**Infrastructure Design**

**Phonebook administration guide**

Prepared for:

**London Business School**

Author:

**Julian Rawcliffe**

## **Document Information**

|  |  |
| --- | --- |
| **Title** | Infrastructure Design |
| **Subject** | Phonebook administration guide |
| **Version** | 1.0 |
| **Status** | DRAFT |

## **Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 | 08/10/13 | Initial draft. |
|  |  |  |
|  |  |  |
|  |  |  |

**Contents**

Document Information

Revision History

[**Contents**](#h.1fob9te)

[**1** **Phonebook administration guide**](#h.3znysh7)

[1.1 Introduction](#h.3znysh7)

[1.2 Key stakeholders](#h.2et92p0)

[1.3 Usage within London Business School](#h.tyjcwt)

[1.4 Aim of this document](#h.3dy6vkm)

[**2** **Authentication and authorisation**](#h.1t3h5sf)

[2.1 Login](#h.4d34og8)

[**3** **Role management**](#h.2s8eyo1)

[**4** **Categories**](#h.17dp8vu)

[**5** **Departments**](#h.3rdcrjn)

[5.1 Viewing departments](#h.26in1rg)

[5.2 Editing departments](#h.lnxbz9)

[5.3 Deleting a department](#h.35nkun2)

[**6** **Sub-departments**](#h.1ksv4uv)

[6.1 Listing sub-departments](#h.44sinio)

[6.2 Editing sub-departments](#h.2jxsxqh)

[6.3 Creating new sub-departments](#h.z337ya)

[6.4 Deleting sub-departments](#h.3j2qqm3)

[**7** **Extension ranges**](#h.1y810tw)

[**8** **Locations**](#h.4i7ojhp)

[**9** **Unmanaged data**](#h.2xcytpi)

[**10** **Activity logging**](#h.1ci93xb)

[**11** **Development requirements**](#h.3whwml4)

[**12** **Application roadmap**](#h.2bn6wsx)

[**13** **Appendix 1**](#h.qsh70q)

# **Phonebook administration guide**

## **Introduction**

The phonebook website is a Ruby on Rails web application to manage the data required for the provisioning of directory records suitable for synchronisation to an offsite, out-sourced telephony provider.

## **Key stakeholders**

The key users of the system are the IT Support and infrastructure teams.

## **Usage within London Business School**

The target audience for this document is the infrastructure team who will be responsible for backend administration of the site data.

The document does not describe details of the flow of data into and out of the website database which is documented elsewhere.

## **Aim of this document**

The aim of this document is to describe the administrative functions available and how to perform key operations.

This document does not describe management of actual extension records or any of the data flow and external processing of the site data as this is documented elsewhere.

# **Authentication and authorisation**

All insert, update and delete operations must be performed by an authenticated user.

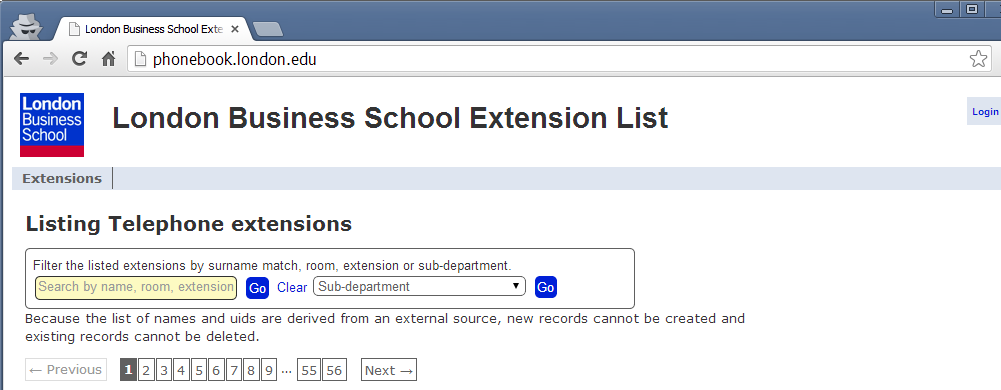
The website does not allow administrative functions to be performed by authenticated users who are given the admin role through the users\_roles table. Members of the edit role can update extension details for users (from the phones table) but cannot add, update or delete any other type of record.

The website uses an LBS Active Directory server as the provider of an LDAP service against which authentication can be performed.

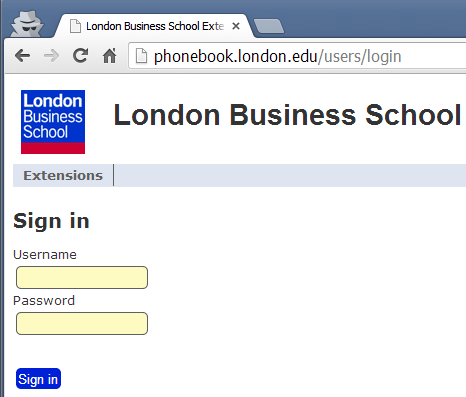
Although can view sections such as sub-departments or rooms by typing the correct URL and may even be able to display the edit form, submitting the form will have no effect and will result in an ‘Access denied’ message being displayed.

## **Login**

To access any of the administrative functions, click the ‘Login’ link at the top-right of the phonebook site,

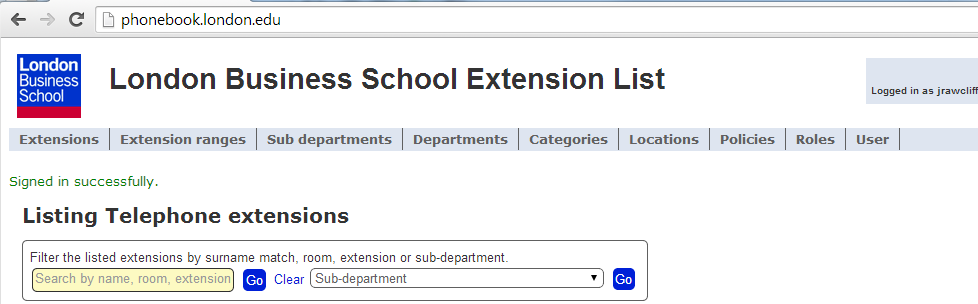


This will display the following form,



Sign in using Active Directory credentials. Upon first login, no permissions will be granted, not even to manage extension records.

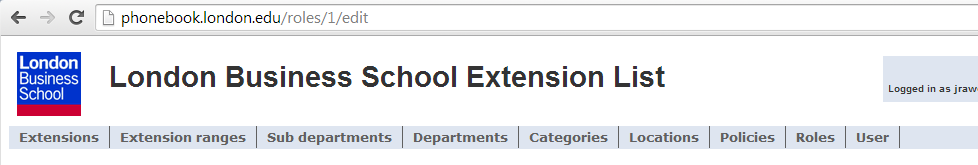
Success login will redirect back to the extension list page with a brief message to indicate successful login, text showing the name of the logged in user along with a navigation bar showing the full range of permitted management operations.



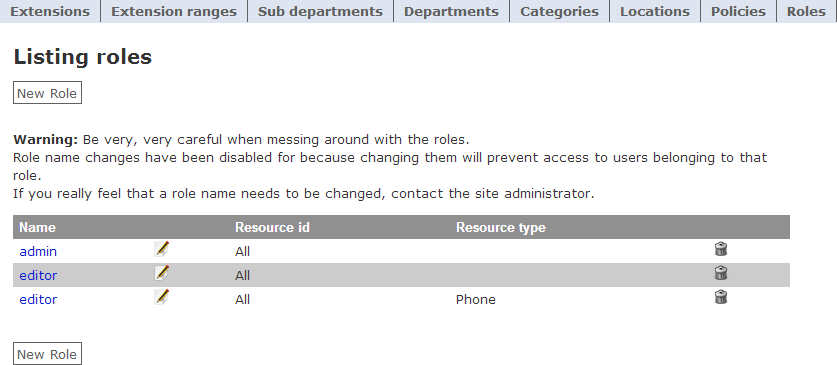
To logout, click the ‘Logout link next to the name of the logged in user.

# **Role management**

The Roles management section of the phonebook site is accessed by clicking on the ‘Roles’ navigation link at the top of the page,

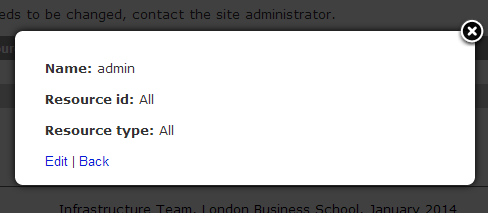


This will then display a list of known roles,



The ‘Resource Id’ and ‘Resource type’ columns are not currently used and can be disregarded.

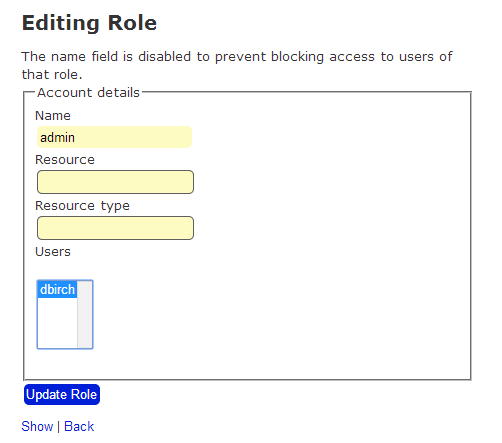
Click on the name of the role for a modal popup of the role’s details,



Click the edit link to bring up the edit form for the role (the edit icon to the right of the Role name can also be used to display the edit form).

The Users multiple select element will show all users that have not yet been assigned to a role; a user may only be assigned to a single role.

To add a user to the role, simply highlight the users to be added (use CTRL-click to select the second and subsequent users) and press the ‘Update role’ button; the Resource fields should be left blank.



Note that the form displays a warning about changing the name of a role. This is because the controllers for the different sections of the site include specific references to the names of the roles when granting access. This is a notoriously difficult activity to achieve without complex coding which is not justified on a simple site such as this.

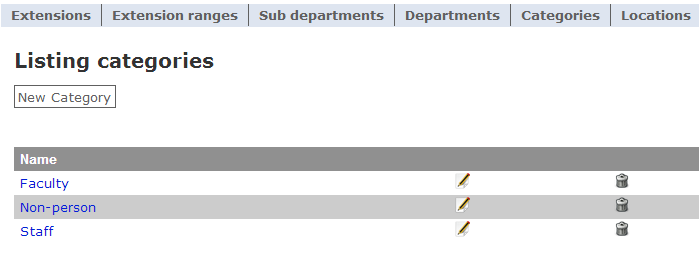
The name element of the role is disabled to prevent changes being made. To make sure that name changes are not applied, the name field has been excluded from the list of permitted fields in the roles controller. A Rails test should be written to confirm this behaviour.

There’s currently no way of removing a user from a role, but this can be achieved by updating or removing the user’s record from the users\_roles database table.

# **Categories**

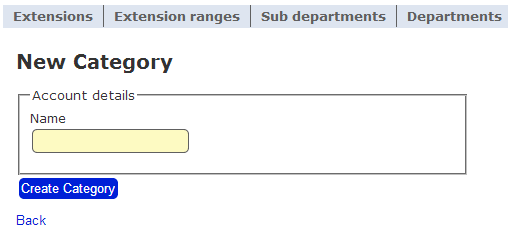
Departments are categorised in one of 3 ways,

* Faculty,
* Staff,
* Non-human



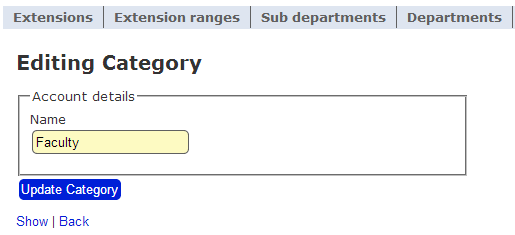
Management of the categories is very straightforward.

Click the ‘New category’ link to bring up the form for a new category,



And enter a name for the category. Press ‘Create Category’ to save the new entry.

Click the ‘Edit’ icon to display the form to edit the name of an existing category,



Press ‘Update Category’ to save the details.

The ‘Show’ link shows the category details in a modal popup.

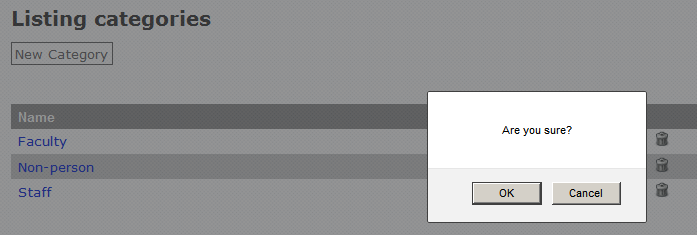


Press ‘Esc’ or click the ‘X’ icon to clear the display.

## **Deleting categories**

Removing a category is achieved by clicking the delete icon on the right-hand side of the index list.

A confirmation prompt will be displayed.



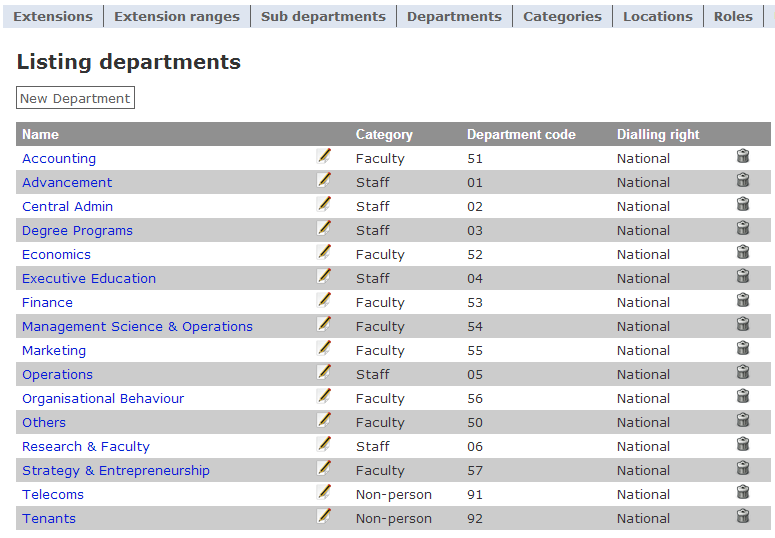
The behaviour of the application with regard to what effect removal of a category record will have on any foreign key references elsewhere in the database is undefined and subject to discussion and inclusion in a later release of the application.

# **Departments**

Departments are one of the most difficult areas for management of general data for the School to manage and, initially at least, the phonebook website is providing an authoritative source of departmental information which can be flowed to other applications particularly SchoolDB (EAD database) where the department is stored as a free text field.

## **Viewing departments**

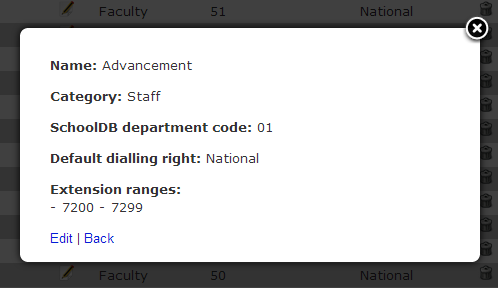
Click on the ‘Departments’ navigation link to show a list of departments,



This list displays the basic department details,

* Name
* Category – Staff, faculty or non-human
* Departmental code – combines with sub-department code to form unique identifier
* Default dialling right – applied to a telephone extension if not overridden on an individual basis.

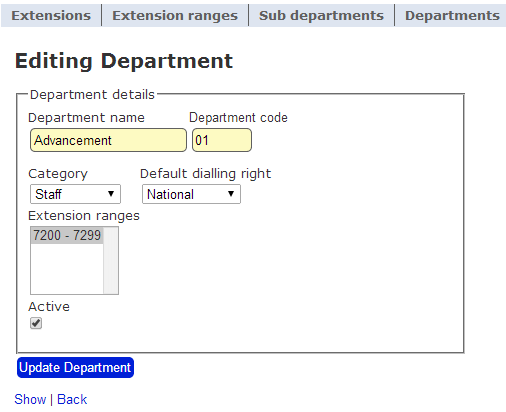
The detail can also be displayed in a modal dialog box by clicking on the department name,



This view also shows extension ranges that have been allocated to the department.

## **Editing departments**

Press the ‘Edit’ link or icon to bring up the form to edit the details of the department,

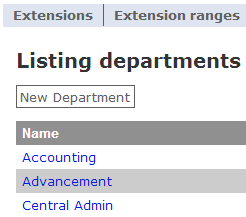


The department name and code are strings which can make enforcing uniqueness difficult. Category and default dialling right are dropdown lists from their respective tables.

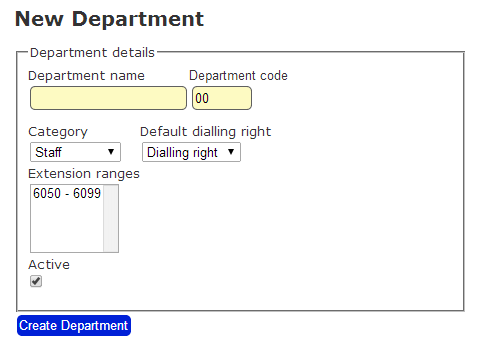
The Extension ranges list shows ranges currently allocated to a department as selected, along with any other ranges that have not assigned to another department. Selecting an extension range is not mandatory.

Create new department

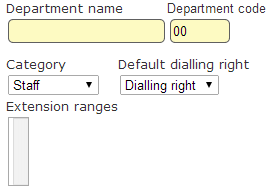
Creating a new department is simply a matter of pressing the new department link on the departments index page,



The new department form is shown below,



The category and dialling right fields are required, but extension ranges can only be assigned when available; it is not a required field and can be assigned to a department using the extension range form.

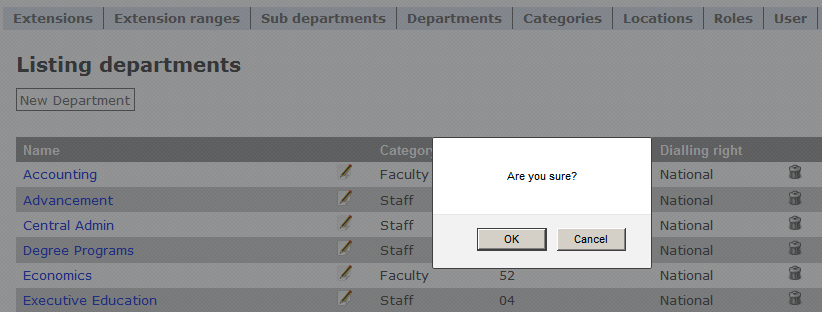


The above diagram shows that no extension ranges are available but this will not prevent the form from being submitted.

## **Deleting a department**

The delete icon at the right-hand side of the index page can be used to delete a department.

The user will be prompted for confirmation of the removal request,



Removal of a department will also remove the association to any extension ranges that have been assigned.

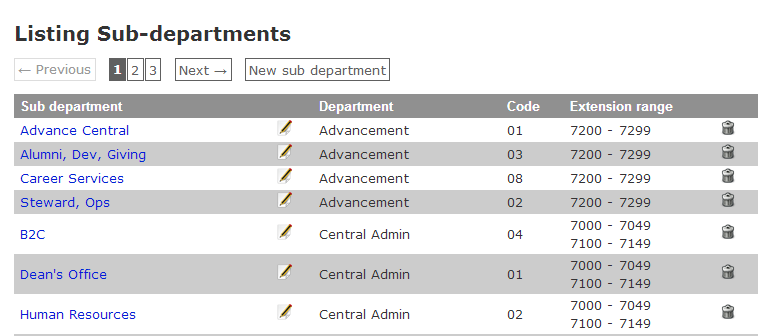
The behaviour of the application with regard to what effect removal of a department will have on any foreign key references elsewhere in the database is undefined and subject to discussion and inclusion in a later release of the application.

# **Sub-departments**

Sub-departments represent the various grouping and teams within departments.

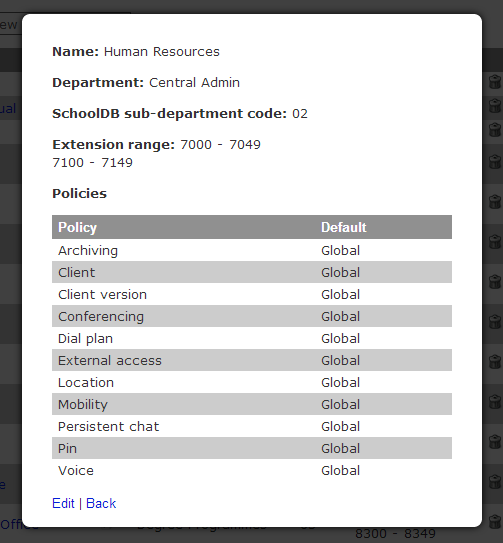
## **Listing sub-departments**

A list of current sub-departments can be found by clicking the ‘Sub-departments’ link in the navigation link.



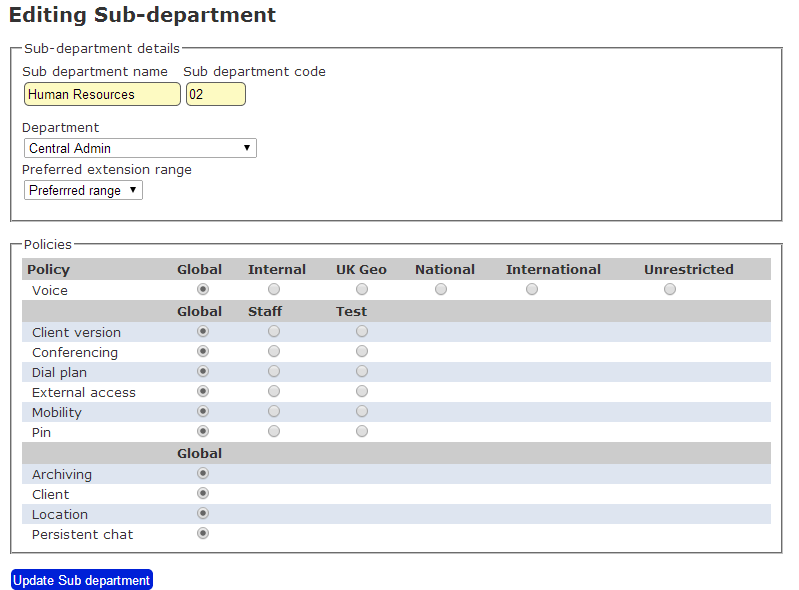
Use the pagination links to move forward and backward through the list.

Click on the name of the sub-department to view the details in a modal dialog,



## **Editing sub-departments**

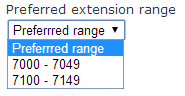
To edit a sub-department, click the ‘Edit’ link in the modal dialog or click the edit icon to the right of the name on the index page. The following form will be displayed,



The Sub-department name and code fields are strings.

The Department is a required field selected from the dropdown list.

The preferred extension range is an optional override from the departmental extension range list,

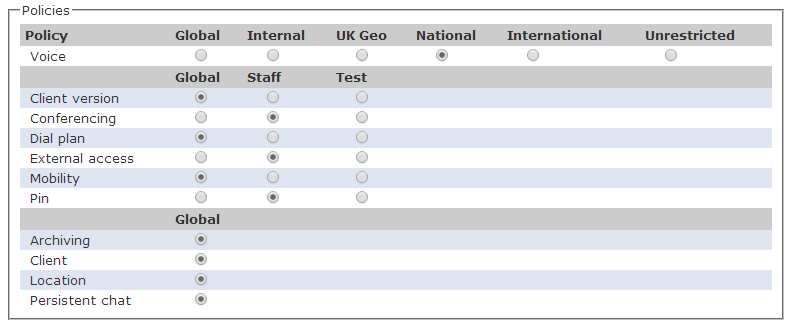


so that the extension numbers available for this sub-department will only be drawn from this list rather than whole range for the department.

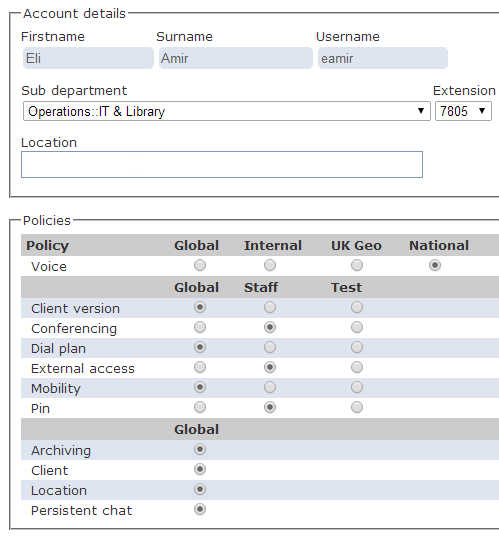
The policies section lists those currently configured with the telephony provider and the selections are the defaults given when a sub-department is initially assigned to an extension record; they can be modified an individual basis.

Changing the sub-department policy settings will apply an update to all extension (phones) records where accounts that had the previous policy setting will pick up teh new value.

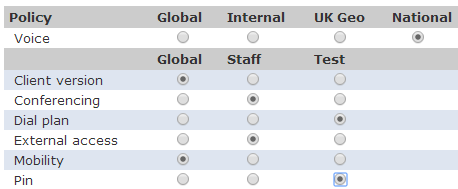
Consider the following sub-department defaults,



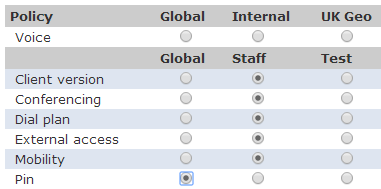
Assigning an account to the IT & Library sub-department will initially show the following,



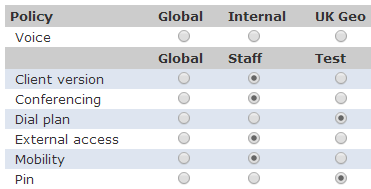
with a subsequent update to be,



An update to the default sub-department policy of,



Will have the following effect on the indiviual account,

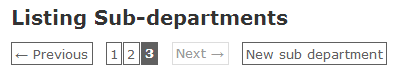


An individual account policy setting that does not match the sub-department default will not be affected if that default policy setting is changed.

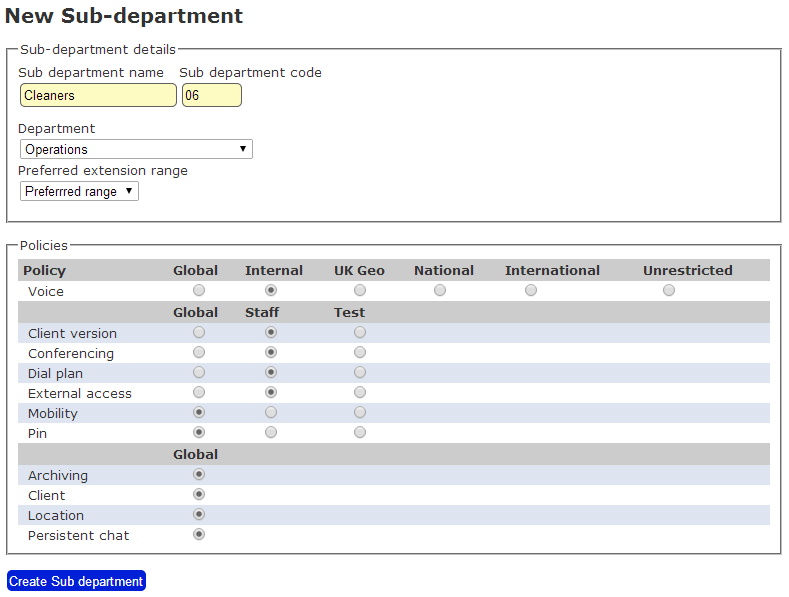
Any (and all) updates to extension accounts as a result of a sub-departmental policy change are recorded in the activity log, immediately before the record for the sub-department itself.

## **Creating new sub-departments**

A new sub-department can be created using the ‘New sub-department’ link next to the pagination links,



The following form will be displayed.



The sub-department name and code fields are free text; the department is selected from the dropdown list.

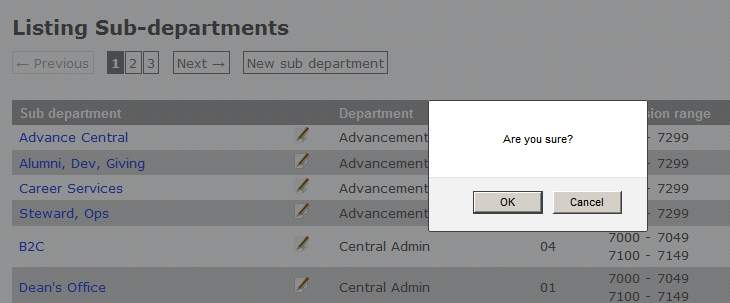
The preferred extension range is optional and only available when editing an existing sub-department although a request has been submitted to have this field populated dynamically when a department is selected. The policy buttons should reflect appropriate defaults for members of the sub-department. The list of policies is hard-coded into the form to match the policy table id values that represent the policy name populated in the Lync portal.

When finished press the ‘Create Sub department’ button to add the new entry.

## **Deleting sub-departments**

Deleting a sub-department is simply a matter of clicking the delete icon on the right-hand side of the index page.

A confirmation of the action will be requested,



The behaviour of the application with regard to what effect removal of a sub-department will have on any foreign key references elsewhere in the database is undefined and subject to discussion and inclusion in a later release of the application.

# **Extension ranges**

Extension ranges represent contiguous blocks of extensions that can be assigned.

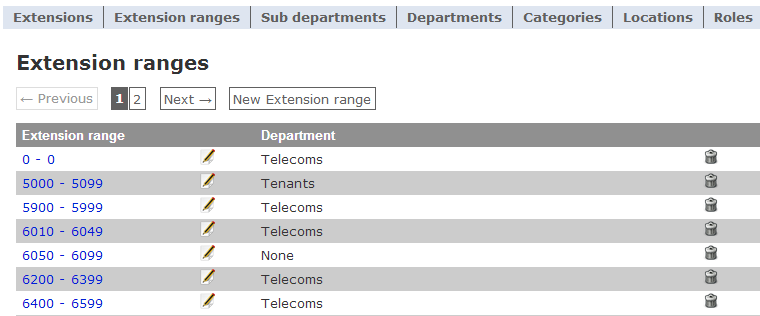
A by-product of the extension range notion is the need to maintain sets of extension records as a lookup reference when deciding which extensions have already been assigned to telephones.

Do not confuse this lookup table with the ‘Extensions’ navigation menu item which is used to maintain the extension records assigned to telephones. The contents of the extensions lookup table is managed silently by the extension\_ranges controller.

Managing individual extension details is described in the phonebook user guide.

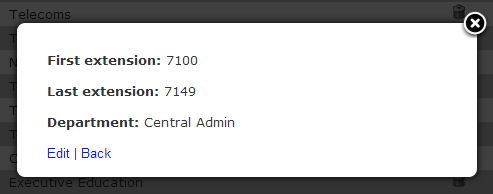
## **Viewing extension ranges**

Current extension ranges can be viewed by clicking on the ‘Extension ranges’ link in the navigation menu,



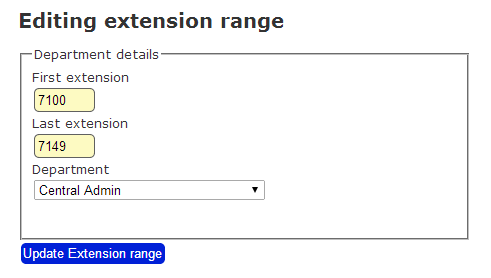
The pagination links can be used to list the remaining extension\_range items.

Clicking on the extension range first and last numbers will display a modal dialog box with the extension range details,



## **Editing extension ranges**

Clicking on the ‘Edit’ link or the edit icon to the right of the extension range link will display a form which can be used to edit the details,

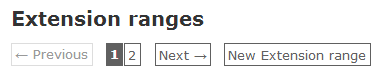


The first and last extension fields are free text fields, the department is optional and selected from the dropdown list. The extension range may also be assigned to a department through the departments controller.

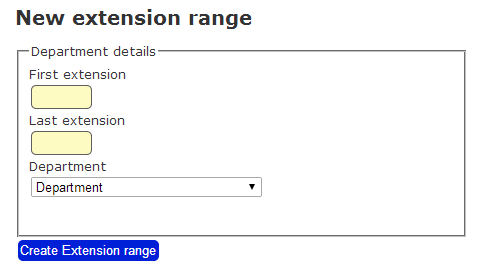
Changing the first or last value for an extension range does not currently change the records in the extensions table and won’t be reflected in the extensions available when editing individual phone records.

## **Creating extension ranges**

The extension range pagination links have a link to create a new extension range,



A new extension range can be created by completing the form details,

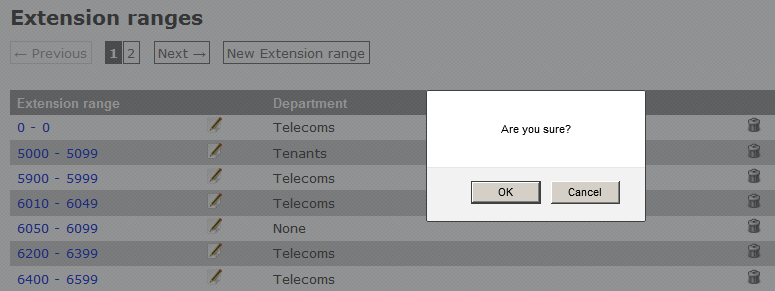


The first and last extension fields are free text fields, the department is optional and selected from the dropdown list. The extension range may also be assigned to a department through the departments controller.

Creating a new extension range triggers a private controller action to create extension records between the first and last values submitted for the range.

## **Deleting extension ranges**

An extension range may be deleted by clicking the delete icon on the right-hand side of the extension range index page.



Confirmation of the removal will be requested before proceeding.

Deleting an extension range will trigger a controller action to remove the associated records from the extensions table.

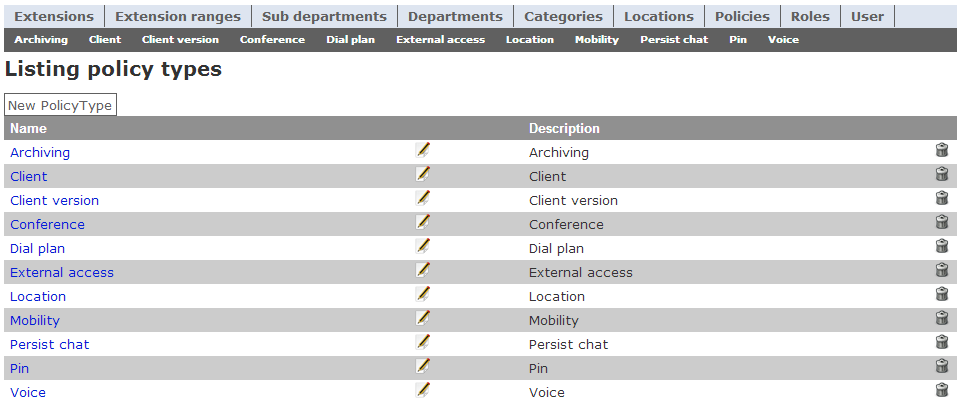
Currently no check is made whether there are phone records that have an extension reference (via the extensions table) between the first and last value of the extension.

The behaviour of the application with regard to what effect removal of an extension range will have on any foreign key references elsewhere in the database is undefined and subject to discussion and inclusion in a later release of the application.

# Policies

The policies menu link will display a list of the currently supported Lync policy types with a sub-menu[[1]](#footnote-0) linking to lists of the policies available for each type.

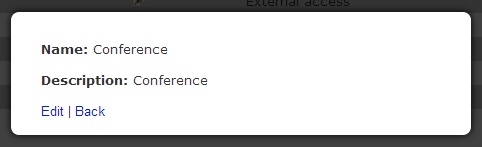
It is not expected that there will much need for frequent policy operations.



The policy types are provided more to provide a consistent database model structure so that individual policies can maintain a foreign key reference to a types table; the data in the Lync portal is inconsistent and difficult to map directly on to a simple database model.

## Viewing policy types

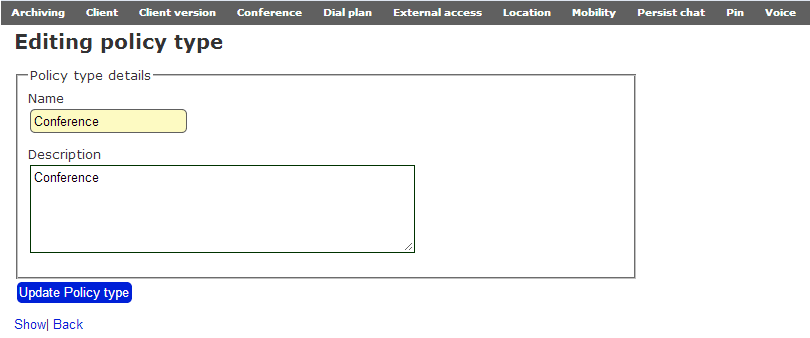
Simply click on the name of a policy type for a modal dialog diosplaying the details,



Press escape to close the popup.

## Editing policy types

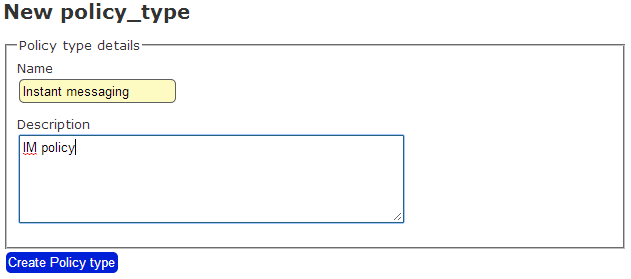
Either clicking the edit icon near the policy type name or the Edit link in the modal dialog will bring up the policy type edit page,



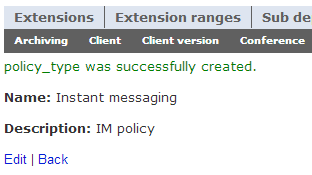
This is a very simple form with a unique name and a simple description. Press the update button to save any changes.

Creating policy types

Creating a new policy type is simply a matter of pressing the ‘New policy type’ button on the index page



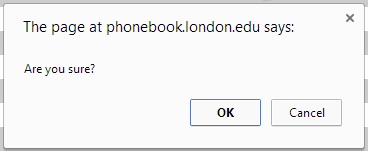
Press the ‘Create policy type’ button when complete. A manual update will be required to the policies layout template to have the new policy type included in the menu. It will also require a new Rails ‘*scaffold*’ (model, view, controller and other items) before any policies of that type can be defined.



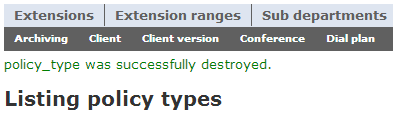
Press the back link to return to the main list.

Deleting policy types

To delete a policy type simply press the delete icon on the right-hand side and a confirmation prompt will be displayed.



Press OK to confirm and a message will be displayed on the index page.



Policy tables

There is some uncertainty around exactly how to best represent policy details within the extension management website but the specific policy tables and controller elements all have the same structure but are not really intended to be used to manage actual policies.

The simplest method of adding a new policy is by executing a SQL query along the lines of,

set identity\_insert pin\_policies on  
insert into pin\_policies (id, name, policy\_type\_id, description, lync\_policy\_name, created\_at, updated\_at) Values  
('5', 'Universal', 10, 'Universal PIN policy', 'PIN-Universal', Getdate(), Getdate())  
set identity\_insert pin\_policies off

The policy\_type\_id column needs to match the id column of that type from the policy\_types table.

Most importantly, the lync\_policy\_name must match the policy name that has been added to the Lync portal as it will be used by the provisioning code when specifying the level of service available to an account.

Creating a new policy by whatever means will require manual updates to the following:

* policy layout template in app/views/layouts
* the sub-department form in app/views/sub\_departments/\_form.html.erb
* the phones form in app/views/phones/\_form.html.erb

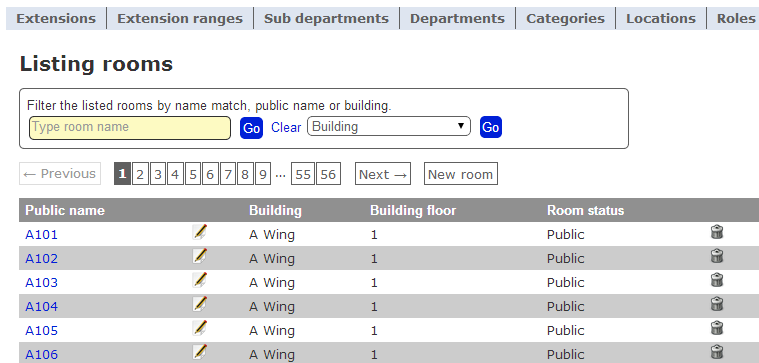
# **Locations**

Locations represent rooms, offices, lecture theatres and all other spaces around the school as identified by the Estates team in the Operations department.

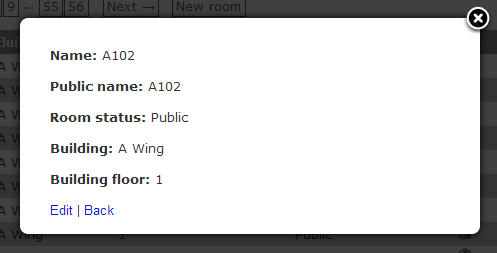
The location information maintained by the School is somewhat haphazard and prone to inconsistencies and has been arranged for the benefit of inclusion in the phonebook website in a manner that makes most sense for synchronising into Active Directory and hence into the outsourced telephony service.

## **Viewing location records**

The recognised locations may be viewed from the ‘Locations’ navigation link on the website,



Clicking on the public name link displays a modal dialog with details of the locations,

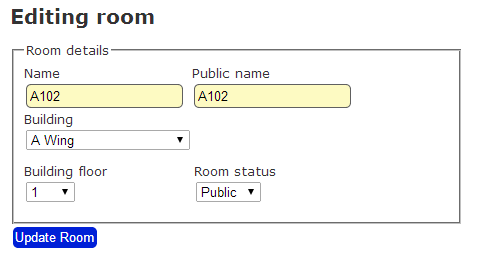


Telephone extensions are usually applied to locations with a public status.

## **Editing locations**

Clicking the ‘Edit’ link or the edit icon will display the form to modify the details of a location.

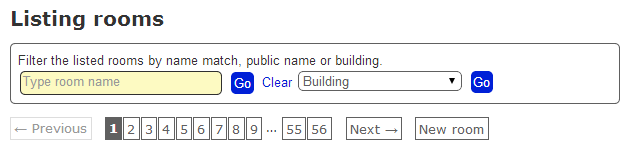
Since a room is unlikely to move between floors or buildings this actions is unlikely to be used very often.



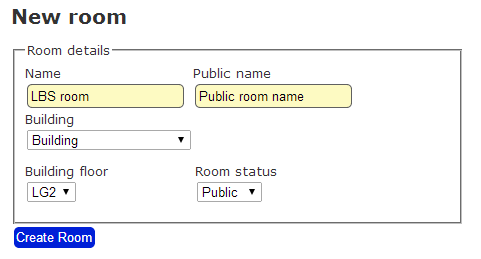
Press the ‘Update room’ button to save the changes.

## **New locations**

A new location can be added using the ‘New room’ link next to the pagination list on the index [page,



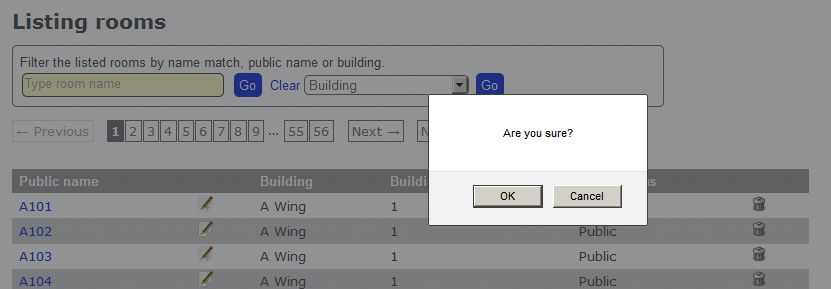
This will display the form for entering a new location with a default status of ‘Public.



Enter the relevant details and press ‘Create room’ to save the location. The room will immediately be available in the Location search field on the Extensions update form.

## **Deleting locations**

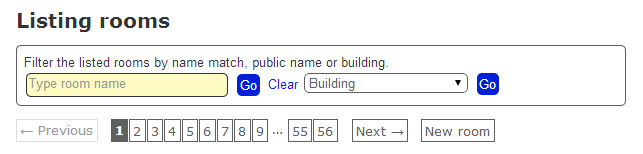
Deleting a location is simply a matter of clicking the delete icon at the right-hand side of the index list; a confirmation prompt will be displayed.



The behaviour of the application with regard to what effect removal of a location record will have on any foreign key references elsewhere in the database is undefined and subject to discussion and inclusion in a later release of the application.

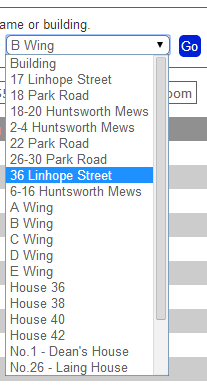
## **Searching locations**

Because of the number of locations in use a simple search form has been added to the locations section, similar to that used by the Extensions page, allowing simple text or building-based queries to be run.

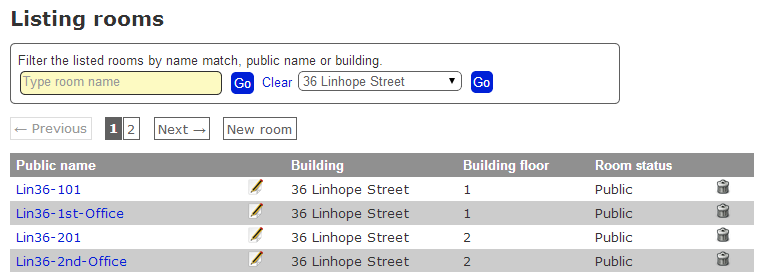


## **Search by building**

The search form contains a dropdown select list of all know buildings with locations,



Selecting a building an pressing the ‘Go’ button



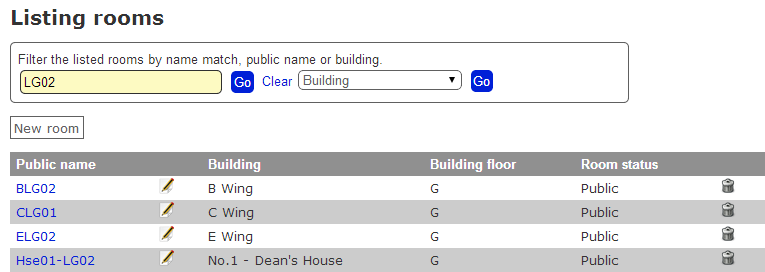
Notice how that pagination links are displayed when more than one page of results are returned. Any previous paginated selection will be cleared by the search.

The results are also ordered by room status with Public locations listed first.

To clear the building selection, either select ‘Building’ at the top of the list and press the adjacent ‘Go’ button or press the ‘Go’ button next to the search text box.

## **Search by name**

To search for a location by name type the name into the text box on the left and press the ‘Go’ button.



The search query wraps a SQL wildcard character on either side of the search text and the search is performed against the name and public\_name column of the locations table. Be warned that the names in the table are not the most obvious or friendly although efforts have been made to improve their utility compared to the raw data maintained by the Estates department.

To clear the search, press the ‘Clear’ link followed by either of the ‘Go’ buttons.

# **Unmanaged data**

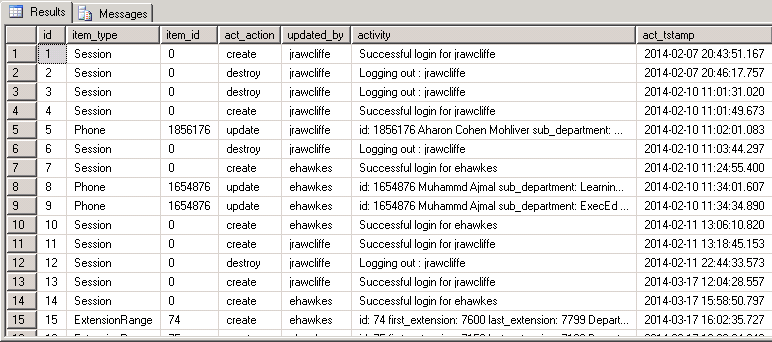
The following tables in the phonebook database do not currently have a management function on the website because the information contained in the tables is unlikely to ever need changing, or at least if it does it can easily be added directly into the database.

* buildings – Sainsbury, Taunton, Plowden, A wing, etc,
* building\_floors – LG2, LG, G, 1, 2, 3, 4, 5, 6.
* room\_statuses – public or hidden.
* dialling\_rights – internal, local, national or international (unlimited)

Models have to exist for these tables, but it would be a simple activity to create website sections (Views and Controllers) to enable these functions. The most difficult question regards how to link to them from the navigation menu.

# **Activity logging**

An important feature of the phonebook website is the recording of all insert, update and delete operations carried out through the site.

This information is stored in the activity\_logs table in a format based largely on the Papertrail rubygem (we don’t actually require the table versioning that it provides) and is applied to each controller that is accessible through the website.

An log record is created for all insert, update and delete operations. The item\_type entries for ‘Session’ are in reference to login and logout.

The log includes the name of the user performing the action and the timestamp of when it was attempted. There is no guarantee that the action listed in the log was actually completed, but it would not have thrown a fatal error.

Currently the only way to inspect the logs is by using SQL Server Management Studio to interrogate the activity\_logs table.

# **Development requirements**

## **Development platform**

The following elements need to be provided to support ongoing development of the administration interface.

* Ruby on Rails 4 hosting platform
* Preferably a MS SQL Server database. Although MySQL can be used for development, it should not be used to prepare data for incorporation into test or production instances.

## **Issue tracking and enhancements**

Any problems with site functionality or requests for improvements or additional features should be logged via http://casetracker.london.edu/ against the ‘telephone directory management’ project.

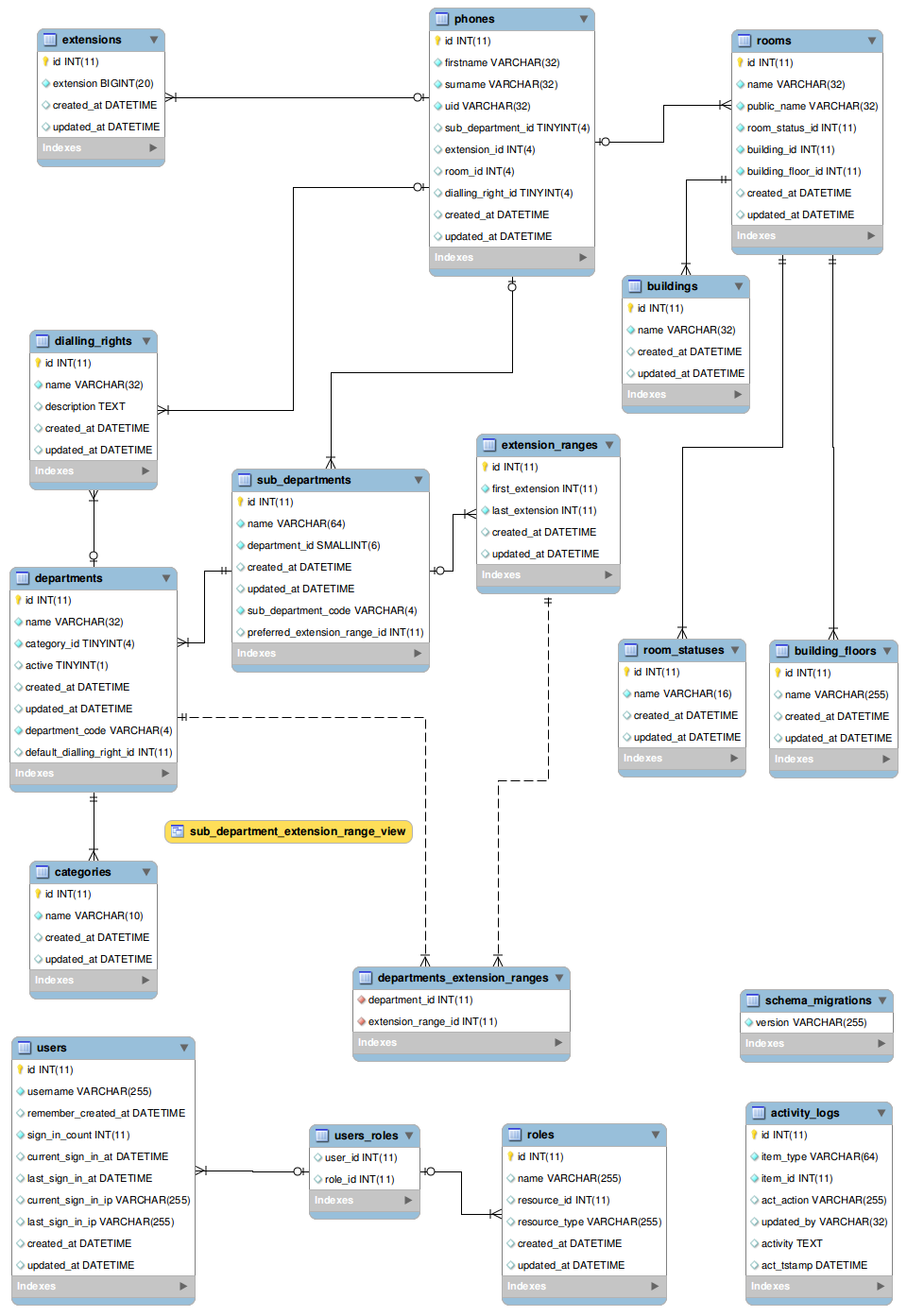
# **Application roadmap**

The following additional administration functions are in consideration for future releases:

* A view and controller for better user management.
* Improved error handling – routing to controller root rather than application root\_url
* Automated testing using Rails’ built-in capabilities.
* Use of foreign keys constraints across database table, although the ‘Rails way’ is that this should be handled through the model.
* Reporting – request volumes and user activity.

# **Appendix 1**

The following diagram shows the database schema ,



1. In Ruby on Rails terms, this involves using a separate layout, app/views/layouts/policies\_layout.html.erb [↑](#footnote-ref-0)