" type="checkbox"/> Callaghan - Huxley
Date	Date		<input checked="" type="checkbox"/> Quimbah - University, <input checked="" type="checkbox"/> Callaghan - Auchmuty, <input checked="" type="checkbox"/> Quimbah - TAFE, <input checked="" type="checkbox"/> Callaghan - Huxley
Number of attendee	Text Box		<input checked="" type="checkbox"/> Quimbah - University, <input checked="" type="checkbox"/> Callaghan - Auchmuty, <input checked="" type="checkbox"/> Quimbah - TAFE, <input checked="" type="checkbox"/> Callaghan - Huxley
Reason for Attendin	Combo Box	Voluntary, Part of a class or course	<input checked="" type="checkbox"/> Quimbah - University, <input checked="" type="checkbox"/> Callaghan - Auchmuty, <input type="checkbox"/> Quimbah - TAFE, <input checked="" type="checkbox"/> Callaghan - Huxley
Preparation Time	Text Box		<input checked="" type="checkbox"/> Quimbah - University, <input checked="" type="checkbox"/> Callaghan - Auchmuty, <input checked="" type="checkbox"/> Quimbah - TAFE, <input checked="" type="checkbox"/> Callaghan - Huxley
Comments	Text Box		<input checked="" type="checkbox"/> Quimbah - University, <input checked="" type="checkbox"/> Callaghan - Auchmuty, <input checked="" type="checkbox"/> Quimbah - TAFE, <input checked="" type="checkbox"/> Callaghan - Huxley

Buttons: Reset, Submit

Figure 9 - Edit Report

8 User Testing Procedures

8.1 Testing Approach

The application will undergo a series of recursive data driven functional tests which will be completed for each test item. The functional test will be reinitiated once the software development team have fixed any bugs identified; therefore relaunching another iteration of testing to be carried out on those same features.

The test cases developed should follow the outline already supplied within the test activities section, and include the testing operations for each pass criteria specified. The testing method goes to a more in-depth explanation of how the testing will be performed. The more traditional black box testing approach will be used; which will rely heavily on test cases.

The consistency throughout the application will be noted; which includes the overall look and appearance of the application and the ease of navigation. The testing procedures developed will include appropriate comments about the design, and any other aspects that should be altered; aiming to improve the consistency, and navigation ease.

8.2 Testing Activities

8.2.1 Create Report

USE CASE:	Create Report		
DESCRIPTION:	When a custom report is required, admin staff may create this report referencing certain fields in the database. Relevant users can then access this report.		
SUCCESS CRITERIA:	The desired report will be created, saved and viewable by other users. Data should also be correctly written to the database and confirmations should be correctly displayed on the screen.		
FEATURES:	Feature	Sub-Feature	Testing Method
	Database Fields for Custom Reporting	Display/View Database Fields	Create report console will be opened. When selecting the Custom report option, a list of available database fields should be displayed
		Select fields for custom report	Select all available fields and create custom report. Once the report is created, view the report
	Creation/Saving of Report	Create & Save Report	Create a new report, then saved for use by another user.
		Viewable by another user	Ensure another user can access the newly created report
	Edit an Existing Report	Edit & Save Report	Editing an existing report, then saved for use by another user.
Viewable by another user		Ensure another user can access the recently edited report	

8.2.2 System Administration

USE CASE:	System Administration		
DESCRIPTION:	A users access rights may need to be edited if they are a new user, have changed job positions or left the organisation		
SUCCESS CRITERIA:	User modification or addition is correctly written to database, and confirmations are correctly displayed on screen.		
FEATURES:	Feature	Sub-Feature	Testing Method
	Add User	Input validation	Non-standard characters will be entered as a new user
		Input validation	A greater number of characters than the database field can support will be entered as a new user
		Input validation	An existing users details will be entered as a new user
		Add new user details to database	A standard user's details will be entered into the application for a user to be created
	Edit User	Input validation	Non-standard characters will be entered as an edited user
		Input validation	A greater number of characters than the database field can support will be entered as an edited user
		Input validation	An existing users details will be entered as an edited user
Editing existing users' details in database		A standard user's details will be altered in the application	

8.2.3 Editing Form

USE CASE:	Editing Form		
DESCRIPTION:	Over time, field values will become out of date and will need to be maintained		
SUCCESS CRITERIA:	Form should be successfully edited, written to the database, and all confirmations are correctly displayed on the screen.		
FEATURES:	Feature	Sub-Feature	Testing Method
	Edit Field (These features apply to each field)	Input validation	Non-standard characters will be entered
		Input validation	A greater number of characters than the database field can support will be entered
		Edit Field Values	The selected field will be edited with valid characters

8.2.4 View Reports

USE CASE:	View Reports		
DESCRIPTION:	Admin staff and Library staff can view a selection of pre-designed reports		
SUCCESS CRITERIA:	The desired report will be created and displayed on the screen.		
FEATURES:	Feature	Sub-Feature	Testing Method
	Run report from predefined template	N/A	Selected report will be run from application interface

8.2.5 New Form Entry

USE CASE:	New Form Entry		
DESCRIPTION:	Details are entered into the form and submitted		
SUCCESS CRITERIA:	New form is entered with correct input validation, written to the database and all confirmations are correctly displayed on the screen.		
FEATURES:	Feature	Sub-Feature	Testing Method
	Input data into Field (These features apply to each field)	Input validation	Non-standard characters will be entered
		Input validation	A greater number of characters than the database field can support will be entered
		Input Field Values	The selected field will be edited with valid characters
	Confirmation of Submission	N/A	Enter new form, with valid field values and submit

8.2.6 Validate User Access

USE CASE:	Validate User Access		
DESCRIPTION:	Admin staff and Library staff can view a selection of pre-designed reports		
SUCCESS CRITERIA:	User is correctly granted or denied access to system components according to the Access Control List (ACL)		
FEATURES:	Feature	Sub-Feature	Testing Method
	Validate User Access	Grant Access	Login with user that access granted to system
		Deny Access	Login with user that has no access to the system