



**COS 301 Final year project  
Post-Doctoral management System**

**Functional requirements and application  
design document**

October 17, 2014

Version 4.0 (Final)

Iteration 6

**Prepared for**

**Client: Ms. Cathy Sandis (UP DRIS)**

**Supervisor: Prof. Stefan Gruner (SSFM Group)**

**by**

**Soft**Server** Group**

**Group members**

Kgothatso Phatedi Alfred Ngako (12236731)

Tokologo “Carlo” Machaba (12078027)

Mathys Ellis (12019837)

Change log			
Date	Version	Description	Person
10/02/2014	v 0.0	Original SRS document created	Mathys Ellis
02/03/2014	v 0.1	Added to glossary	Mathys Ellis
11/03/2014	v 0.2	Added more functional requirements which relate more to the use case diagrams	Alfred Ngako
16/03/2014	v 0.3	Added domain objects data relation diagrams	Alfred Ngako
16/03/2014	v 0.4	Added some preconditions and did some formatting	Mathys Ellis
17/03/2014	v 0.5	Added rest of preconditions and all the post-conditions. Also added to the glossary	Mathys Ellis
20/03/2014	v 0.6	Updated the domain objects diagram	Mathys Ellis
20/03/2014	v 0.65	Updated the domain objects plus diagram	Alfred Ngako
20/03/2014	v 0.7	Updated the use case prioritization	Alfred Ngako
16/03/2014	v 0.75	Did some restructuring and document for- matting	Mathys Ellis
17/03/2014	v 0.8	Added to the glossary	Mathys Ellis
12/05/2014	v 0.9	Created new Functional requirements and application design document.	Mathys Ellis
15/05/2014	v 0.91	Transferred necessary content from old SRS document. Performed editing and restructuring of document.	Mathys Ellis
15/05/2014	v 0.92	Added and redefined domain objects.	Mathys Ellis
15/05/2014	v 0.93	Added and redefined use case diagrams and preconditions.	Mathys Ellis
16/05/2014	v 0.95	Added and redefined use case diagrams and preconditions and postconditions. Also updated domain objects.	Mathys Ellis
19/05/2014	v 0.97	Added process specifications and added approval levels of applications.	Mathys Ellis
21/05/2014	v 0.98	Added all details for imports and exports service use case diagram and process specifica- tion for generate on-demand user account.	Mathys Ellis
23/05/2014	v 0.99	Added user acceptance test.	Alfred Ngako
01/08/2014	v 1.01	Added interface diagrams.	Carlo Machaba
13/08/2014	v 2.01	Started with iteration 3 documentation.	Mathys Ellis

13/08/2014	v 2.4	Added and updated use case diagrams and descriptions. Added and updated post conditions and also domain objects.	Mathys Ellis
02/10/2014	v 2.5	Added and updated use case diagrams and descriptions. Added and updated preconditions conditions.	Mathys Ellis
03/10/2014	v 3.0	Added and updated use case diagrams and descriptions. Added and updated preconditions and postconditions also update data flow diagrams and domain objects. Added application design section and restructured the whole document.	Mathys Ellis
03/10/2014	v 3.01	Added and updated activity and interface diagrams	Alfred Ngako
17/10/2014	v 4.0	Finalised and revised document	Mathys Ellis

# Contents

<b>1 Project Repository</b>	<b>10</b>
<b>2 Document description:</b>	<b>11</b>
2.1 Document purpose: . . . . .	11
2.2 Documentation methodology . . . . .	11
2.3 Document conventions: . . . . .	12
2.4 References: . . . . .	12
<b>3 Functional requirements</b>	<b>13</b>
3.1 Introduction . . . . .	13
3.2 Required functionality . . . . .	13
3.2.1 User gateway . . . . .	13
3.2.2 Application services . . . . .	14
3.2.3 Non-Application services . . . . .	19
3.3 Use case diagrams . . . . .	24
3.4 Use case prioritization . . . . .	39
3.4.1 User gateway . . . . .	39
3.4.2 Application services . . . . .	39
3.4.3 Non-Application services . . . . .	40
3.5 Services contracts for use cases . . . . .	41
3.5.1 Preconditions . . . . .	41
3.5.2 Postconditions . . . . .	49
3.5.3 Request and result data structures . . . . .	58
3.6 Domain Objects . . . . .	59
3.6.1 Overview . . . . .	59
3.6.2 Persistent domain objects . . . . .	60
3.6.3 Non-Persistent domain objects . . . . .	65
3.7 Data flow diagrams . . . . .	67
3.8 Process specifications . . . . .	82
<b>4 Application Design</b>	<b>132</b>
4.1 Overview . . . . .	132
4.2 Client layer . . . . .	133
4.2.1 Summary . . . . .	133
4.2.2 Web page template/Style . . . . .	133
4.3 Client layer . . . . .	135
4.3.1 Summary . . . . .	135
4.4 EJB Layer . . . . .	136
4.4.1 Summary . . . . .	136

4.4.2	Annotations . . . . .	136
4.4.3	Class diagrams of interfaces . . . . .	137
4.5	Persistence Layer . . . . .	140
4.5.1	Summary . . . . .	140
4.5.2	ERD diagram of database . . . . .	141
<b>5</b>	<b>Glossary:</b>	<b>142</b>

## List of Figures

1	Use case diagram of Post-doctoral application management system . . . . .	24
2	Use case diagram of User gateway . . . . .	24
3	Use case diagram of Application services . . . . .	25
4	Use case diagram of Application progress viewer service . . . . .	25
5	Use case diagram of New application services . . . . .	26
6	Use case diagram of Application renewal service . . . . .	26
7	Use case diagram of Referees' report service . . . . .	27
8	Use case diagram of Grant holder application finalisation service . . . . .	27
9	Use case diagram of HOD Recommendation service . . . . .	28
10	Use case diagram of Dean endorsement service . . . . .	28
11	Use case diagram of DRIS approval service . . . . .	29
12	Use case diagram of Progress Report Management Service . . . . .	29
13	Use case diagram of Meeting management service . . . . .	30
14	Use case diagram of Forward and Rewind Service . . . . .	30
15	Use case diagram of Non-Application services . . . . .	31
16	Use case diagram of Google Scholar services . . . . .	31
17	Use case diagram of Location Management Service . . . . .	32
18	Use case diagram of Audit-Trail services . . . . .	32
19	Use case diagram of Notification services . . . . .	33
20	Use case diagram of User account management services . . . . .	33
21	Use case diagram of Report service . . . . .	34
22	Use case diagram of Archival services . . . . .	34
23	Use case diagram of Application import services . . . . .	35
24	Use case diagram of Announcement Services . . . . .	35
25	Use case diagram of Notifier services . . . . .	36
26	Use case diagram of CV management service . . . . .	36
27	Use case diagram of Pre post condition management services . . . . .	37
28	Use case diagram of Neural Network Management Services . . . . .	37
29	Use case diagram of Application success evaluation services . . . . .	38

30	Overview of the data structures and relationships for the persistent domain objects of the system . . . . .	60
31	Overview of the data structures and relationships for the non-persistent domain objects of the system . . . . .	65
32	Data Flow for Application progress viewer services . . . . .	67
33	Data Flow for New Application services . . . . .	67
34	Data Flow for Application Renewal services . . . . .	68
35	Data Flow for Application Success Evaluation services . . . . .	68
36	Data Flow for Archival Service services . . . . .	69
37	Data Flow for Audit Trail services . . . . .	70
38	Data Flow CV Management services . . . . .	70
39	Data Flow for Dean's Endorsement services . . . . .	71
40	Data Flow for DRIS Approval services . . . . .	71
41	Data Flow for Grant Holder Finalisation services . . . . .	72
42	Data Flow for HOD Recommendation services . . . . .	72
43	Data Flow for Meeting Management services . . . . .	73
44	Data Flow for Notification Service services . . . . .	73
45	Data Flow for Nueral Network services . . . . .	74
46	Data Flow for Archival Service services . . . . .	75
47	Data Flow for Pre Post Condition services . . . . .	75
48	Data Flow for Referee Report services . . . . .	76
49	Data Flow for User Account Management services . . . . .	77
50	Data Flow for Location management services . . . . .	78
51	Data Flow for User Gateway. . . . .	79
52	Data Flow for Google scholar services . . . . .	79
53	Data Flow for Forward and rewind services . . . . .	80
54	Data Flow for Reporting Service services . . . . .	80
55	Data Flow for Announcement services . . . . .	81
56	Data Flow for Application import services . . . . .	81
57	Activity diagram of the Login user use case. . . . .	82
58	Activity diagram of the Generate user on-demand account use case. . . . .	82
59	Activity diagram of the Activate user on-demand account use case. . . . .	83
60	Activity diagram of the Authenticate user use case. . . . .	84
61	Activity diagram of the User account retrieval use case. . . . .	85
62	Activity diagram of the Load pending applications use case. . . . .	86
63	Activity diagram of the create anouncement use case. . . . .	87
64	Activity diagram of the load all anouncements use case. . . . .	87
65	Activity diagram of the load all pending anouncements use case. . . . .	87
66	Activity diagram of the remove anouncement use case. . . . .	88
67	Activity diagram of the update anouncement use case. . . . .	88
68	Activity diagram of the Get All Applications use case. . . . .	88

69	Activity diagram of the View application progress use case. . . . .	89
70	Activity diagram of the Create Final Progress Report For Application use case. . . . .	90
71	Activity diagram of the Create renewal Application use case. . . . .	90
72	Activity diagram of the Does Application Have Final Progress Report use case. . . . .	91
73	Activity diagram of the Get Renewable Applications for Fellow use case. . . . .	92
74	Activity diagram of the Create Application Sucess Neural Network use case. . . . .	92
75	Activity diagram of the Get Application Success Rating use case. . . . .	92
76	Activity diagram of the Scale Value use case. . . . .	93
77	Activity diagram of the Create CV use case. . . . .	93
78	Activity diagram of the Update CV use case. . . . .	94
79	Activity diagram of the Deny Application use case. . . . .	94
80	Activity diagram of the Endorse Application use case. . . . .	95
81	Activity diagram of the Approve Funding use case. . . . .	96
82	Activity diagram of the Check Appligibilty use case. . . . .	97
83	Activity diagram of the Deny Funding use case. . . . .	98
84	Activity diagram of the foward application use case. . . . .	98
85	Activity diagram of the load movable applications use case. . . . .	99
86	Activity diagram of the rewind application use case. . . . .	99
87	Activity diagram of the search Google Scholar use case. . . . .	99
88	Activity diagram of the Create Grantholder CV use case. . . . .	100
89	Activity diagram of the Finalise Application use case. . . . .	100
90	Activity diagram of the Ammend Application use case. . . . .	101
91	Activity diagram of the Approve Application use case. . . . .	101
92	Activity diagram of the Deny Application use case. . . . .	102
93	Activity diagram of the create department use case. . . . .	102
94	Activity diagram of the create faculty use case. . . . .	103
95	Activity diagram of the create institution use case. . . . .	103
96	Activity diagram of the get all department for faculty use case. . . . .	104
97	Activity diagram of the get all faculties institution use case. . . . .	104
98	Activity diagram of the get all institutions use case. . . . .	104
99	Activity diagram of the get department use case. . . . .	105
100	Activity diagram of the get faculty use case. . . . .	105
101	Activity diagram of the get institution use case. . . . .	105
102	Activity diagram of the update department use case. . . . .	106
103	Activity diagram of the update faculty use case. . . . .	106
104	Activity diagram of the update institution use case. . . . .	106
105	Activity diagram of the Add Minute Comment use case. . . . .	107
106	Activity diagram of the Create Meeting use case. . . . .	107
107	Activity diagram of the End Meeting use case. . . . .	108
108	Activity diagram of the Get All Active Meetings use case. . . . .	108
109	Activity diagram of the Get All Meetings use case. . . . .	109

110	Activity diagram of the Start Meeting use case. . . . .	109
111	Activity diagram of the Update Meeting use case. . . . .	110
112	Activity diagram of the correctNeuralNetwork use case. . . . .	110
113	Activity diagram of the createNeuralNetwork use case. . . . .	111
114	Activity diagram of the getDefaultNeuralNetwork use case. . . . .	111
115	Activity diagram of the loadAllNeuralNetworks use case. . . . .	112
116	Activity diagram of the makeNeuralNetworkDefaultNetwork use case. . . . .	112
117	Activity diagram of the removeNeuralNetwork use case. . . . .	112
118	Activity diagram of the runNeuralNetwork use case. . . . .	113
119	Activity diagram of the scaleValue use case. . . . .	113
120	Activity diagram of the trainNeuralNetwork use case. . . . .	114
121	Activity diagram of the updateNeuralNetwork use case. . . . .	114
122	Activity diagram of the updateNeuralNetworkSynapses use case. . . . .	115
123	Activity diagram of the Can the fellow open a New Application use case. . .	115
124	Activity diagram of the Create CV use case. . . . .	116
125	Activity diagram of the Create New Application use case. . . . .	116
126	Activity diagram of the Get Open Application use case. . . . .	117
127	Activity diagram of the Link Grant Holder to Application use case. . . . .	117
128	Activity diagram of the Link Referee to Application use case. . . . .	118
129	Activity diagram of the Submit Application use case. . . . .	118
130	Activity diagram of the Get Notifications for Person use case. . . . .	119
131	Activity diagram of the Get Notifications from Person use case. . . . .	119
132	Activity diagram of the Send batch Notifications use case. . . . .	120
133	Activity diagram of the load all pending issues for session use case. . . . .	120
134	Activity diagram of the load all pending issues for user use case. . . . .	121
135	Activity diagram of the send daily reminders use case. . . . .	121
136	Activity diagram of the evaluatePostCondition use case. . . . .	121
137	Activity diagram of the evaluatePreCondition use case. . . . .	122
138	Activity diagram of the findPrePostConditionalMethodByClassName use case.	122
139	Activity diagram of the load loadPrePostConditionalMethods use case. . .	122
140	Activity diagram of the updatePrePostConditionalMethod use case. . . . .	123
141	Activity diagram of the Create Progress Report use case. . . . .	123
142	Activity diagram of the Update Progress Report use case. . . . .	124
143	Activity diagram of the Create Referee Report use case. . . . .	124
144	Activity diagram of the Submit Referee Report use case. . . . .	125
145	Activity diagram of the Count Pending Reports use case. . . . .	125
146	Activity diagram of the Export PDF Report use case. . . . .	126
147	Activity diagram of the Export PDF Report use case. . . . .	126
148	Activity diagram of the Generate System ID use case. . . . .	127
149	Activity diagram of the Get All Security Roles use case. . . . .	127
150	Activity diagram of the Get user by Email use case. . . . .	128

151	Activity diagram of the Get User by System ID use case. . . . .	128
152	Activity diagram of the Remove User use case. . . . .	129
153	Activity diagram of the Update User use case. . . . .	129
154	Activity diagram of the View All Users use case. . . . .	130
155	Activity diagram of the Authenticate User as Owner use case. . . . .	130
156	Activity diagram of the Get Http Session from Session use case. . . . .	131
157	Activity diagram of the Logout use case. . . . .	131
158	System overview . . . . .	132
159	Website Concept Sample 1 . . . . .	133
160	Website Concept Sample 2 . . . . .	134
161	Annotation interfaces . . . . .	136
165	ERD diagram of MySQL database . . . . .	141

# **1 Project Repository**

<https://github.com/mox1990/Project-Postdoc.git>

## **2 Document description:**

### **2.1 Document purpose:**

This functional requirements and application design document serves the purpose of providing a detailed breakdown of the SoftServe's Post-Doctoral management system's expected functionality and how it will be realised in terms of application design. Further it defines the services contracts required by each of the stakeholders from the proposed software system. Thus this document serves as a contract between SoftServe and the client, Mrs Cathy Sandis of the DRIS of the University of Pretoria in terms of project functional requirements and realisation thereof.

### **2.2 Documentation methodology**

The documentation and software development methodology used by the project adhere to the guidelines set out by the scrum agile methodology. Thus this document has undergone and will undergo various iterations that may extend or reduce the contents of the document.

This document was created using the requirement elicitation techniques and requirement definitions as specified by Klaus Pohl's book Requirements Engineering: Fundamentals, Principles, and Techniques [Dr.Pohl, K., 2010]. The requirements, vision and scope were elicited from the following sources:

- Numerous interviews with the client.
- On-line research into UP Post doctoral applications.
- Correspondence with the UP IT department.
- Collecting and analysing various documents such as:
  - The initial project request document
  - Application forms
  - Renewal forms
  - CV templates
  - Approval and recommendation forms

## **2.3 Document conventions:**

- Documentation formulation tool: LaTeX
- Modelling language: UML 2.0, ERD Crow-Foot notation

## **2.4 References:**

- Dr.Phol, K., 2010, *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer, Heidelberg.
- DRIS homepage. [online] Available: <http://web.up.ac.za/default.asp?ipkCategoryID=1630> [Accessed on: 31 March 2014].

## **3 Functional requirements**

### **3.1 Introduction**

This section outlines the functional requirements for SoftServe’s Post-Doctoral management system. The required functionality, domain objects, process specification and use cases related to the functional requirements of the project will be discussed in this sections.

### **3.2 Required functionality**

The following sections will discuss the required functionality of all the major services handled by the system. Namely:

- User gateway
- Application services
- Non-Application services

#### **3.2.1 User gateway**

The user gateway provides the access control services of the system and acts as a centralised gateway through which all users have to go in order to gain access to the system and its services.

- The gateway must provide a user login facility which allows the users to authenticate themselves using their account user name or email address and their account password.
- The gateway must insure that the correct user privileges are loaded before allowing the system to proceed.
- The gateway must insure that the user is allocated a session so that the system can identify the user.
- All internal stakeholders should be able to log in with their PeopleSoft account details once the system is integrated but until such time they should login with the credentials specified at the time of account creation.
- The gateway needs to facilitate user account recovery.

### 3.2.2 Application services

The application services encompasses the entirety of the Post-Doctoral application processes undergone by prospective and research fellows namely new and renewal applications.

The main users of these services will be the prospective and research fellows who wishes to apply for a new Post-Doctoral fellowship or renew their current fellowship or track the progress of any of their applications within the application work-flow. The majority of stakeholders will only make use of particular set of services which are provided under the Application services. Therefore most of the different stakeholders' usage of the system will be focused in this set of services only as these services capture the work-flow of the application processes.

As specified in the vision and scope document under section 7.2 the application process of each application is broken up into stages. These stages run concurrently until they reach the stage where the post-doctoral committee meeting is to take place. In this particular stage the applications are batch reviewed. After which all reviewed eligible applications are once again processed individually. In order to manage the work flow of applications a notification event will trigger after each stage is complete and the applications will be automatically forwarded to the next by means of changing the application status. Only the DRIS will have the power to forward or rewind any application through the stages. It should be noted that if the DRIS moves an application back then all the data of the stages that have been complete will be removed.

These services will capture the work-flow diagram in the Vision and Scope document, section 7.2 - Figure 2 as follows:

- **Application status levels:** Each stage the application goes through requires a different type of task to be preformed or group of tasks to be preformed by the appropriate user. Thus a status level is assigned to each stage in order to manage this work-flow. The status levels are list below in order of first to last. Note \* indicates that the stage is different for new and renewal applications:
  1. **\*Open application** - This application is a newly created prospective or renewal application.
  2. **\*Submitted application** - This application is submitted by the fellow and awaits referral if referral is required else it is automatically moved to Referred status.
  3. **\*Referred application** - This application has a completed list of referral reports from the specified referees or if the application is a renewal that that has been

submitted. Note no referees are required for UP PhD Postdoc or Externally funded applications.

4. **Finalised application** - This application has been finalised by the respective grant holder.
5. **Recommended application** - This application has been recommended by the respective HOD.
6. **Endorsed application** - This application has been endorsed by the respective Dean's office.
7. **Eligible application** - This application has been checked for eligibility by the DRIS and has been found to be eligible.
8. **Funded application** - This application has been approved for funding and is active.
9. **Completed application** - This application has passed its expiry date and has been completed.

Special application status levels are as follow:

1. **Declined application** - This is any application that has been declined by some authority in the process change.
  2. **Terminated application** - This application has been completed or ended by the DRIS before its expiry date.
  3. **Closed application** - This application has been closed by the DRIS before due to the fellow not taking up the application.
- **New application service:** This service deals with the creation and opening of prospective applications for prospective fellows.
    1. A prospective fellow should be able to open only one new application at a time.
    2. A prospective fellow should be able to add their CV in the required format.
    3. A specified grant holder should be able to add their CV in the required format. If they are a NRF A or B rated researcher they are not required to enter their CV.
    4. A owner of a CV should be able to add various qualifications and work experience to their CV.
    5. A owner of a CV should be able to update their CV if it has been created.
    6. Once the application has been finalised the CV will be locked until the application is complete or denied.
    7. A prospective fellow should be able to specify their intended grant holder.
    8. A prospective fellow should be able to specify their intended referees.

9. A application should be made available for stakeholders such as the DRIS, HOD and Dean to deny or approve it at the correct stage in the process.
- **Application renewal service:** This service deals with the renewal currently active applications for research fellows.
    1. A research fellow should be able to open a new renewal application for a particularly current active application.
    2. A research fellow should be able to add their progress report on all the work they have been working on.
    3. A owner of a CV should be able to add various qualifications and work experience to their CV.
    4. A owner of a CV should be able to update their CV if it has been created.
    5. Once the application has been finalised the CV will be locked until the application is complete or denied.
    6. A renewal application should be made available for stakeholders such as the DRIS, HOD and Dean to deny or approve it at the correct stage in the process.
  - **Referees' report service:** This service provides the functionality required by the referees of an application to create referral reports.
    1. A referee should be able to login and create a referral report for the prospective fellow that has identified him/her as referee of there application.
  - **Grant holder application finalisation service:** This service provides the functionality required by the grant holders of an application to review and update the application as they see fit.
    1. A research fellow's grant holder should be able to finalise the renewal application of that research fellow.
    2. A prospective fellow's grant holder should be able to finalise the prospective fellows application who he wishes to supervises.
    3. A grant holder should be able to decline any application.
    4. A grant holder should be able to request amendment for any application and be able to state the reason why.
  - **HOD approval service:** This service provides the functionality required by the HODs to review and create recommendation reports for the applications sent to them.
    1. A HOD of a department should be able to login and approve, decline or ask for amendment of any pending applications.

2. A HOD of a department must be able to create a recommendation report for applications they approve.
- **Dean endorsement service:** This service provides the functionality required by the Deans to review and endorse the applications sent to them.
    1. A member of the dean's office should be able to login and endorse the applications, that they approve of, with a motivation and be able to rank them.
    2. A member of the dean's office should be able to login and deny applications that they disprove of.
  - **DRIS approval service:** This service provides the functionality required by the DRIS to check the eligibility, approve funding and maintain active applications sent to them.
    1. A DRIS member must be able to review pending applications that need to be automatically check applications for eligibility and manually approve or deny them thereof.
    2. A DRIS member must be able to finally approve and create funding reports for them or deny the funding of any eligible applications and also be able to provide motivation and details thereof.
    3. A DRIS member must be able update the funding reports of currently active applications.
    4. A DRIS member must be able to update the progress report of an application.
    5. A DRIS member must be able to terminate or close a currently active application.
  - **Meeting management Service:** This service provides the functionality required by the DRIS to create and manage Post-Doctoral committee meetings and committee members the ability to comment on meetings.
    1. A DRIS member must be able to schedule a meeting for a particular date and add endorsed applications and committee members who need to attend.
    2. A DRIS member must be able to edit any meeting that is still to be held.
    3. A DRIS member must be able to start, close or review meetings.
    4. Committee members attending a meeting must be able to comment on the meeting.
    5. The convener of meeting must also be able to comment on a meeting.
    6. Attendees of a meeting must be able to see the content of all applications attached to the meeting.

- **Application progress viewer service:** This service provides the functionality required by the prospective fellows, research fellows, grant holders and DRIS to check the progress of any application in pipeline and data concerning the application.
  1. A prospective fellow's prospective or a research fellow's renewal application status changes should be made available for their review if they have an application in the system.
  2. The grant holder of an application should be able to view the status changes of the application.
  3. The DRIS must be able to view the status changes for any of the applications in the system.
  4. Owners of an application should be able to review the attachments of any of the applications.
- **Progress report management service:** This service provides the functionality required by the research fellows to check the application for any outstanding progress reports so that they can create them and provide edit functionality of such reports for the DRIS.
  1. A research fellow should be able see which applications of theirs have outstanding reports.
  2. A research fellow should be able to create any outstanding progress report for a particular application.
  3. The number of progress reports for a particular application is equivalent to the number of years that application is active for.
- **Forward and Rewind Service:** This service provides the functionality required by the DRIS to forward and rewind applications through stages in the application workflow.
  1. A DRIS member must be able to forward any application through the various stages in the application pipeline (Submitted to recommend).
  2. A DRIS member must be able to rewind any application through the various stages in the application pipeline (Submitted to declined).
  3. Must allow the DRIS member to create forward and rewind report.
- **Application success evaluation Services:** This service provides the functionality required by the system to evaluate applications using a neural network which is stored on the system in order to determine the success of the application.
  1. A DRIS member, Dean's Office member and HOD must be able to query the neural network for a particular application.

2. A System administrator must be able to create neural network for such an evaluation.
3. A system administrator must be able to train the neural network using already stored applications.

### 3.2.3 Non-Application services

- **Report services:** The report services provides the reporting generation facilities for the DRIS in order to extract valuable information and allow them to provide electronic and hard copy data for review or archiving. The DRIS is the only stakeholder that will make use of this service. Note reports are temporal objects and do not get saved by the system.
  - The DRIS member must be able to access a report generation tool which effectively allows them to:
    1. Open new report.
    2. Select report data from the database.
    3. Generate report.
    4. View report.
  - The DRIS member viewing the report must be allowed to be exported the report to either a pdf or a spreadsheet format.
- **User account management services:** The user account management services provide each user who has an account on the system with the facilities to manage their account and also a facility for the system administrator to manage the accounts on the system.
  1. A prospective fellow will be create a new account if they don't have one.
  2. Any identified user that is not already on the system should be allowed to create a new account.
  3. If integrated with peoplesoft the system should be able to pull all the account information of personnel but until such time the system administrator will have to be allowed create the accounts of all DRIS members, Dean's office members, HODs and post-doctoral committee members.
  4. A user should be able to modify their account details.
  5. A administrator should be able to modify any user account details.
  6. A administrator should be able to remove any user account.
  7. Allow the creation and activation of On-demand users which are users which don't have an account but need an account with other privileges than a prospective fellow account.

- **Notification services:** The system will need to generate automated notifications that are sent internally and to the corresponding email of the recipient. To prevent spamming the system will only allow users with the correct security roles to make use of the service. It is important that this service runs asynchronously after it has been engaged since the service makes use of external systems that may be un-responsive. In case where the external systems fail this service needs to handle it.
  1. The system must be able to create a new notification.
  2. Stakeholders with the correct security roles must be able to create a new notification.
  3. A notification must allow for the specification of recipient.
  4. A notification must allow for the specification of a message.
  5. The service must allow the notification to be sent to both the user account and recipient's email address.
  6. The service must allow a batch of notifications to be sent.
  7. The service must be able to send emails not associated with any internal notification.
  8. The service must be able to see if there are any notifications that have an unsent email counterpart and attempt to resend it.
- **Audit-Trail services:** The Audit-Trail services provide a means for the system administrator or DRIS members to view all the actions that were performed by a user of the system. It is important to note that the audit entries are read-only and can only be inserted by the system itself. The monitored actions are hard wired into the system so to prevent any tampering.
  1. An authorised user must be able to generate a report via the reporting services for the audit log.
  2. The system should be able to insert audit log entries when the monitored actions occur.
- **Archival services:** The archival services of the system will be able to back up the current state of the database to a specified location. Further it should be able to remove old records from the working database that are no longer in use and store them in a location so that they can be retrieved on a on-demand basis.
  1. The system administrator needs to be able to perform an automatic archival of old data.
  2. The system administrator needs to be able to perform a backup of the current database.

- 3. The system should be able to notify the system administrator of any possible archival data.
- 4. The system administrator needs to be able to restore any backed up or archived data.
- **Application import services** The application import services will be used to import existing applications into the system via a GUI wizard. It will not make use of batch import, nor will it import from files.
  - 1. The DRIS member will be able to import applications: to the extend where they include their core application information, fellow, grant holder and funding report as the other information is not required to manage a already approved application.
  - 2. Once the data is finalised it cannot be changed by this service.
- **Location management services:** The system will need to manage the current locations that exist with in the institutions that are registered with the system. A location is the a department, faculty or institution. Departments may be fictional and used as a simple place holder for cross departmental applications. Also if integration with peoplesoft or another system is to occur it will allow this service to synchronise with it. This will be a system service and not be exposed directly to users except in the case of the system administrator wishing to edit or create new locations. Though in order to preserve data integrity locations cannot be removed.
  - 1. The service must allow the creation and update of locations.
  - 2. The service needs to be able to retrieve all institutions.
  - 3. The service needs to be able to retrieve all faculties in the institution.
  - 4. The service needs to be able to retrieve all departments under a faculty of a particular institution.
- **CV management services:** Each grant holder, prospective fellow and research fellow will have a CV. Thus this service will provide the management there of. This will be a system service which will be used indirectly the users of the system.
  - 1. The service must be able to create and update a CV for a particular user.
  - 2. Prospective fellows can only update/create their CVs at the start of a new application.
  - 3. Research fellows can only update/create their CVs at the start of a renewal or new application.
  - 4. Grant holders can only update/create their CVs at the start of a finalisation stage of an application.

- **Notifier services:** This system service is used to provide tracking of outstanding tasks for users and to notify them.
  1. The service must be able to determine any outstanding issues for a particular user.
  2. The service must be able to send out daily reminders of outstanding issues to users.
- **Announcement services:** This service will give system administrators and DRIS members the ability to publish global announcements on the systems that can be viewed by users.
  1. System administrators and DRIS members must be able to create and manage announcements.
  2. Announcements must be duration based.
  3. The service must be able to automatically remove expired announcements from the system.
- **Google scholar services:** This service is used to interface with Google scholar through HTML scrapping. It will be used to retrieve search results from custom queries
  1. The service must be able to communicate with Google scholar and preform searches that is in accordance to fair usage policy.
  2. Must be able to return the results of searches to the system in a system interpretable manner.
- **Pre post condition management services:** This service must be able to load all the methods which have been demarcated to have pre post conditions and provide a means to evaluate them.
  1. A system administrator must be able to edit the pre or post condition of any method in a scripting language.
  2. The system must be able to evaluate pre and post conditions for any demarcated method.
  3. The system may not be allowed to remove any records of such pre and post conditions.
- **Neural network management services:** This service provides the system with a means to manage the neural networks store by the system.
  1. A system administrator must be able to create, edit and remove a Neural network on the system.

2. A system administrator must be able to select the default neural network for the system.
3. The system must be able to run an input vector through a neural network.
4. The system must be able to correct the neural network using back propagation and gradient descent and update the synapses accordingly.

### 3.3 Use case diagrams

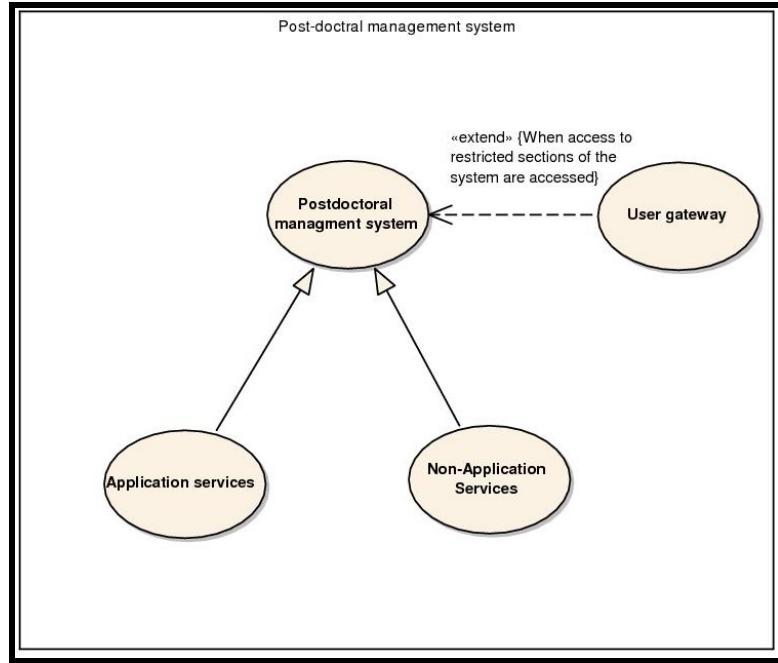


Figure 1: Use case diagram of Post-doctoral application management system

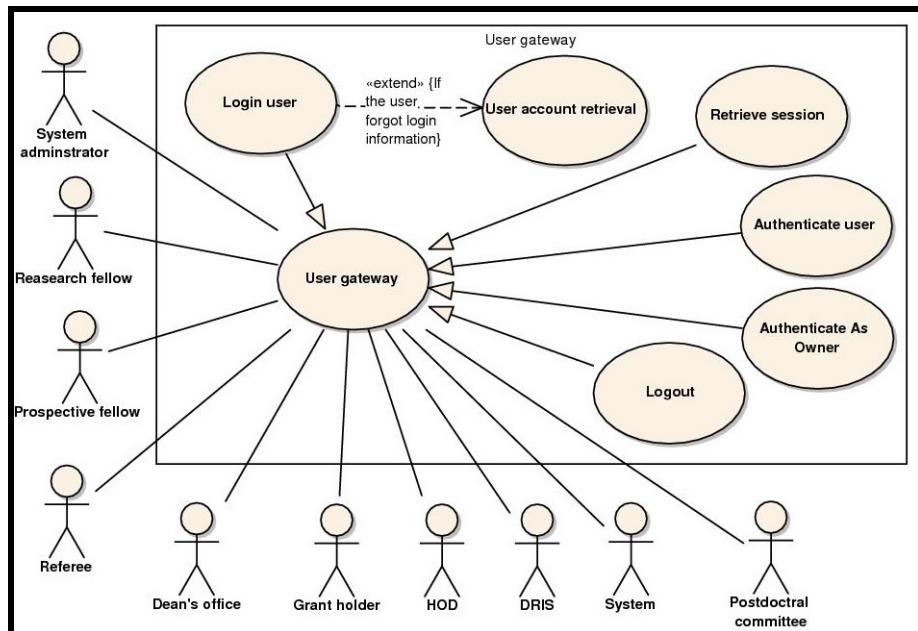


Figure 2: Use case diagram of User gateway

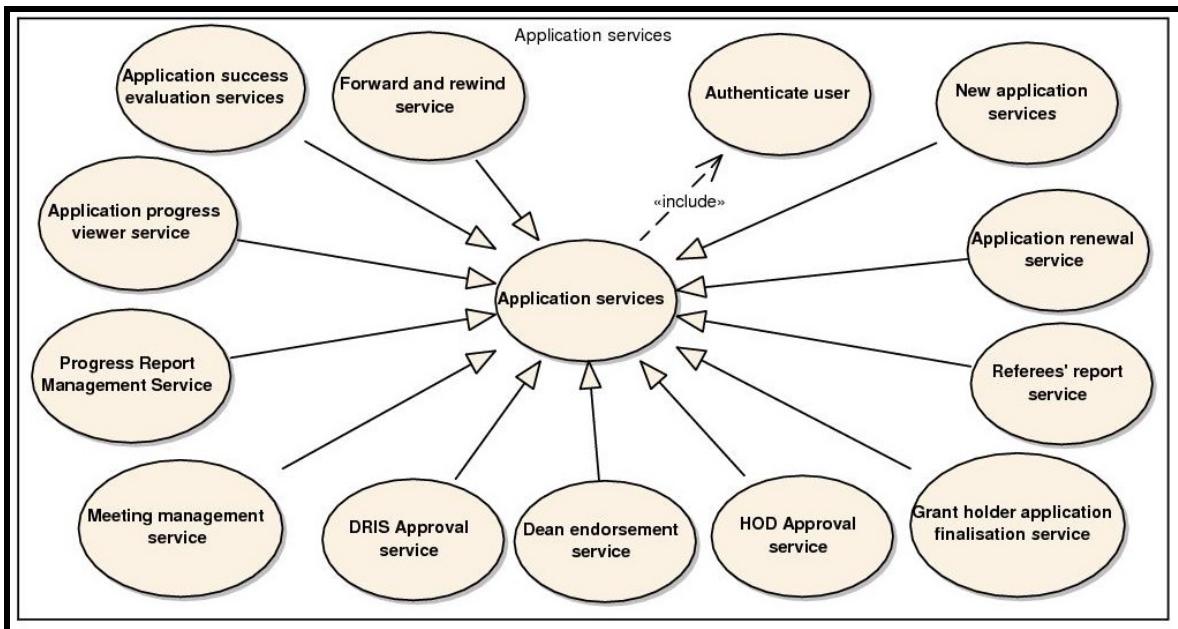


Figure 3: Use case diagram of Application services

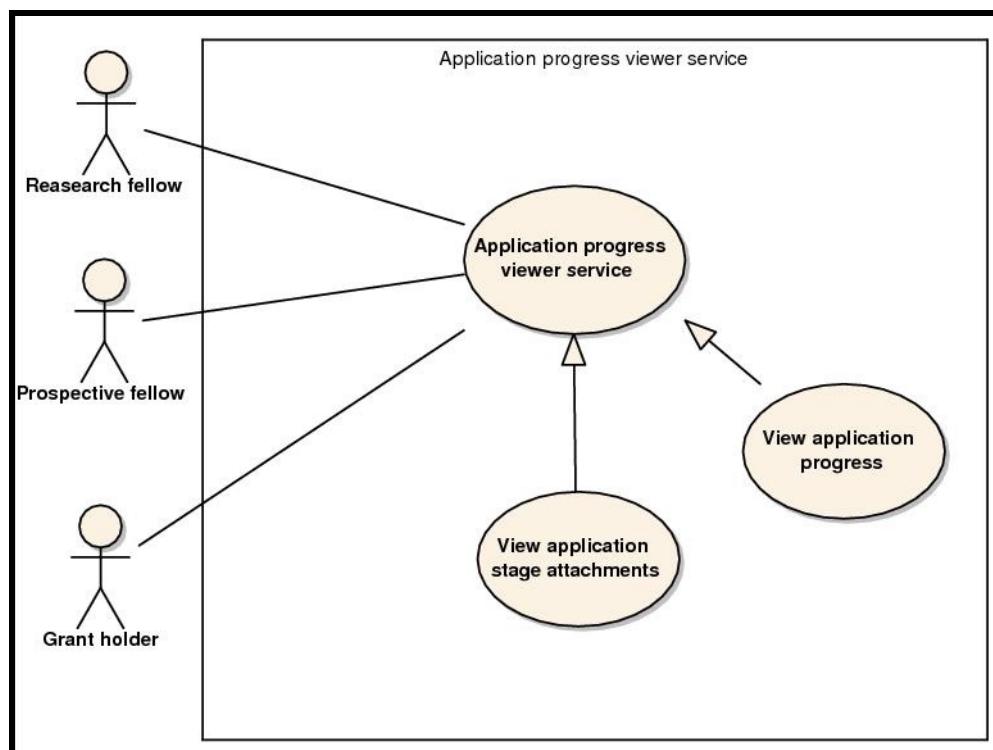


Figure 4: Use case diagram of Application progress viewer service

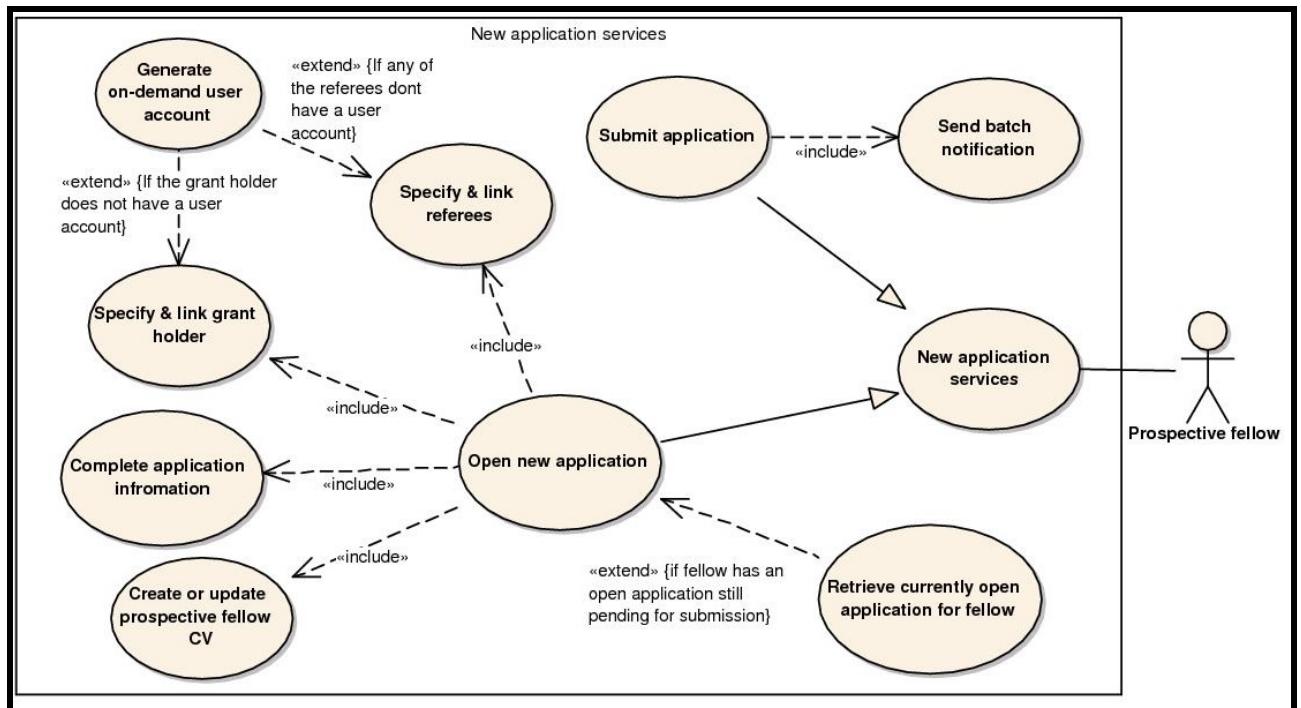


Figure 5: Use case diagram of New application services

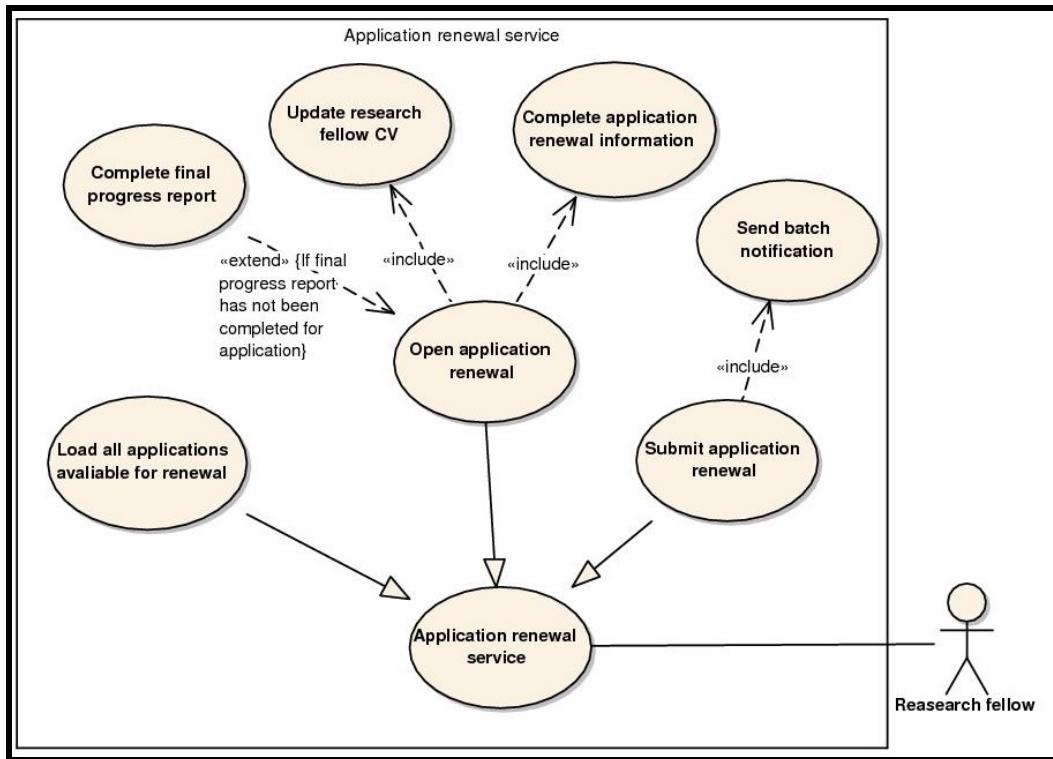


Figure 6: Use case diagram of Application renewal service

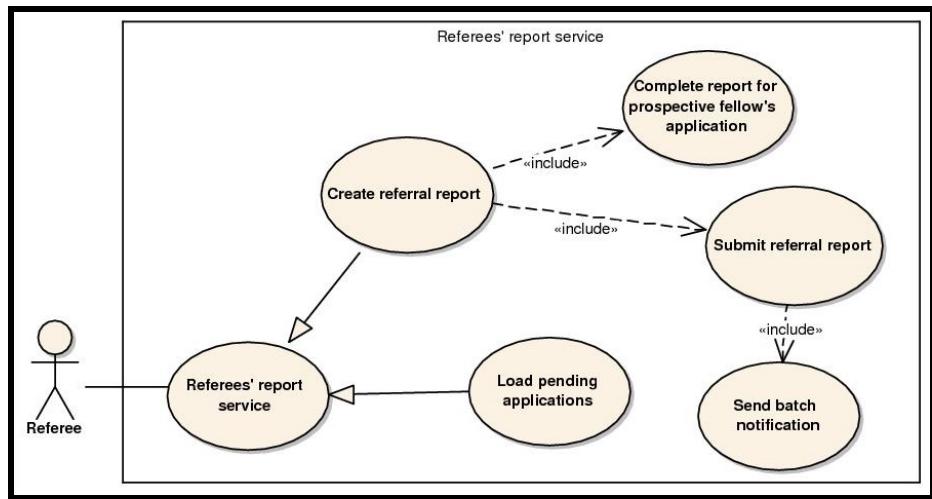


Figure 7: Use case diagram of Referees' report service

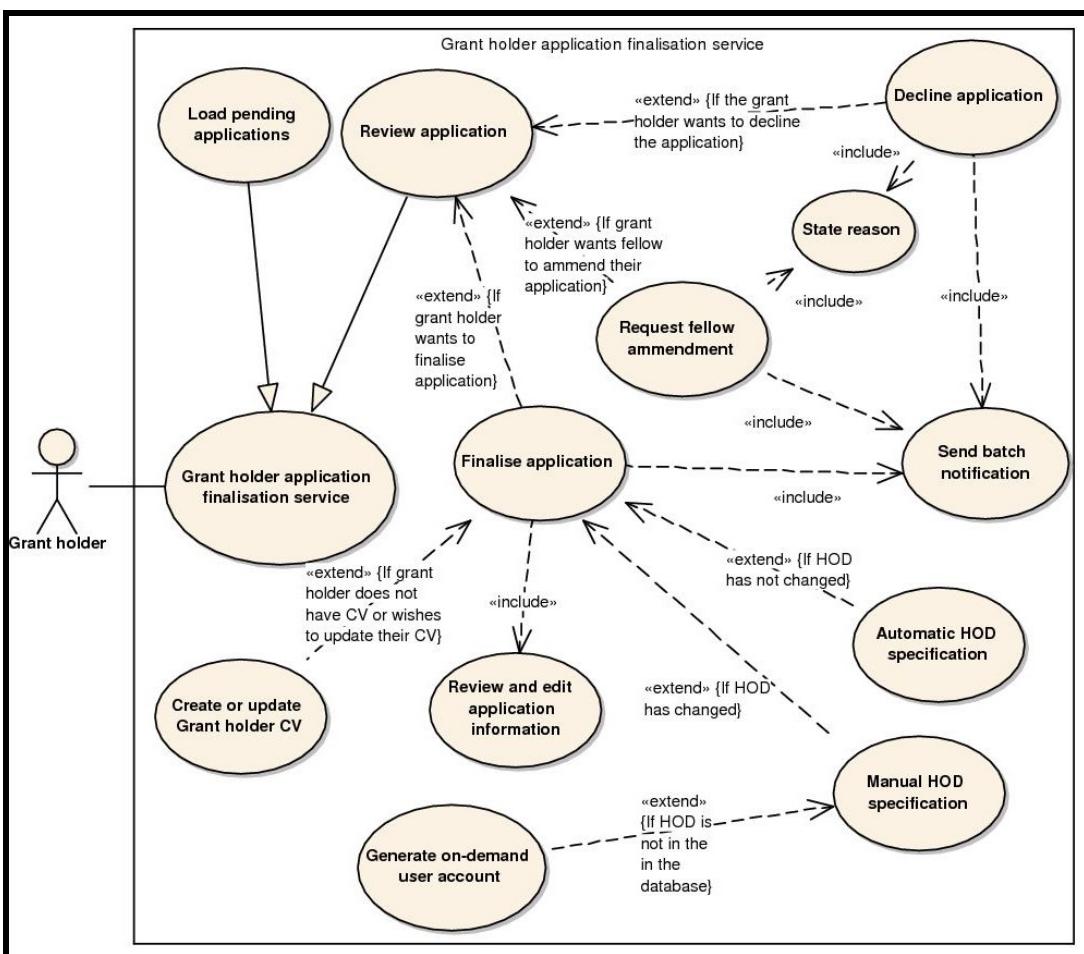


Figure 8: Use case diagram of Grant holder application finalisation service

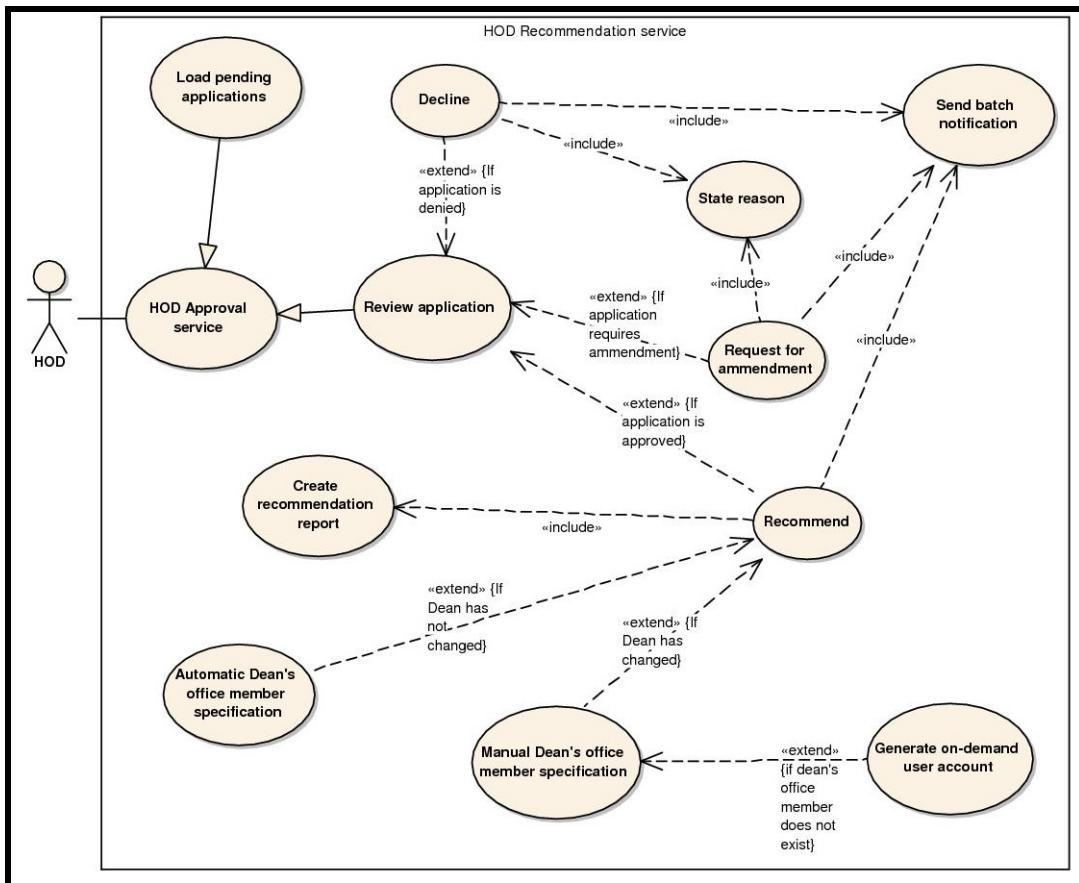


Figure 9: Use case diagram of HOD Recommendation service

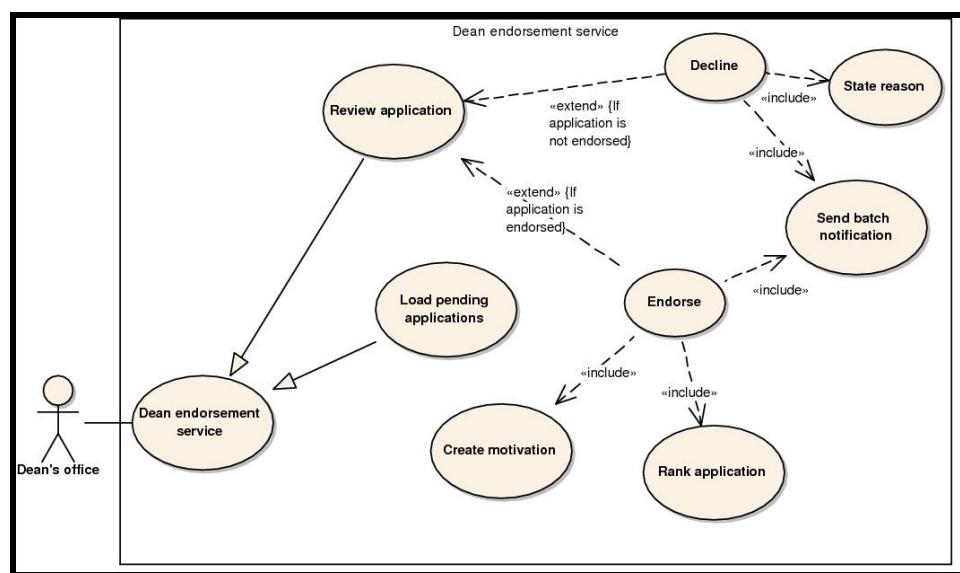


Figure 10: Use case diagram of Dean endorsement service

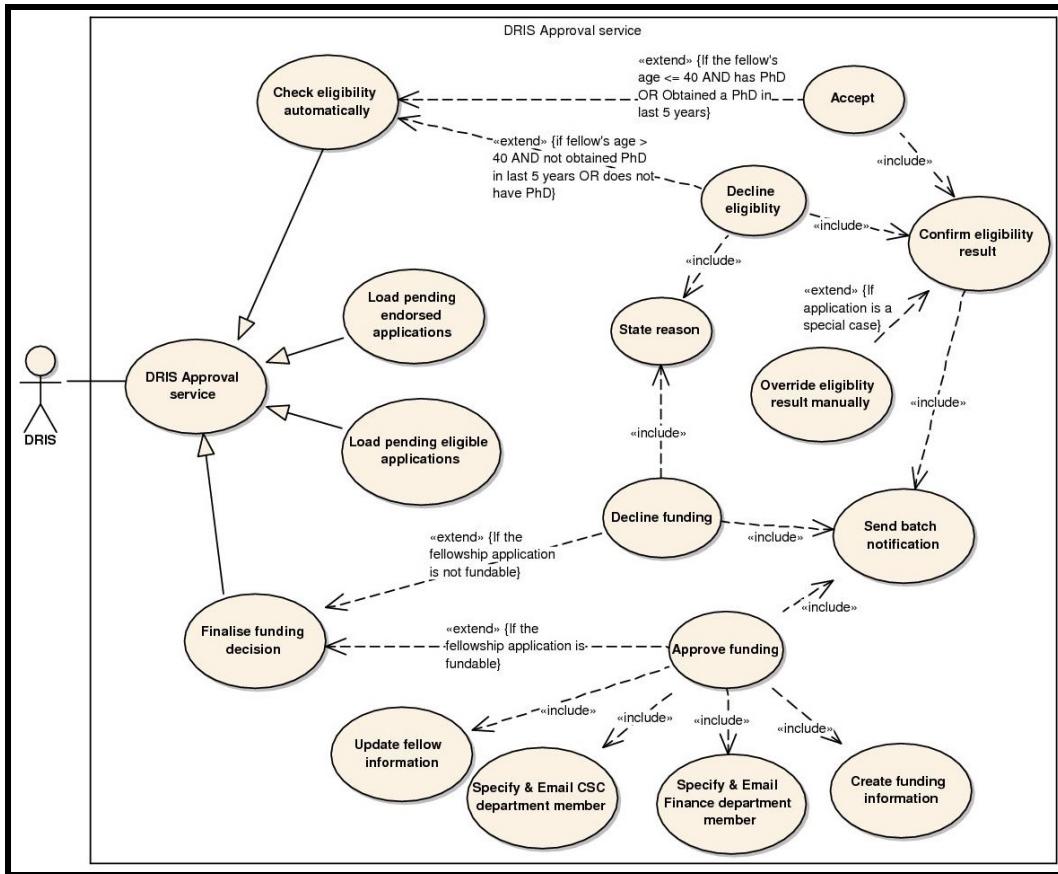


Figure 11: Use case diagram of DRIS approval service

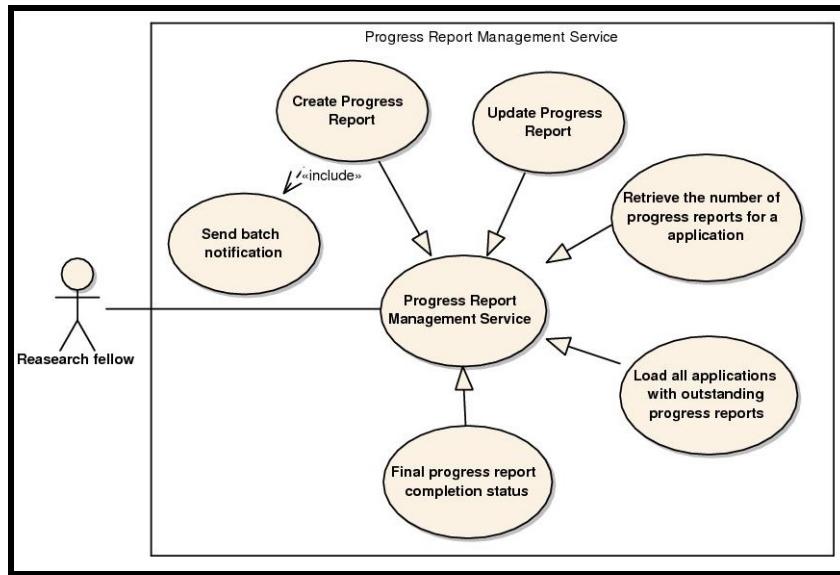


Figure 12: Use case diagram of Progress Report Management Service

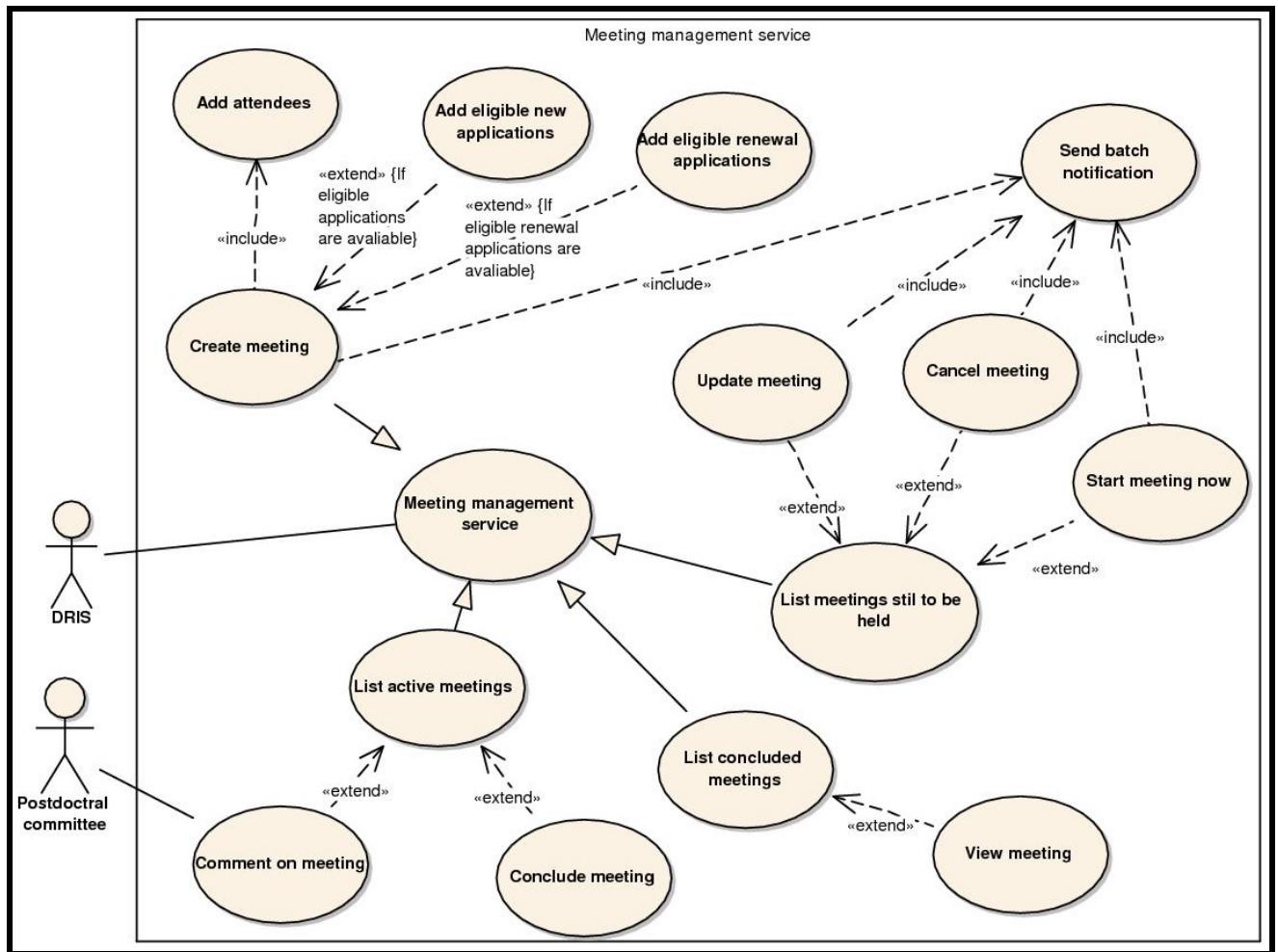


Figure 13: Use case diagram of Meeting management service

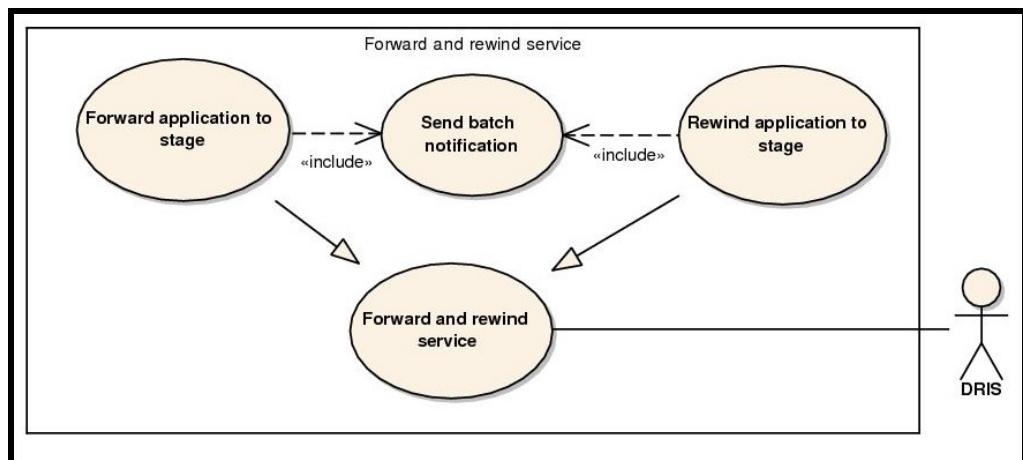


Figure 14: Use case diagram of Forward and Rewind Service

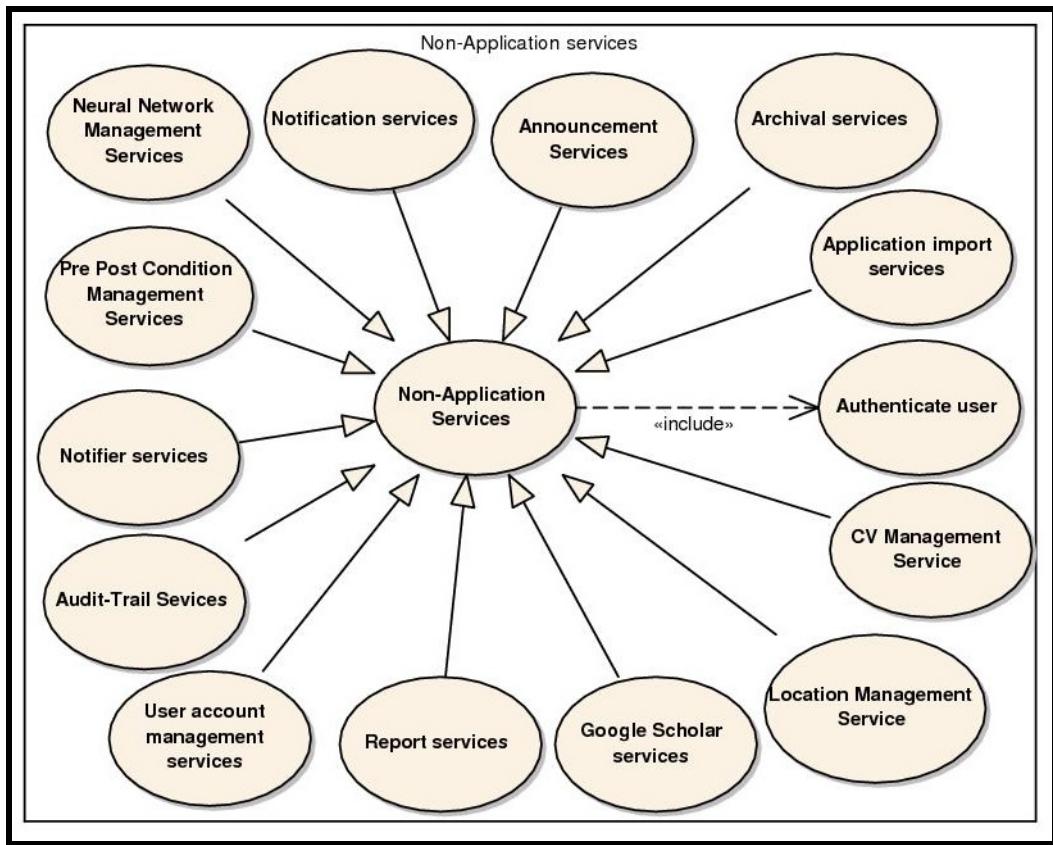


Figure 15: Use case diagram of Non-Application services

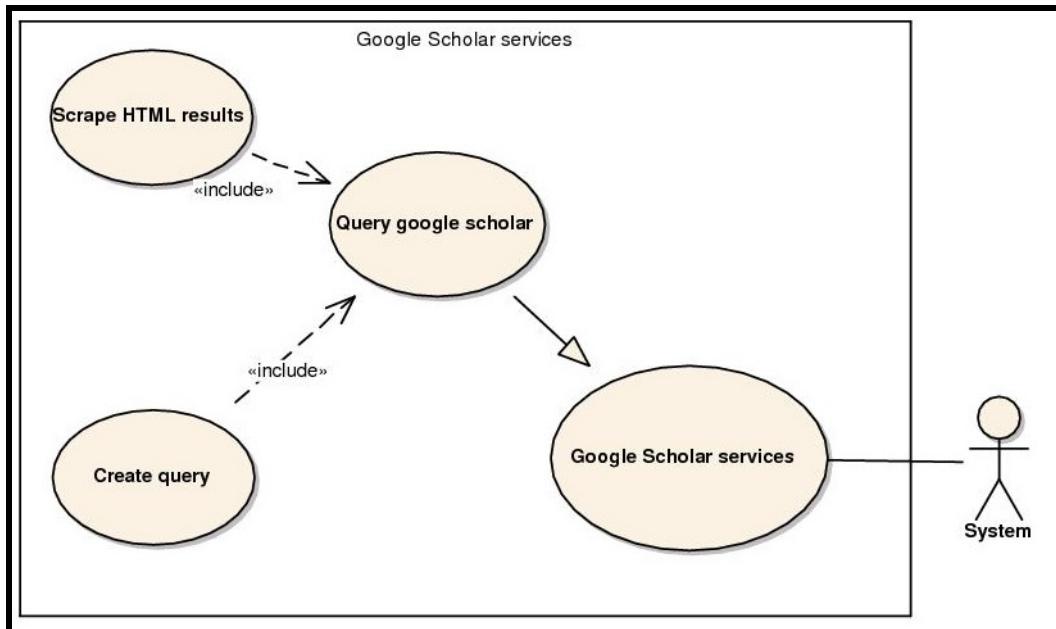


Figure 16: Use case diagram of Google Scholar services

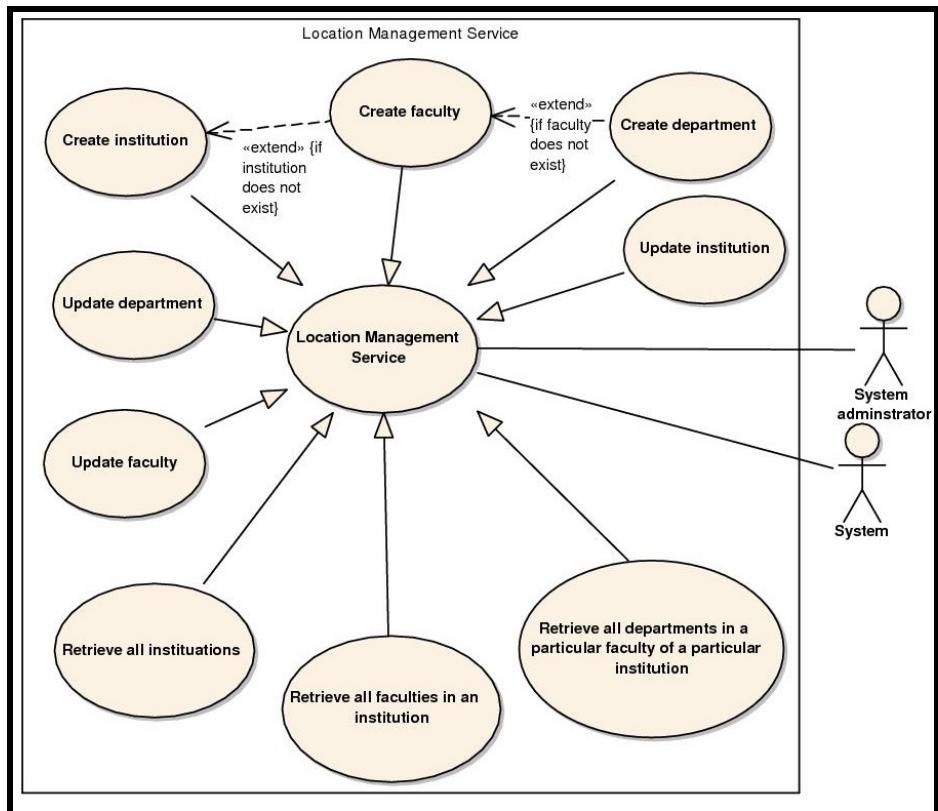


Figure 17: Use case diagram of Location Management Service

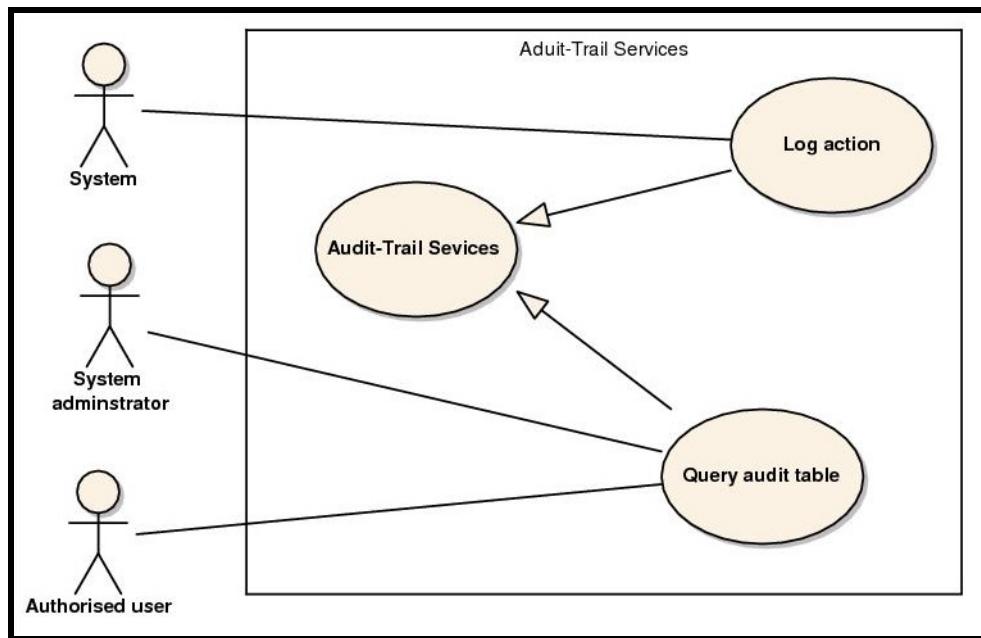


Figure 18: Use case diagram of Audit-Trail services

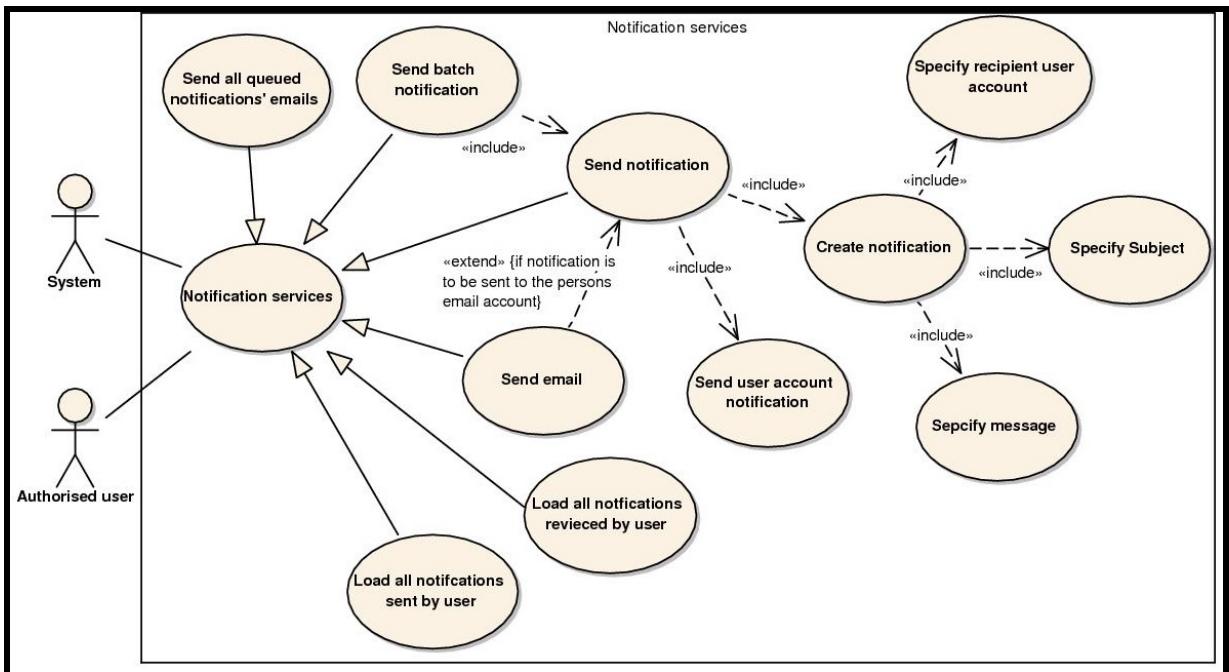


Figure 19: Use case diagram of Notification services

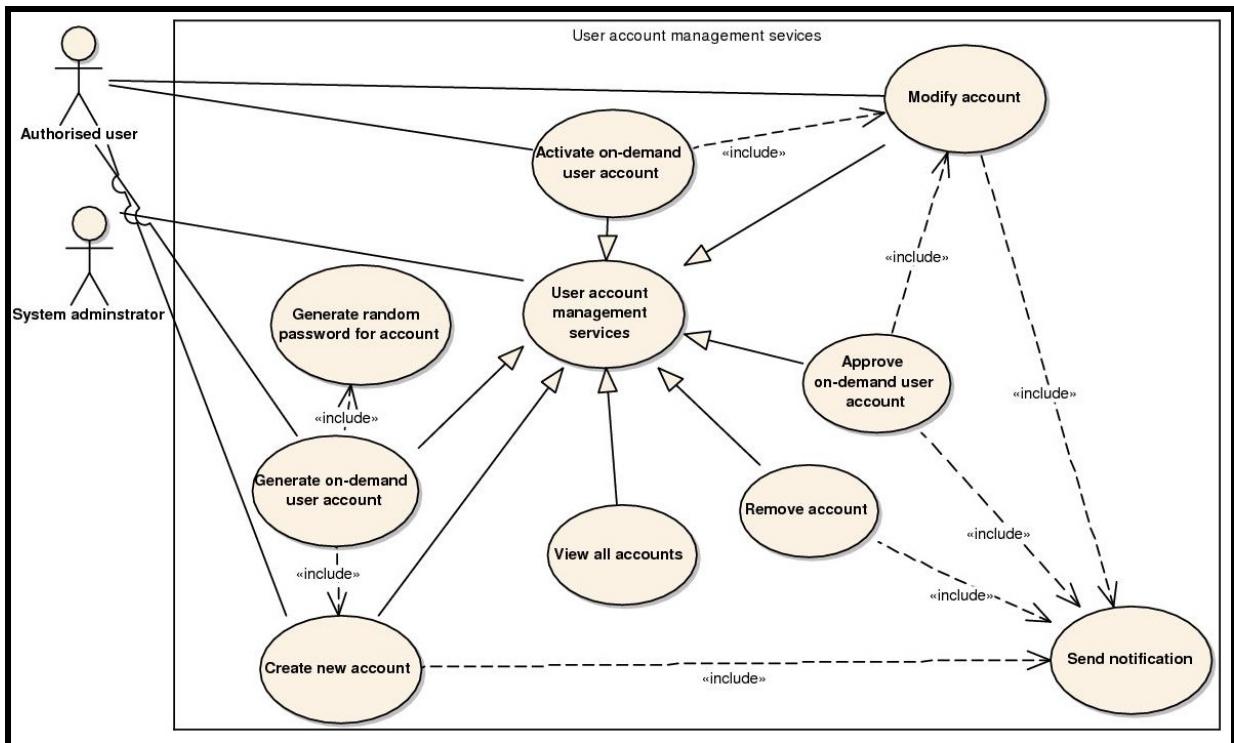


Figure 20: Use case diagram of User account management services

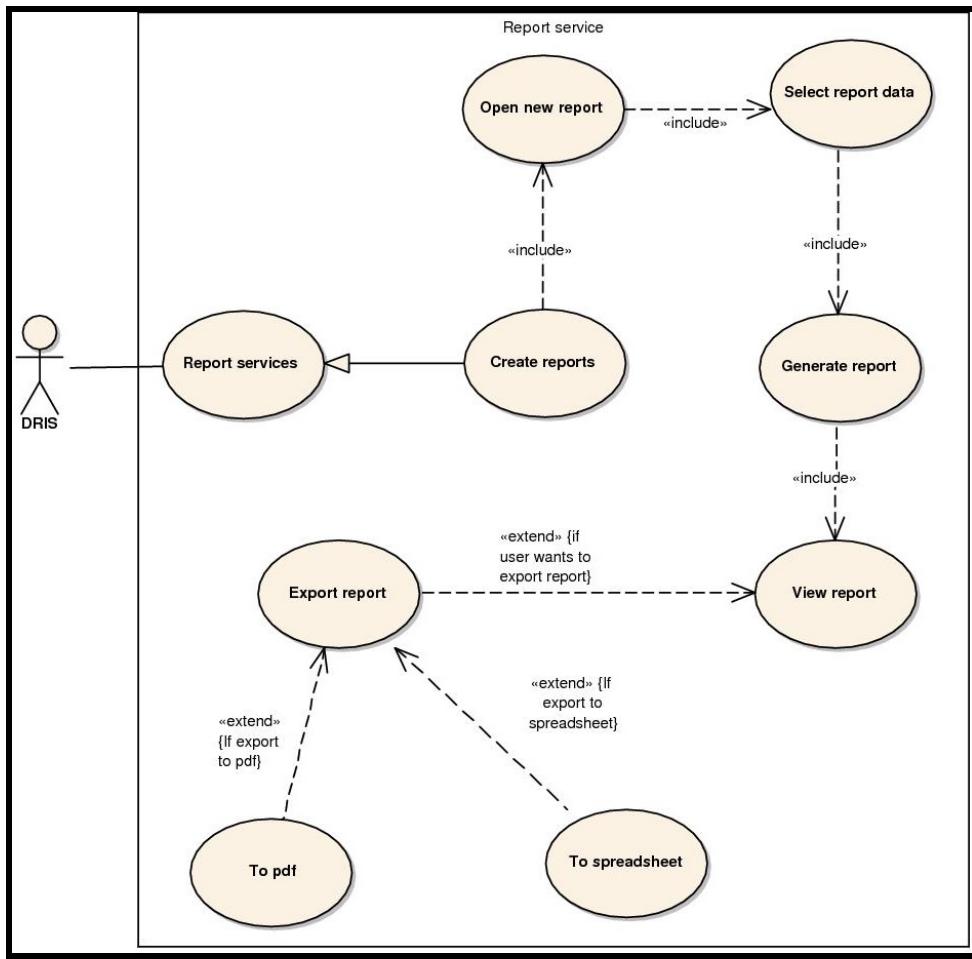


Figure 21: Use case diagram of Report service

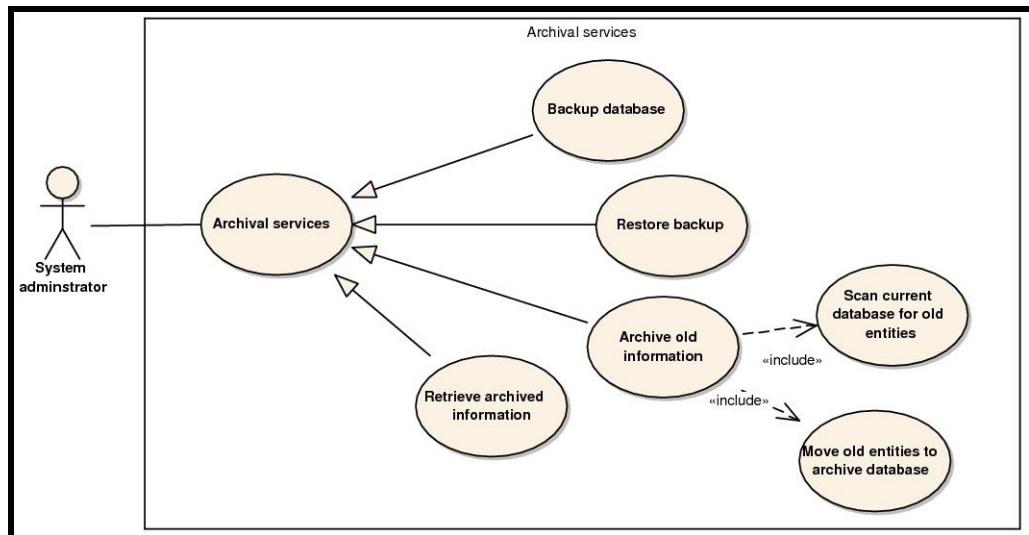


Figure 22: Use case diagram of Archival services

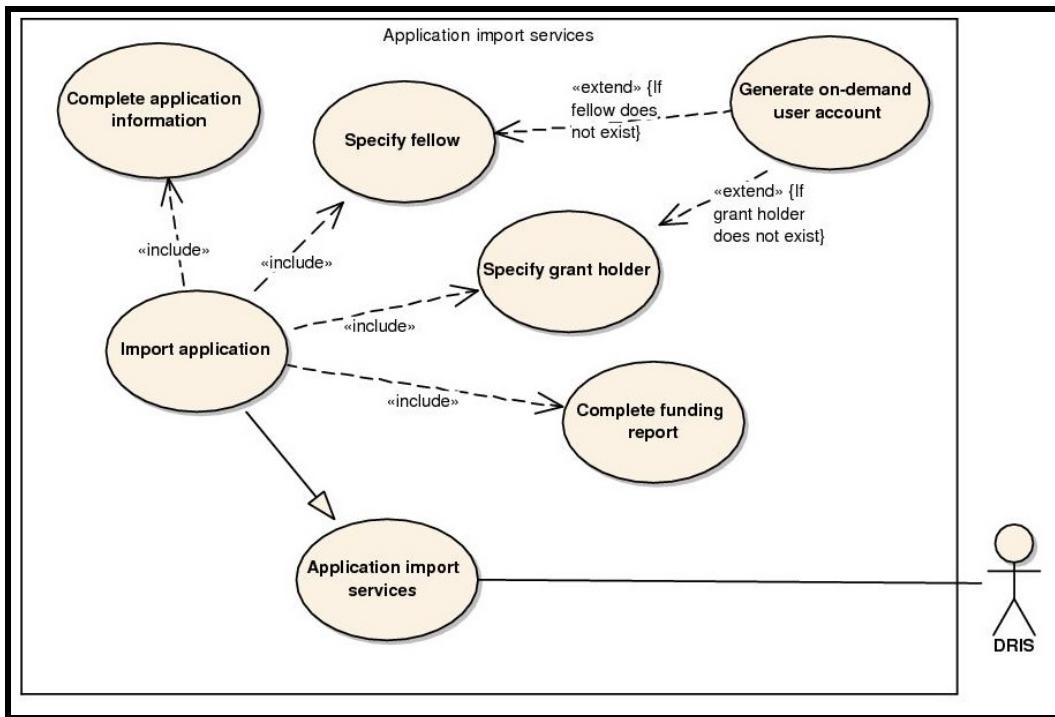


Figure 23: Use case diagram of Application import services

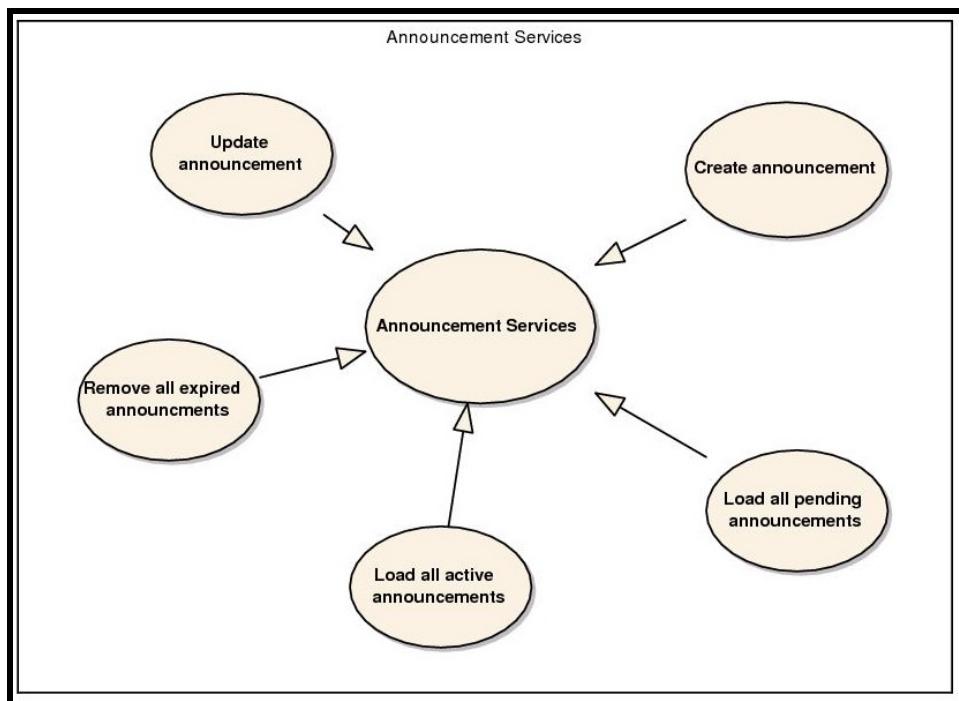


Figure 24: Use case diagram of Announcement Services

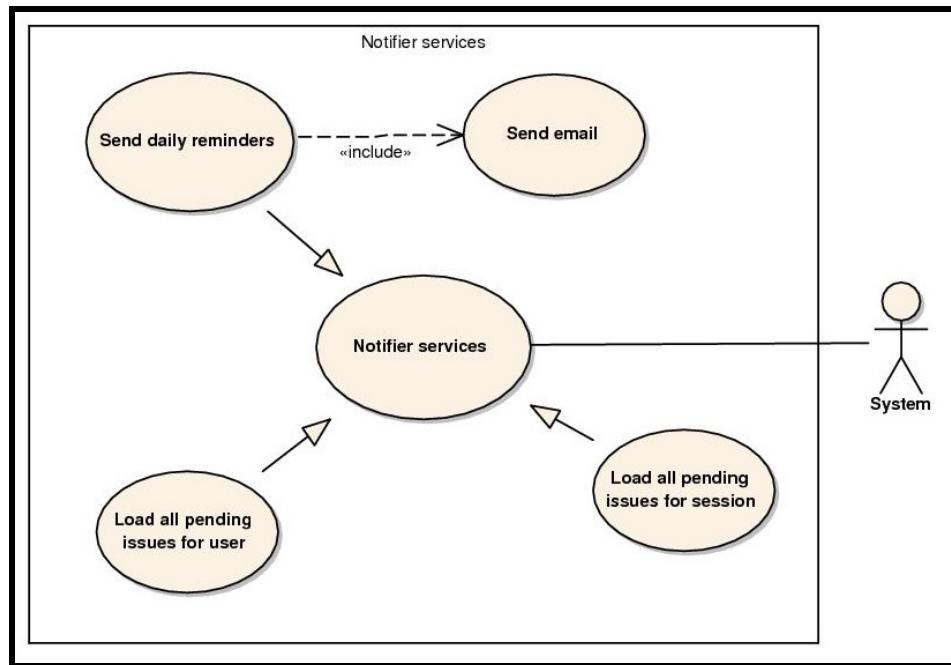


Figure 25: Use case diagram of Notifier services

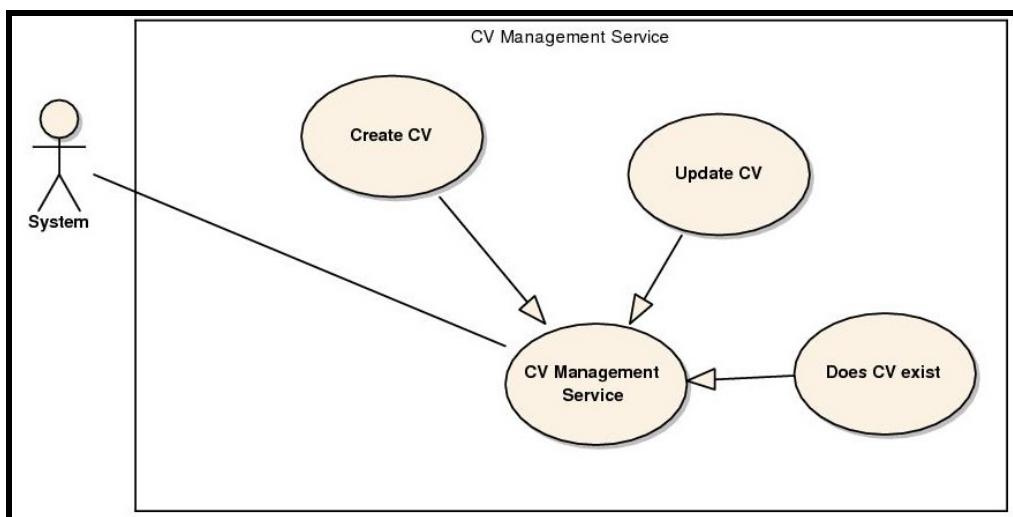


Figure 26: Use case diagram of CV management service

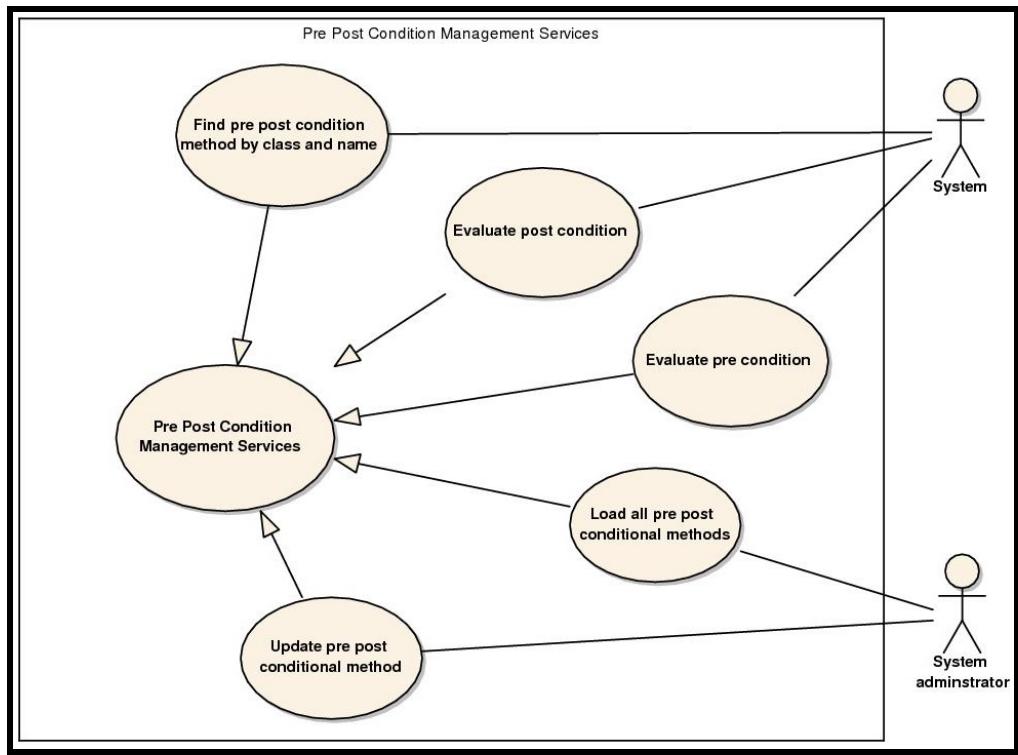


Figure 27: Use case diagram of Pre post condition management services

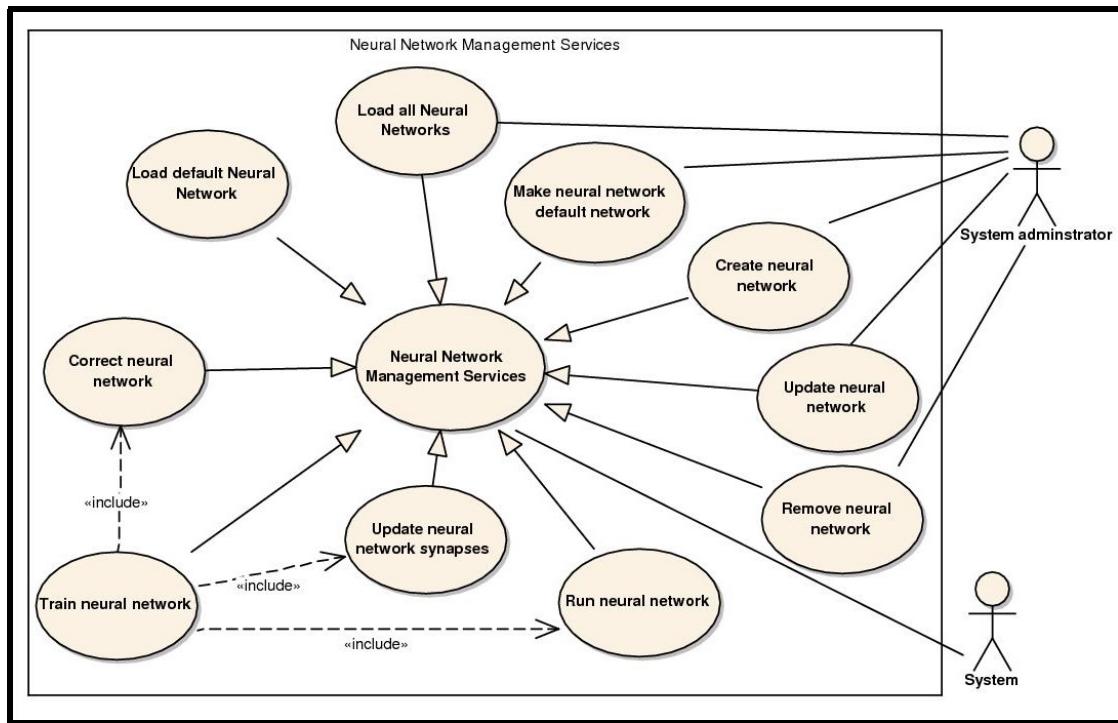


Figure 28: Use case diagram of Neural Network Management Services

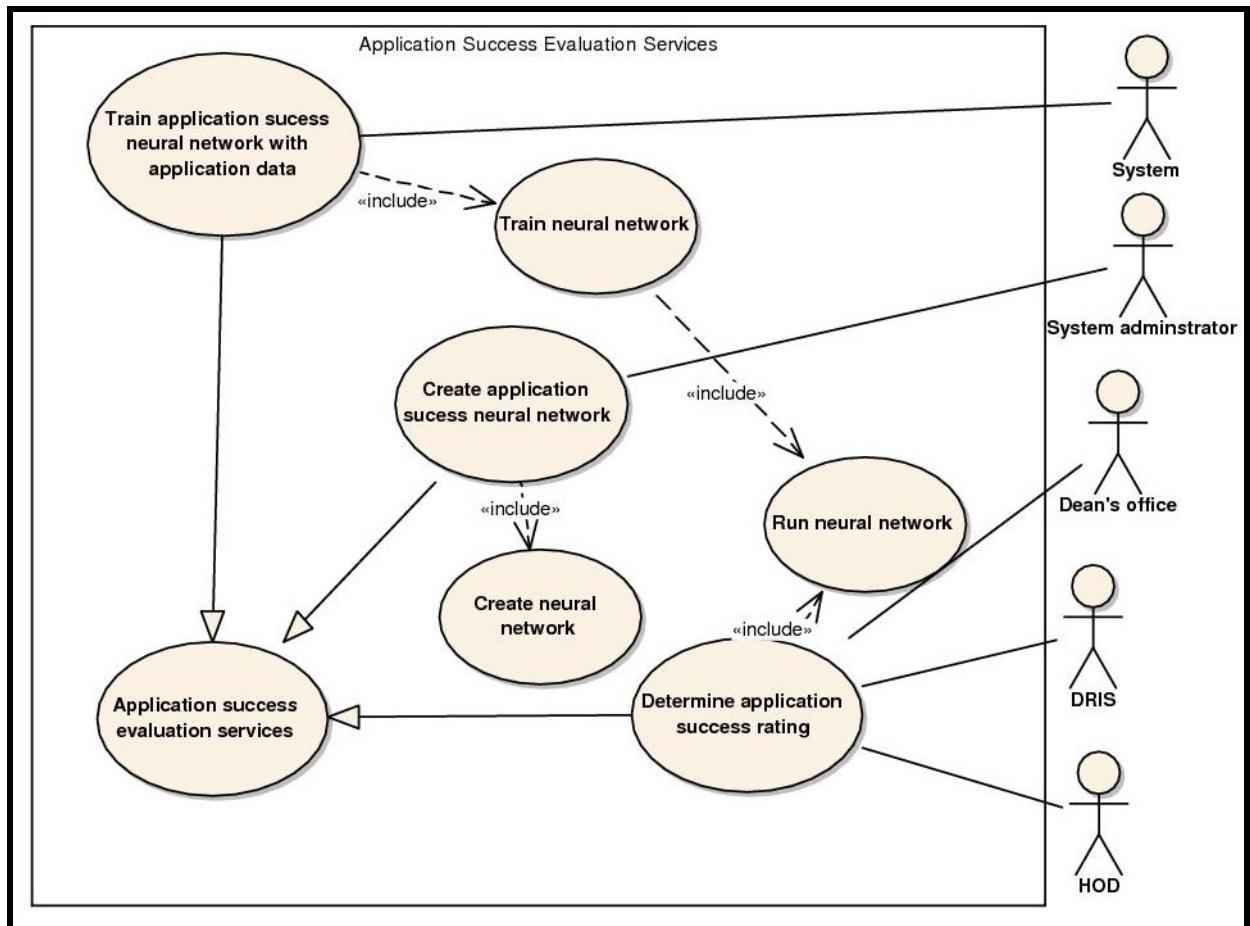


Figure 29: Use case diagram of Application success evaluation services

## **3.4 Use case prioritization**

This section states the ranking in terms of priority of the service use case per use case diagram figure. The priorities are: Critical, Important and Nice to have.

### **3.4.1 User gateway**

- Login user: Critical
- Create prospective fellow user account: Critical
- Generate on-demand user account: Critical
- Activate on-demand user account: Critical
- Authenticate user: Critical
- User account retrieval: Important

### **3.4.2 Application services**

- New application service: Critical
- Application renewal service: Critical
- Referees' report service: Critical
- Grant holder application finalisation service: Critical
- HOD Approval service: Critical
- Dean endorsement service: Critical
- DRIS approval service: Critical
- Progress report management service: Critical
- Meeting management service: Important
- Application progress viewer service: Important
- Fast-forward and rewind service: Important
- Application success evaluation service: Nice to have

### **3.4.3 Non-Application services**

- Notification services: Critical
- User account management services: Critical
- Audit-Trail services: Critical
- CV management service: Critical
- Location management service: Critical
- Report services: Important
- Notifier services: Important
- Archival services: Nice to have
- Application import services: Nice to have
- Google scholar services: Nice to have
- Announcement services: Nice to have
- Neural network management services: Nice to have
- Pre post condition management services: Nice to have

## 3.5 Services contracts for use cases

This section states the preconditions and postconditions of the each use case per use case diagram figure.

### 3.5.1 Preconditions

These are conditions that must be met by the system or user before they are allowed to use the use case.

- **User gateway**

- Login user: Requesting user is a user of the system and not logged in.
- User account retrieval: Requesting user has forgotten their user credentials.
- Authenticate user: User is logged in.
- Authenticate as owner: User is logged in and an object has been selected for ownership testing.
- Logout: The user is logged in.
- Retrieve session: The user is logged in.

- **Application services**

- New application services: The user is a prospective fellow.
- Application renewal service: Renewals are open and the user is a research fellow that is still in possession of active fellowship.
- Referees' report service: If user is a referee.
- Grant holder application finalisation service: The user is a grant holder.
- HOD Approval service: The user is a HOD.
- Dean endorsement service: The user is a member of the dean's office.
- DRIS Approval service: The user is a member of the DRIS.
- Meeting management service: The user is a member of the DRIS or post-doctoral committee.
- Application progress viewer service: The user is a prospective fellow or research fellow or grant holder or DRIS member.
- Forward and rewind service: The user is a member of the DRIS.

- Progress report management service: The user is a research fellow or a member of the DRIS.
- Application success evaluations services: The user is a system administrator or the system.

- **Application progress viewer service**

- View application progress: A application is selected.
- View application stage attachments: A application is selected.

- **New application services**

- Generate on-demand user account: The prospective fellow has identified a referee or grant holder not on the system.
- Submit information: The fellow's cv, application information, grant holder and referee information is complete.
- Open new application: An application is currently open for the fellow or all applications owned by the fellow are completed/closed/terminated.

- **Application renewal service**

- Load all applications available for renewal: There are applications of the research fellow that are available for renewal.
- Open renewal application: The research fellow has a fellowship that is renewable or has a currently open renewal application.
- Complete progress report: The final progress report has not been completed.
- Submit renewal application: If all the required information for the renewal has been completed.

- **Referees' report service**

- Load pending applications: There are submitted applications that need a referral report from the referee.
- Create referral report: The application selected needs a referral report from the referee.
- Submit referral report: If the referral report has been completed.

- **Grant holder application finalisation service**

- Load pending applications: There are referred applications that need to be finalised by the grant holder.

- Create or Update Grant holder CV: The grant holder does not have a CV or has a CV.
- Review application: There are pending applications.
- Finalise application: The grant holder has selected an application that is pending.
- Request fellow amendment: The grant holder has selected an application that is pending.
- Decline application: The grant holder has selected an application that is pending.
- Generate on-demand user account: The grant holder has identified a HOD not on the system.

- **HOD Recommendation service**

- Load pending applications: There are finalised applications that need to be recommended by the HOD.
- Review application: There are pending applications.
- Recommend: The HOD has selected an application that is pending.
- Decline: The HOD has selected an application that is pending.
- Request for amendment: The HOD has selected an application that is pending.
- Create recommendation report: The selected application has been marked for recommendation.
- Generate on-demand user account: The grant holder has identified a Dean's office member not on the system.

- **Dean endorsement service**

- Load pending applications: There are recommend applications that need to be endorsed by the Dean's office.
- Review application: There are pending applications.
- Endorse: The Dean's office member has selected an application that is pending.
- Decline: The Dean's office member has selected an application that is pending.

- **DRIS approval service**

- Load pending endorsed applications: There are endorsed applications that need to be checked for eligibility by the DRIS.
- Load pending eligible applications: There are eligible applications that need to be approved for funding by the DRIS.
- Check eligibility automatically: There are endorsed application available for eligibility check.

- Decline eligibility: The fellow is older than 40 and has not obtained their PhD in the last 5 years or if the prospective fellow does not have a PhD.
- Accept: The fellow is younger than 40 or is 40 and they have a PhD or if they have obtained a PhD in the last 5 years.
- Override eligibility result manually: Application is a special case.
- Confirm eligibility result: A eligibility result for the selected application is determined.
- Finalise funding decision: There are eligible applications available for funding approval.
- Approve funding: The Dean’s office member has selected an application that has a pending funding decision.
- Decline funding: The Dean’s office member has selected an application that has a pending funding decision.
- Fund application: All funding information is complete.

- **Progress report management service**

- Create progress report: The research fellow has an outstanding progress report.
- Update progress report: A valid progress report has been selected.
- Retrieve the number of progress reports for a application: A valid application has been selected.
- Load all applications with outstanding progress reports: There are applications with outstanding progress reports.
- Final progress report completion status: A valid application has been selected.

- **Meeting management service**

- Create meeting: The user is a DRIS member.
- Add eligible new applications: There are eligible new applications available.
- Add eligible renewal applications: There are eligible renewal applications available.
- Add attendees: There are post doctoral committee members available.
- List meetings still to be held: There are meetings still to be held available.
- Update meeting: The meeting is not active nor concluded and the user is a DRIS member.
- Cancel meeting: The meeting is not active nor concluded and the user is a DRIS member.

- Start meeting: The meeting is not active nor concluded and the user is a DRIS member.
- List active meetings: There are active meetings available.
- Comment on meeting: The selected meeting is active and the user is a attendee.
- Conclude meeting: The selected meeting is active and the user is a DRIS member.
- List concluded meetings: There are meetings that have been concluded available.

- **Forward and rewind service**

- Forward application to stage: The selected application has a status between Open and recommended.
- Rewind application to stage: The selected application has a status between Submitted and Declined.

- **Non-Application services**

- Notification services: The user is the system or authorised user.
- User account management services: The user is the system or system administrator or owner of an account.
- Archival services: The user is a system administrator.
- CV Management service: The user has a CV or needs to create a CV.
- Location management service: The user is the system administrator or the system.
- Google scholar services: The user is the system.
- Report services: The user is a DRIS member or system administrator.
- Announcement services: The user is a DRIS member or system administrator.
- Audit-Trail service: The user is a system administrator or the system.
- Notifier service: The user is the system.
- Neural Network management services: The user is the system or a system administrator.
- Pre Post condition management services: The user is the system or a system administrator.

- **Google scholar services**

- Query Google scholar: Google scholar is available and the query is valid.

- **Location management services**

- Create Faculty: The institution exists.

- Create Department: The faculty under which it falls exist.
- Update Institution: A valid institution is selected.
- Update Faculty: A valid faculty is selected.
- Update Department: A valid department is selected.
- Retrieve all Institutions: There are institutions available.
- Retrieve all faculties in an institution: There are faculties available under the institution.
- Retrieve all departments in a particular faculty of a particular institution: There are departments available under the faculty.

- **Audit-Trail services**

- Log action: If requesting user is the system and a User has preformed a critical task.
- Query audit table: If the user is system administrator.

- **Notification services**

- Send all queued notifications' emails: There are unsent emails available and 1 minute has passed since the last check.
- Create notification: The message has a recipient and message.
- Send user account notification: If notification is ready to be sent.
- Send email: If notification is ready to be sent or a standalone email is to be sent.
- Send batch notifications: There is more then one notification request waiting.
- Load all notifications received by user: There are notifications that the user has received available.
- Load all notifications sent by user: There are notifications that the user has sent available.

- **User account management services**

- Create new account: The user is a system administrator or is a prospective fellow.
- Modify account: The user is a system administrator or is the owner of the account.
- Remove account: The user is a system administrator.
- View all accounts: The user is a system administrator.
- Generate on-demand user account: The system requires a new account to be created for a person not the system.

- Approve on-demand user account: The user is a system administrator and there are pending on-demand user accounts available.
- Activate on-demand user account: The user is the owner of the account and the account is a dormant user account.

- **Report service**

- Create report: The user is a system administrator or a DRIS member.
- Open new report: No report is currently open.
- Select report data: A report is open and there is data available for report.
- Generate report: Data has been selected for the report.
- View report: The report has been generated.
- Export report: The user wants to export report and the user is busy viewing the report.
- To Spreadsheet: The user wants to export report to a MS Excel spreadsheet file.
- To pdf: The user wants to export report to a PDF.

- **Archival services**

- Retrieve archived information: The user is a system administrator or the system and there is any archived information available.
- Archive old information: The user is a system administrator or the system.
- Backup database: The user is a system administrator or the system.
- Restore backup: The user is a system administrator or the system and there are backups available.

- **Application import services**

- Import application: The user wants to create a already funded and still active application that already exists but not on the system.
- Generate on-demand user account: If the fellow or grant holder of the application does not exist.

- **Announcement services**

- Create announcement: The user wants to create a new announcement.
- Update announcement: The user wants to update the selected pending/active announcement.
- Remove all expired announcements: There are expired announcements on the system and a 1 Day has passed since the last check.

- Load all active announcements: There are active announcements available.
- Load all pending announcements: There are pending announcements available.

- **Notifier services**

- Send daily reminders: There are issues outstanding for any user and a day has passed since the last check.
- Load all pending issues for user: A valid user is selected and the user is the system and there are outstanding issues for the user.
- Load all pending issues for session: The user is the owner of the session and there are outstanding issues for the user.

- **CV management services**

- Create CV: The system requires to create a CV for some user.
- Update CV: The system requires to update a CV for an existing CV of a particular user who is the owner of the CV.

- **Pre post condition management services**

- Find pre post condition method by class and name: Method must exist.
- Evaluate post condition: Post condition script must exist.
- Evaluate pre condition: Pre condition script must exist.
- Update pre post conditional method: A valid method must be selected.
- Load all pre post conditional methods: There must be at least 1 pre post conditional method available.

- **Neural network management services**

- Load all neural networks: There must be at least 1 neural network available.
- Make neural network default network: A valid neural network must be selected.
- Create a neural network: The system wants to create a neural network.
- Update a neural network: A valid neural network has been selected.
- Remove a neural network: A valid neural network has been selected.
- Correct neural network: A valid neural network has been selected and target vectors have been provided.
- Update neural network synapses: A valid neural network has been selected.
- Run neural network: A valid neural network has been selected and an input vector has been provided.

- **Application success evaluation services**

- Train application success neural network with application data: A valid application success neural network exists and application data exists.
- Create application success neural network: User is an system administrator.
- Determine application success rating: A valid application has been selected.

### 3.5.2 Postconditions

These are conditions that must be met by the system and the data after the use case has been used.

- **User gateway**

- Login user: The user is logged in and has a session.
- User account retrieval: The password has been reset and the user is notified.
- Authenticate user: The user is authenticated with the required security role or has not been authenticated.
- Authenticate as owner: The user is the owner of the object or is not the owner.
- Logout: The user is logged out and has its session invalidated.
- Retrieve session: The user's session is retrieved.

- **Application services**

- New application services: The user can make use of these services.
- Application renewal service: The user can make use of these services.
- Referees' report service: The user can make use of these services.
- Grant holder application finalisation service: The user can make use of these services.
- HOD Approval service: The user can make use of these services.
- Dean endorsement service: The user can make use of these services.
- DRIS Approval service: The user can make use of these services.
- Meeting management service: The user can make use of these services.
- Application progress viewer service: The user can make use of these services.
- Forward and rewind service: The user can make use of these services.
- Progress report management service: The user can make use of these services.
- Application success evaluations services: The user can make use of these services.

- **Application progress viewer service**

- View application progress: The various stages completed thus far by the application is available for viewing.
- View application stage attachments: The various stage attachments made thus far to the application is available for viewing.

- **New application services**

- Open new application: The current open application is opened for editing or a new open application has been created.
- Generate on-demand user account: The identified referees or grant holder has been added to the system and has a pending user account.
- Create or update prospective fellow CV: The prospective fellow's CV has been created or updated.
- Complete application information: The application's information is stored.
- Specify and link grant holder: Grant holder has been linked to the application.
- Specify and link referees: All the referees have been linked to the application.
- Submit information: The application status is changed to submitted if it has referees or referred if it has none or is a renewal.
- Send batch notifications: The fellow and the next stage's members are notified, either referees or grant holder.

- **Application renewal service**

- Load all applications available for renewal: All applications available for renewal has been loaded.
- Open renewal application: The currently open renewal application is opened for editing or a new renewal is created for the selected application.
- Complete progress report: The final progress report is stored.
- Update research fellow CV: The research fellow's CV is updated.
- Complete application renewal information: The renewal application information is stored.
- Submit renewal application: The application status is changed to referred.
- Send batch notifications: The fellow and the grant holder are notified.

- **Referees' report service**

- Load pending applications: All submitted applications with pending reports are loaded.

- Create referral report: The referral report is generated.
- Submit referral report: If the referral report is attached to the application. If it is the last referral report for the application the application status is changed to Referred.
- Send batch notifications: The fellow and the grant holder are notified.

- **Grant holder application finalisation service**

- Load pending applications: All the referred applications that need to be finalised by the grant holder are loaded.
- Create or Update Grant holder CV: The grant holders CV is stored.
- Review application: A pending application has been selected and is visible.
- Finalise application: The application status is changed to Finalised.
- Request fellow amendment: The application is changed to an Open application.
- Decline: The application status is changed to declined and the reason is stored.
- Automatic HOD specification: A HOD has been selected for sending the application to.
- Manual HOD specification: A HOD has been selected for sending the application to.
- Generate on-demand user account: Identified a HOD has been created a pending account on the system.
- Send batch notifications: The fellow, the grant holder and if finalised the specified HOD are notified.

- **HOD Recommendation service**

- Load pending applications: All the finalised applications that need to be recommended by the HOD are loaded.
- Review application: A pending application has been selected and is visible.
- Recommend: The applications status is changed to recommend and forwarded to the next stage.
- Decline: The application status is changed to declined and the reason is stored.
- Request for amendment: The application status has changed to finalised.
- Create recommendation report: A recommendation report is created for the application.
- Automatic Dean's office member specification: A member of the dean's office has been selected for sending the application to.

- Manual Dean’s office member specification: A member of the dean’s office has been selected for sending the application to.
- Generate on-demand user account: The grant holder has identified a Dean’s office member not on the system.
- Send batch notifications: The fellow, the grant holder and if recommended the specified dean are notified.

- **Dean endorsement service**

- Load pending applications: All the recommended applications that need to be endorsed by the Dean’s office have been loaded.
- Review application: A pending application has been selected and is visible.
- Endorse: Application status is changed to endorsed and forwarded to the next stage.
- Decline: The application status is changed to declined and the reason is stored.
- Create motivation: The application’s motivation has been created.
- Send batch notifications: The fellow, the grant holder and if endorsed the DRIS are notified.

- **DRIS approval service**

- Load pending endorsed applications: All the endorsed applications that need to be checked for eligibility by the DRIS are loaded.
- Load pending eligible applications: All the eligible applications that need to be approved for funding by the DRIS are loaded.
- Check eligibility automatically: The application has been checked for eligibility.
- Decline eligibility: The eligibility is set to not eligible.
- Accept: The eligibility is set to eligible.
- Override eligibility result manually: The eligibility result has been inverted.
- Confirm eligibility result: The application status is changed to eligible if the final result is eligible or declined if is not eligible.
- Finalise funding decision: A funding decision has made with regards to the application .
- Approve funding: The applications status has changed to funded.
- Update fellow information: The research fellows information has been updated.
- Specify and email CSC department member: The email to CSC department has been created and waits to sent.

- Specify and email finance department member: The email to Finance department has been created and waits to sent.
- Complete funding information: The funding report and application funding details are completed.
- Decline: The applications status had been changed to declined.
- Send batch notifications: The fellow, the grant holder and if funded the CSC and finance department of UP are notified.

- **Progress report management service**

- Create progress report: The progress report has been created for the application.
- Update progress report: The progress report has been updated.
- Retrieve the number of progress reports for a application: The number of progress reports has been retrieved. (Which is equal to the number of years the application is valid for)
- Load all applications with outstanding progress reports: All applications with outstanding progress reports are loaded.
- Final progress report completion status: A yes is returned if there is a final report or a No is if there isn't.
- Send batch notifications: The fellow, the grant holder and the DRIS are notified.

- **Meeting management service**

- Create meeting: The meeting is scheduled.
- Add eligible new applications: The selected eligible new applications are added.
- Add eligible renewal applications: The selected eligible renewal applications are added.
- Add attendees: The selected post doctoral committee members are added.
- List meetings still to be held: All the meetings still to be held are listed.
- Update meeting: The meeting has been updated.
- Cancel meeting: The meeting has been cancelled and removed.
- Start meeting: The meetings start date is set to now.
- List active meetings: All active meetings are listed.
- Comment on meeting: The comment is created and attached to the meeting.
- Conclude meeting: The meetings end date is set to now.
- List concluded meetings: The meetings that have been concluded are listed.

- Send batch notifications: The DRIS member and the attendees are notified.

- **Forward and rewind service**

- Forward application to stage: The selected application has a status between submitted and endorsed.
- Rewind application to stage: The selected application has a status between Open and endorsed and all the data of stages that were re-winded is removed.

- **Non-Application services**

- Notification services: The user can make use of these services.
- User account management services: The user can make use of these services.
- Archival services: The user can make use of these services.
- CV Management service: The user can make use of these services.
- Location management service: The user can make use of these services.
- Google scholar services: The user can make use of these services.
- Report services: The user can make use of these services.
- Announcement services: The user can make use of these services.
- Audit-Trail service: The user can make use of these services.
- Notifier service: The user can make use of these services.
- Neural Network management services: The user can make use of these services.
- Pre Post condition management services: The user can make use of these services.

- **Google scholar services**

- Query Google scholar: Google scholar has been queried and the results have been returned.
- Create Query: A valid query has been created.
- Scrape HTML results: The retrieved HTML page has been scraped and the results have been extracted.

- **Location management services**

- Create Faculty: The faculty exists.
- Create Department: The department exists.
- Update Institution: The institution is updated.
- Update Faculty: The faculty is updated.

- Update Department: The department is updated.
- Retrieve all Institutions: All the institutions are retrieved.
- Retrieve all faculties in an institution: All the faculties available under the institution are retrieved.
- Retrieve all departments in a particular faculty of a particular institution: All the departments available under the faculty are retrieved.

- **Audit-Trail services**

- Log action: The user and the critical task preformed is logged in the database.
- Query audit table: The entire log available.

- **Notification services**

- Send all queued notifications' emails: All unsent emails are sent if a message cannot be sent its retries are incremented by 1 if its retries are 3 then the email is disabled.
- Create notification: The notification is stored and sent to the user and their email if necessary.
- Send user account notification: The notification is sent to the user.
- Send email: The email is sent to the user's email address, If it succeeds and is not a standalone email the associated notification is marked as sent else unsent..
- Send batch notifications: All notifications are sent.
- Load all notifications received by user: All the notifications that the user has received are loaded.
- Load all notifications sent by user: All the notifications that the user has sent are loaded.

- **User account management services**

- Create new account: The specified user is created on the system.
- Modify account: The specified user account is updated.
- Remove account: The specified user account is marked for removal.
- View all accounts: All the user accounts in the system are loaded.
- Generate on-demand user account: The new user account is generated and has an account status of pending.
- Approve on-demand user account: The user account status is changed to dormant.
- Activate on-demand user account: The user account status is changed to active and the account changes are stored.

- Send notification: The user owner of the user account is notified.

- **Report service**

- Create report: The report is created and after viewing discarded.
- Open new report: A new report is opened.
- Select report data: The report data has been selected, what and how to display.
- Generate report: The report is generated.
- View report: The report is loaded and viewed.
- Export report: The report is exported to the desired format.
- To Spreadsheet: The report is exported to a MS Excel spreadsheet file.
- To pdf: The report is exported to a PDF.

- **Archival services**

- Retrieve archived information: The archive information is retrieved and stored in the working database.
- Archive old information: The mark data in the working database is moved to the archived database.
- Backup database: An exact copy of the current working database is made in the backup database.
- Restore backup: The user is a system administrator or the system and there are backups available.

- **Application import services**

- Import application: The application and the funding data are available in the system.
- Generate on-demand user account: The fellow and/or the grant holder is created on the system.

- **Announcement services**

- Create announcement: A new announcement has been created.
- Update announcement: The announcement has been updated.
- Remove all expired announcements: All expired announcements have been removed from the system.
- Load all active announcements: All the active announcements have been loaded.
- Load all pending announcements: All the pending announcements have been loaded.

- **Notifier services**

- Send daily reminders: Any issues outstanding for any user have been complied into an email which has been sent to the user.
- Load all pending issues for user: All outstanding issues for the user have been loaded.
- Load all pending issues for session: All outstanding issues for the session's user have been loaded.

- **CV management services**

- Create CV: The CV is created for the user.
- Update CV: The CV has been updated for the user.
- Does CV exist: If a CV exists for the user then yes else no.

- **Pre post condition management services**

- Find pre post condition method by class and name: Found pre post condition method.
- Evaluate post condition: Post condition script evaluated.
- Evaluate pre condition: Pre condition script evaluated.
- Update pre post conditional method: Pre post conditional method data updated.
- Load all pre post conditional methods: All pre post conditional methods are loaded.

- **Neural network management services**

- Load all neural networks: All neural network are loaded.
- Make neural network default network: Neural network is set as default.
- Create a neural network: A neural network has been created.
- Update a neural network: Selected neural network has been updated.
- Remove a neural network: Selected neural network has been removed.
- Correct neural network: Selected neural network has been back-propagated.
- Update neural network synapses: Selected neural network's synapses have been updated.
- Run neural network: Input vector has been processed by selected neural network.

- **Application success evaluation services**

- Train application success neural network with application data: Selected application success neural network has been trained with application data.
- Create application success neural network: Application success neural network has been created.
- Determine application success rating: Selected application success neural network has processed application and holds a success value.

### **3.5.3 Request and result data structures**

The system will be following a object oriented approach due it being the paradigm of the Java programming language. Therefore the input and output structure will mainly be in the form of objects. Also the objects that will be produced and used inside the system will adhere to the domain objects specification found the following section. The flow of these structures are also provided below in form of data flow diagrams in the following sections

## **3.6 Domain Objects**

### **3.6.1 Overview**

This section discusses the domain objects that are required by the system in technological neutral manner. Within the system there are two types of domain objects classes: Persistent and non-persistent domain objects.

### 3.6.2 Persistent domain objects

These domain objects represent those domain objects that are stored in some form of persistent data store.

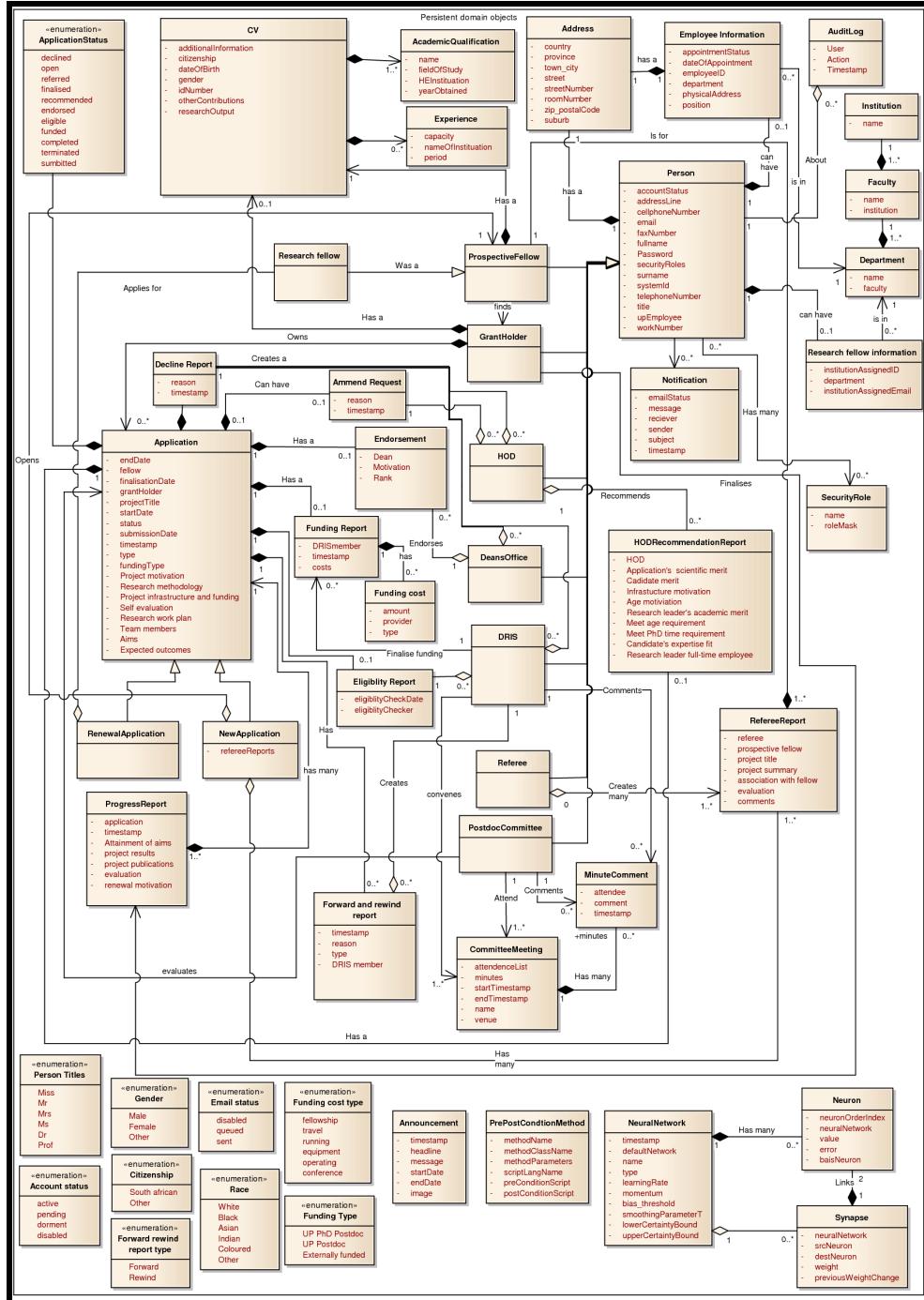


Figure 30: Overview of the data structures and relationships for the persistent domain objects of the system

- **Person** This object represents the stakeholders that will make use of the system. All stakeholders will have accounts which they will use to log on to the system, using a unique user id and a predefined or user specified password. The unique user id can either be a Peoplesoft Emplid number or a email address. The person has an associated **Location** and **SecurityRole(s)**
- **Department** This object represents the department under which a **Person** falls in the **Faculty**. This is used to if they are a **employee** or **Research fellow** at the University of Pretoria. This object will no longer be needed if the system is integrated with peoplesoft as it would cause redundancy.
- **Faculty** This object represents the faculty that may exist under an institution. It is used to map the location of a **Person**.
- **Institution** This object represents the institutions that may exist in the system. It is used to map the location of a **Person**. This allows for generic support if the software solution had to be expanded to other universities.
- **EmployeeInformation** This object represents the information regarding the employment of a **Person** at the University. It also contains the **Department** under which the **Person** falls. Also it contains the physical **Address** of the employee's office. This is made generic for software expansion support.
- **ResearchFellowInformation** This object represents the CSC's or institution equivalents assigned emplid and the institution's assigned email address for an appointed **ResearchFellow**. This allows for possible future integration and better user support. Also it stores the **Department** in the institution to which the fellow was assigned to by the **DRIS**.
- **Address** This object represents a physical address. Thus it is used to expand and provide location details for a particular physical object or person.
- **SecurityRole** This object represents a particular security role of a **Person**. A **Person** may have many different security roles. The security role can be used as a subtype discriminator for relational database architectures.
- **DRIS** This object represents members of Department of Research and Innovation Support who administers the process.
- **FundingReport** This object represents an **Application**'s funding costs as decide per the **DRIS**. It comprises of multiple **FundingCosts** that provides the various funding inputs from various parties for the particular fund-able costs namely fellowship, travel, running, equipment, operating and conference.

- **FundingCost** This object represents a particular funding input cost for a **Application** from a particular funder.
- **DeclineReport** This object represents the report associated with an **Application** that has been declined by some authority. It contains the creation timestamp, **Person** responsible and the reason. Only one such report can exist per **Application**.
- **AmendRequest** This object represents a Amend request for a **Application** that is sent by some **HOD**. It contains the creation timestamp, **HOD** responsible and the reason. This is used to allow fellow to understand what is wrong with their **Application** so that they can correct it. There can be multiple such requests associated with an **Application**.
- **EligibilityReport** This object represents a report associated with an **Application** that has been checked for eligibility. It contains the check timestamp and **DRIS** member responsible. It is primary used for auditing. Only one such report can exist per **Application**.
- **ForwardAndRewindReport** This object represents a report associated with an **Application** that has been either forwarded or re-winded through a particular set of application stages. This is used to track such actions and the reasons for them. There can be multiple such reports associated with an **Application**.
- **ProspectiveFellow** This inherited object represents a prospective fellow who is a holder of a PhD obtained in the last five years (or nearing completion of a PhD) or is 40 years or younger and has a PhD. The prospective fellow can open a **NewApplication**.
- **ResearchFellow** This inherited object represents a research fellow who is a currently a researcher at the University of Pretoria. This object was initially a **ProspectiveFellow**. The research fellow can apply for a **RenewalApplication** if their application falls in their renewal time frame.
- **GrantHolder** This inherited object represents a grant holder who can be a rated researcher by the NRF or not. The system should not require the CV's of A and B rated researchers to be added to the system. The reason for this is that the CV's of such researchers can be easily obtained from the NRF and tend to be very long. A grant holder is the supervisor for a or many **ProspectiveFellow(s)** or **ResearchFellow(s)** and owns the **Application** of the **ProspectiveFellow(s)** or **ResearchFellow(s)**.
- **HOD** This inherited object represents a HOD of a particular department. The HOD creates the recommendation reports for **Application(s)** they consider to meet their requirements.

- **HODRecommendationReport** This inherited object represents a recommendation report highlighting the reasons to why the **Application** of a **ProspectiveFellow** or **ResearchFellow** is needed by the department.
- **Deans Office** The Dean's office object represents the relevant faculty's Dean and Deputy Dean. The Dean's Office creates the **Endorsement** for any the **Application** that is approved by them.
- **Endorsement** This object represents the endorsement of an **Application** of a **ProspectiveFellow** or **ResearchFellow** and contains the rank in comparison to other pending **Application(s)** with a **Endorsement**.
- **Referee** This inherited object represents the referees of any **ProspectiveFellow** and is responsible for creating **RefereeReport** regarding the **ProspectiveFellow**.
- **RefereeReport** This object represents the referral report from an identified referee of a **ProspectiveFellow**.
- **PostDocCommittee** This inherited object represents the individual members of the post-doctoral committee who approves all available **Applications** during committee meetings and records the **Minutes** of the meeting.
- **CommitteeMeeting** This object represents a meeting of the **PostDocCommittee** convened by the **DRIS** that will be review the **Applications** and will evaluate each. This object contains the attendance list, date and time convened and the **MinuteComments** of the meeting.
- **MinuteComment** This object represents a comment made by a **PostDocCommittee** member during a **CommitteeMeeting**.
- **Application** This object represents an applications and will contain the information of **ProspectiveFellow** or **ResearchFellow** and **GrantHolder** who owns it. The object holds the status of the application and any project information. As well as the **HODRecommendationReport** of a **HOD** and **Endorsement** from a **DeansOffice**. It further holds the **EligibilityReport** and **FundingReport** from the **DRIS**. If the application is declined the **DeclineReport** is associated with the application. The application object can also hold multiple **AmmendRequests** if any such requests are created. All the **ProgressReports** created by the **ResearchFellow** are also contained in the application. Lastly if the application forwarded or re-winded then the associated **Forward** **rewind reports** are also contained by the application
- **NewApplication** This inherited object represents new application for a **ProspectiveFellow** who is currently not a fellow in the system. Also it holds any **RefereeReport(s)** that has been created for the application.

- **RenewalApplication** This inherited object represents renewal application for a **ProspectiveFellow** who is a fellow in the system. Also it holds the **ProgressReport** that has been created for the application.
- **ProgressReport** This object represents the periodical report on the research that the **ResearchFellow** has done for the particular period in the duration of their fellowship. These reports are completed on an annual bases.
- **CV** This object represents a CV and contains all the information such as personal details, **AcademicQualification(s)**, **Experience** regarding a **GrantHolder** or **ProspectiveFellow** in the system.
- **AcademicQualification** This object represents a academic qualification and the information regarding it such as the qualification name, field, where it was obtained and when it was obtained.
- **Experience** This object represents a work experience and the information regarding it such as the capacity of the work, where this work was done and when it was done.
- **Notification** This object represents a email or internal message sent by a user to a user via the system. The system itself may also seen as a user with regards to the sender of such notification. It consists of the actual message, subject, timestamp and email status.
- **AuditLog** This object represents a audit log that stores all the actions of all users within the system. Each entry records the action, who committed the action as well as at what time the action was committed.
- **Announcement** This object represents an announcement which has been created on the system for logged in users to view. It also contains a image alongside text.
- **PrePostConditionMethod** This object represents a method which has been demarcated as one having pre or post conditions and contains the scripts that need to executed for each.
- **NeuralNetwork** This object represents a neural network and all the properties it requires for back propagation and initialisation.
- **Synapse** This object represents a synapse in a **NeuralNetwork** which contains a weight and previous weight change. It connects two **Neurons** together. The weights are used to manipulate the values of the input **Neurons** during the feed forward phase.
- **Neuron** This object represents a neuron in a **NeuralNetwork** which contains a value and is used during the feed forward phase as input to the next layer.

### 3.6.3 Non-Persistent domain objects

These domain objects represent those domain objects that are not stored in some form of persistent data store and exist with in the runtime of the software solution.

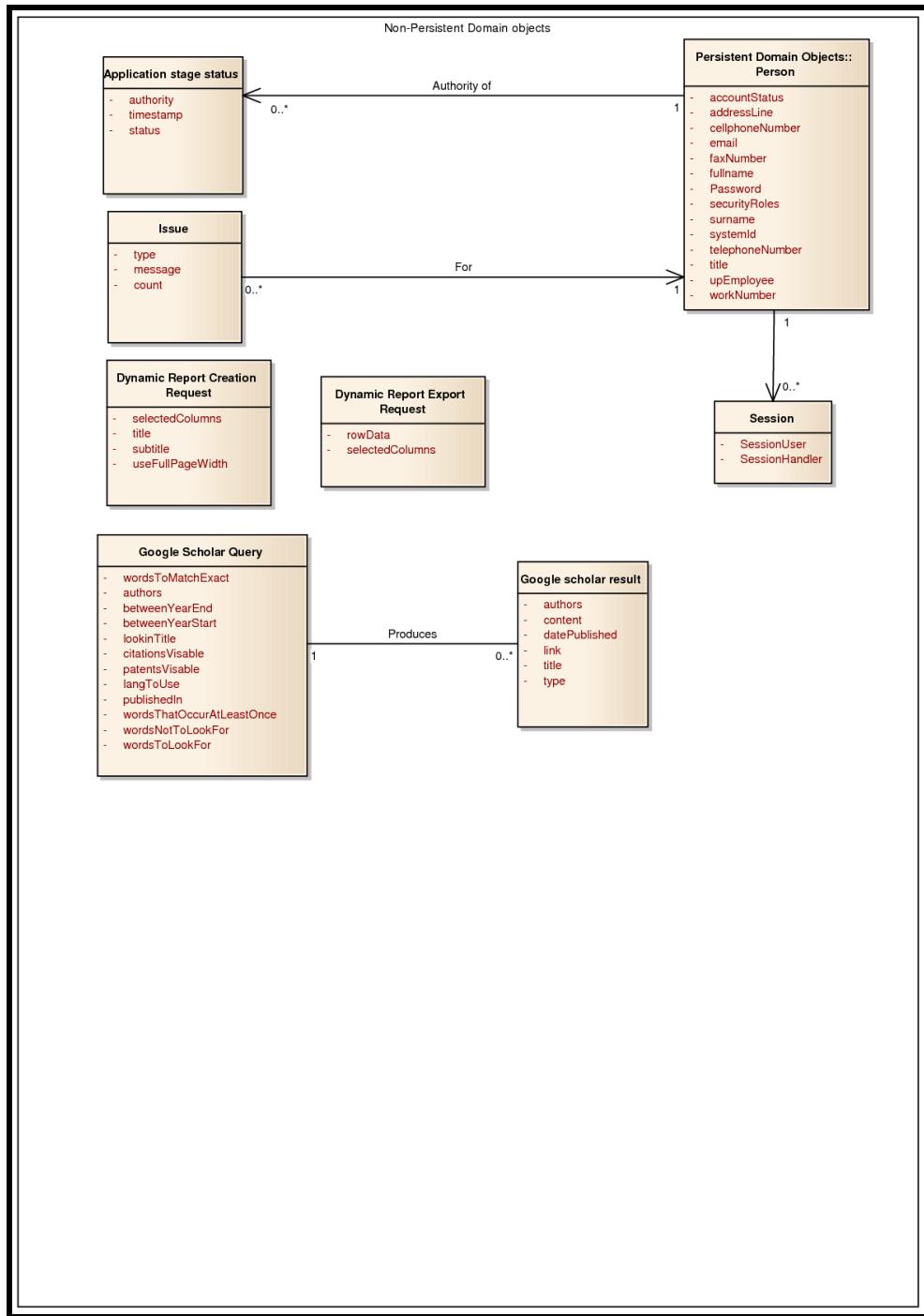


Figure 31: Overview of the data structures and relationships for the non-persistent domain objects of the system

- **ApplicationStageStatus** This object represents a stage change that has occurred for a particular **Application** which was caused by a **Person** in the system.
- **Issue** This object represents an outstanding issue for a particular **Person** in the system. It consists of a message and the number of such instances.
- **Session** This object represents the details of a currently active session on the system. It contains the user (**Person**) and the link to the system's session management system's record. This object will be used throughout the system for auditing and authentication.
- **DynamicReportCreationRequest** This object represents a report creation request that a **DRIS** member is making to the system. It contains all the configuration information needed to generate a report.
- **DynamicReportExportRequest** This object represents a report export request that a **DRIS** member is making to the system. It contains all the configuration and data information needed to generate the report to a particular format.
- **GoogleScholarQuery** This object represents a query that is made to the Google Scholar website by the system.
- **GoogleScholarResult** This object represents a result returned by one a **GoogleScholarQuery** to the Google scholar service.

### 3.7 Data flow diagrams

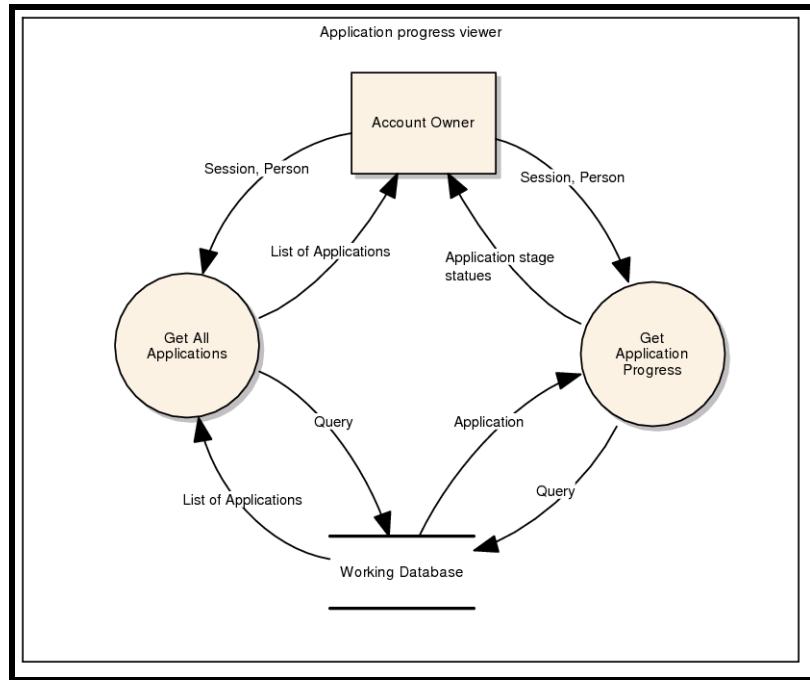


Figure 32: Data Flow for Application progress viewer services

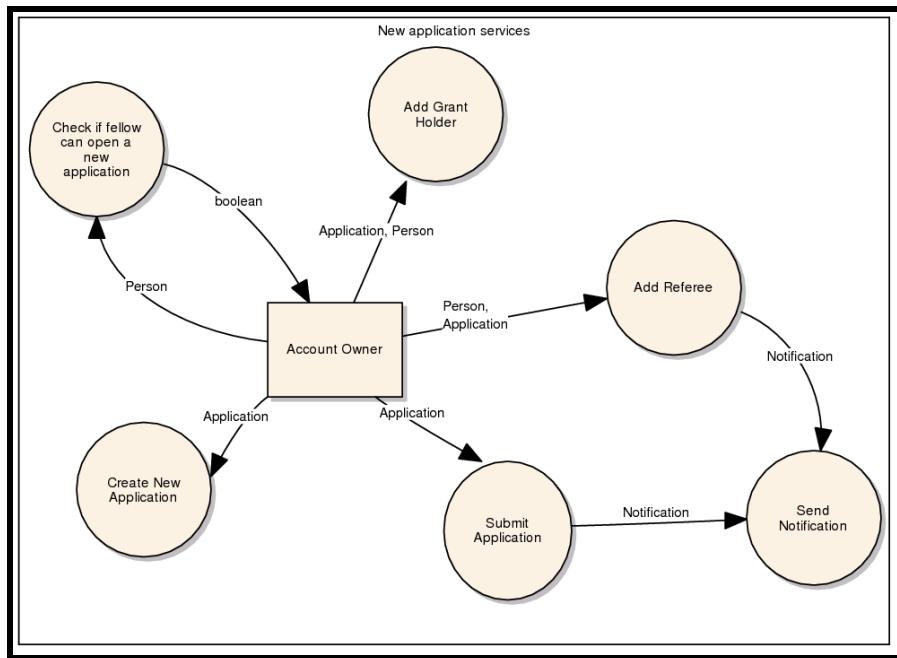


Figure 33: Data Flow for New Application services

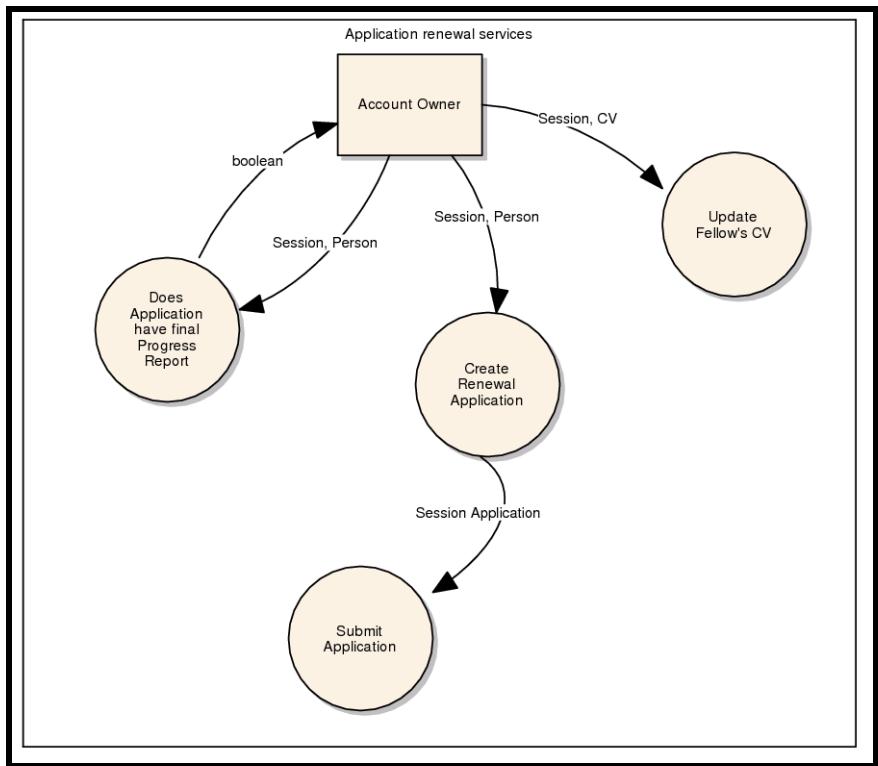


Figure 34: Data Flow for Application Renewal services

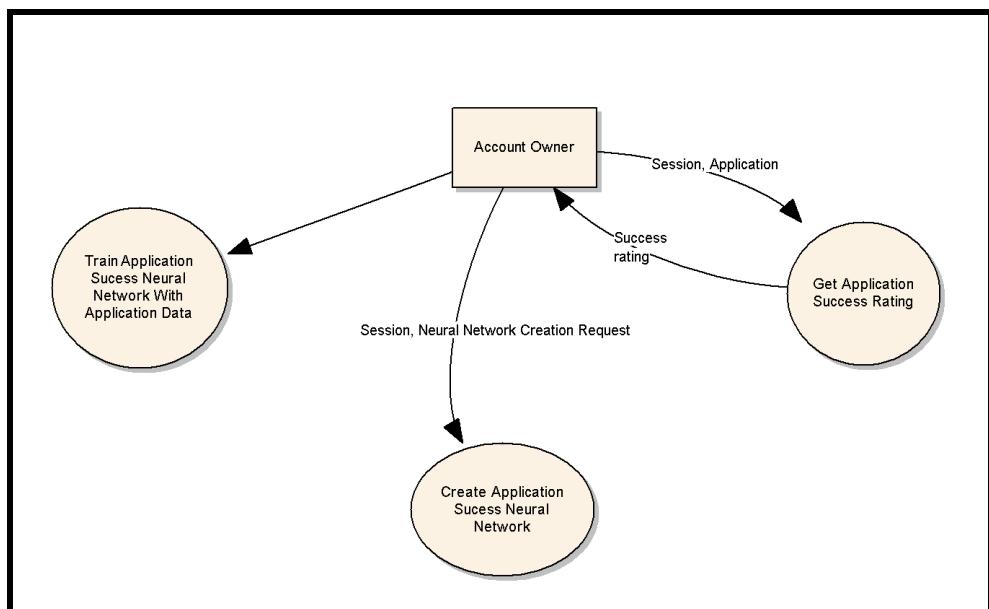


Figure 35: Data Flow for Application Success Evaluation services

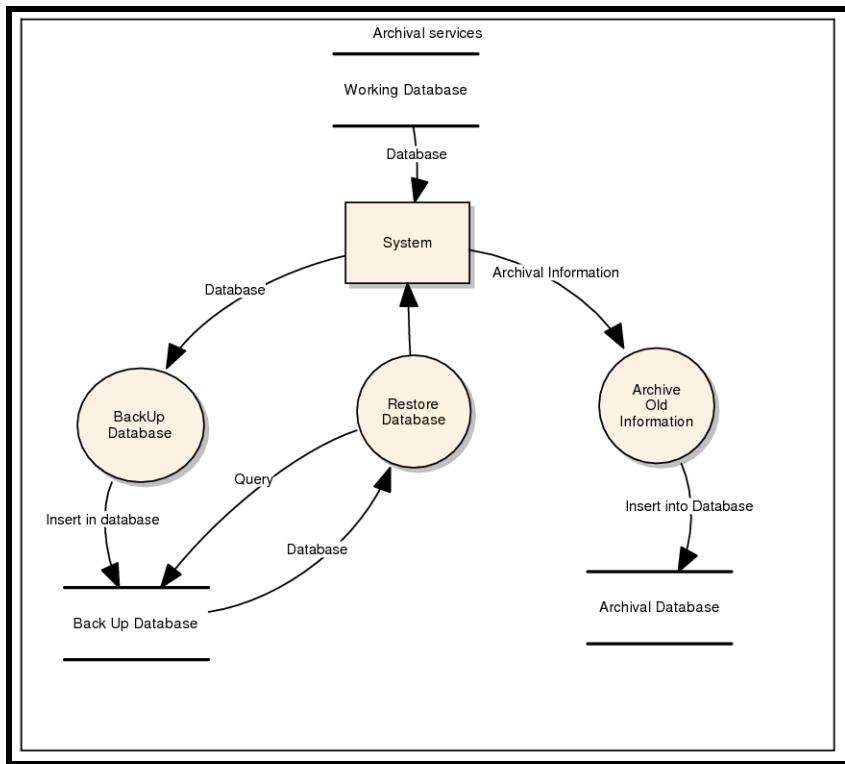


Figure 36: Data Flow for Archival Service services

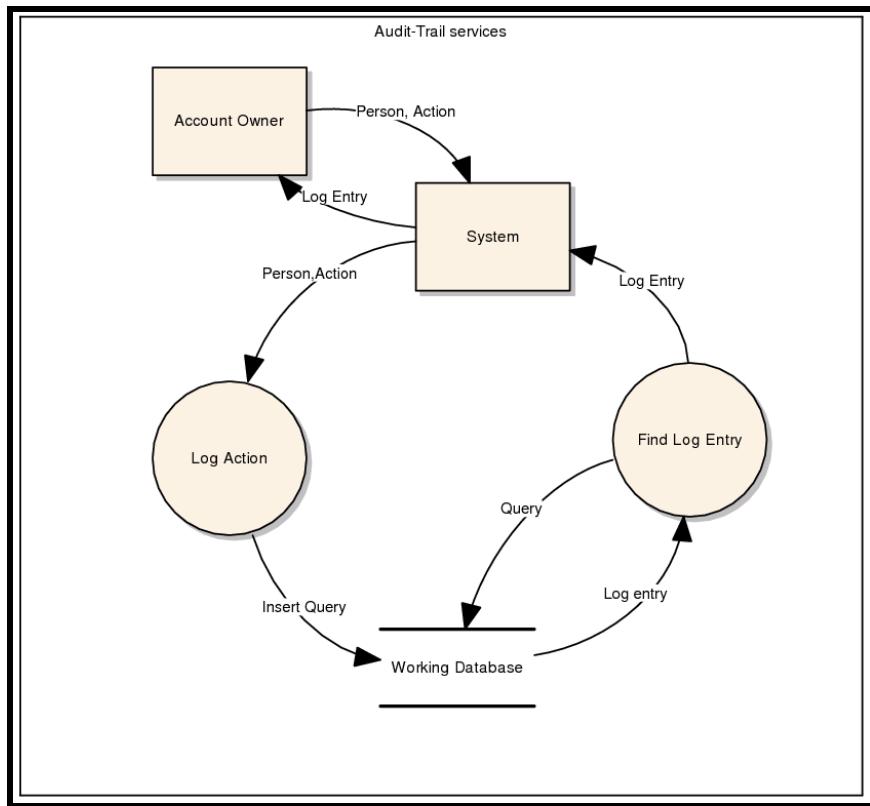


Figure 37: Data Flow for Audit Trail services

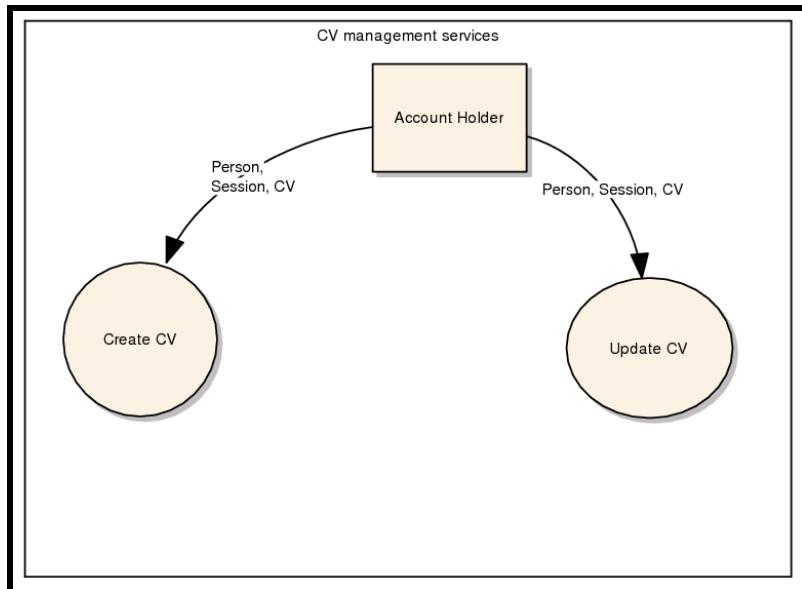


Figure 38: Data Flow CV Management services

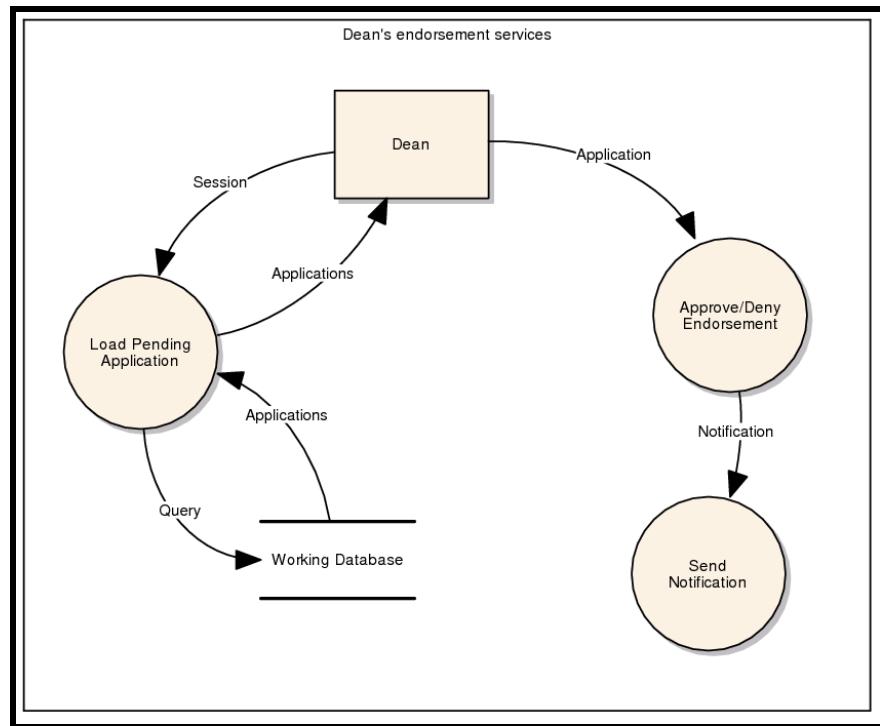


Figure 39: Data Flow for Dean's Endorsement services

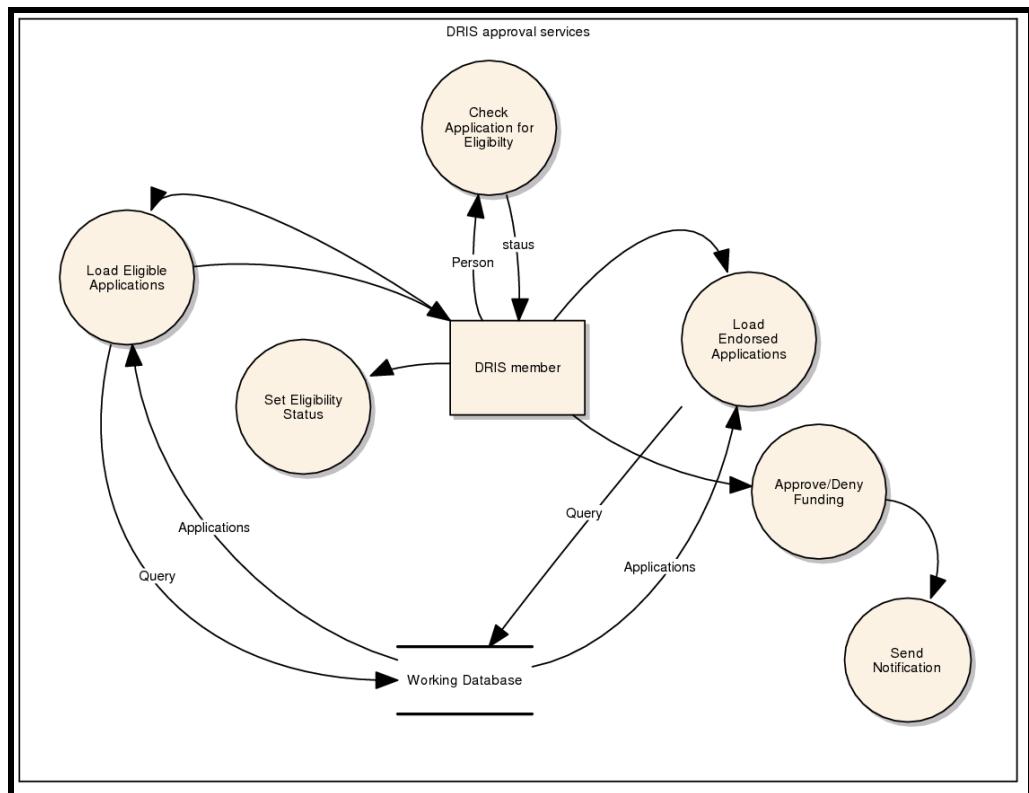


Figure 40: Data Flow for DRIS Approval services

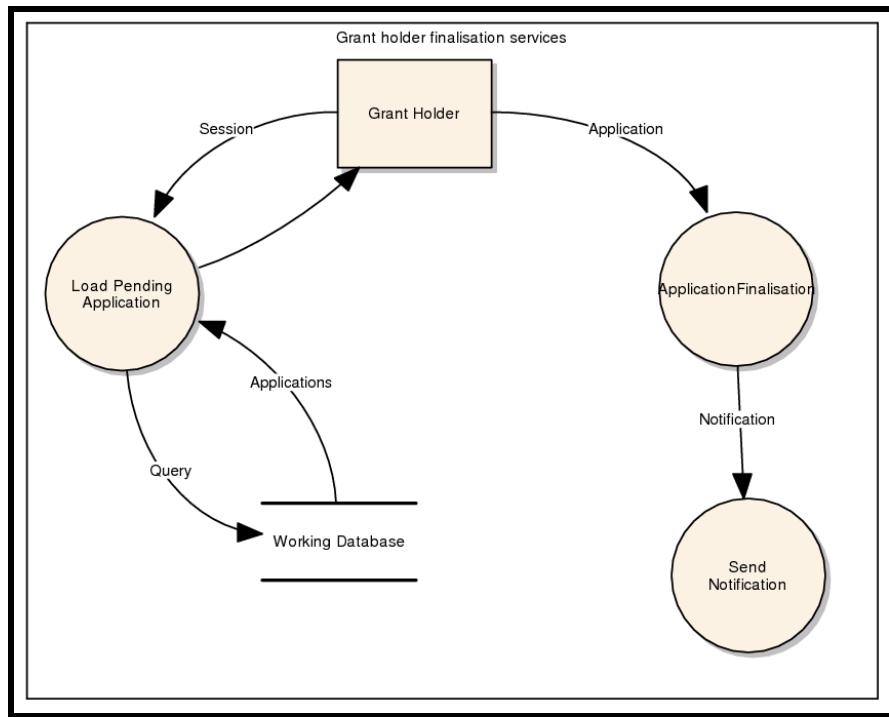


Figure 41: Data Flow for Grant Holder Finalisation services

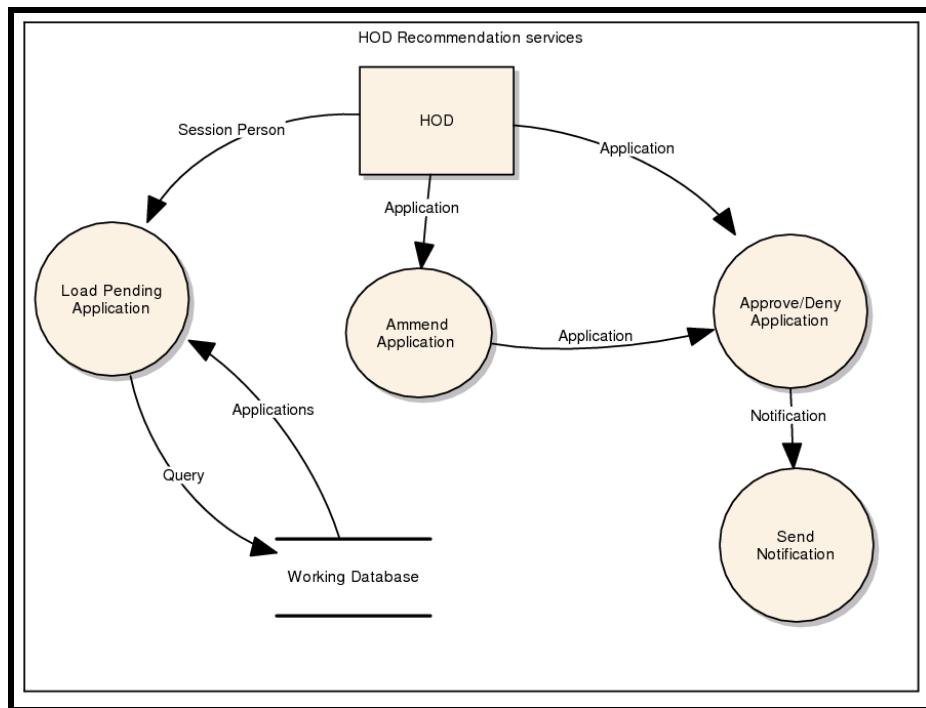


Figure 42: Data Flow for HOD Recommendation services

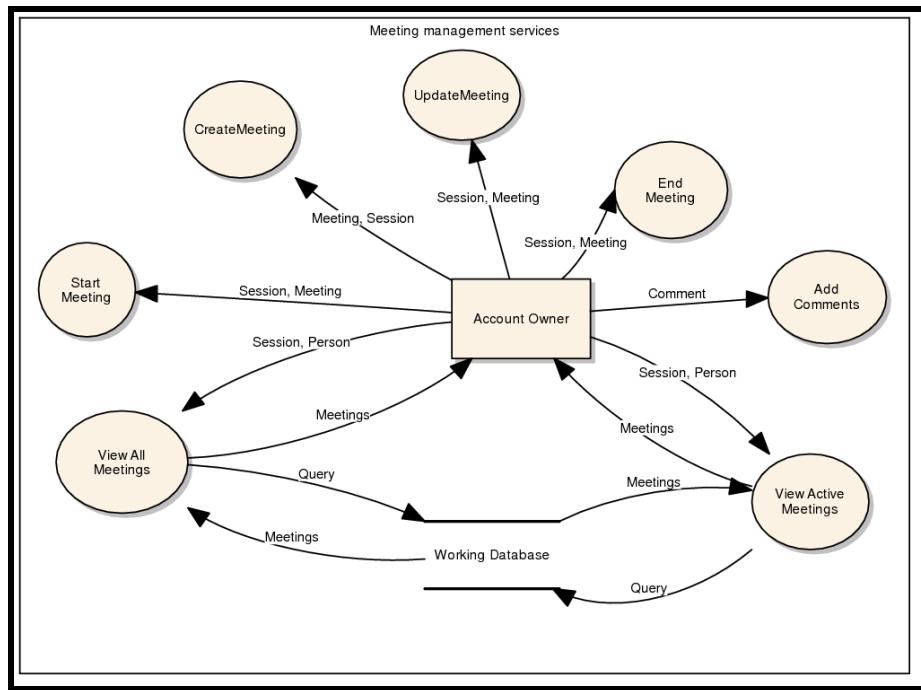


Figure 43: Data Flow for Meeting Management services

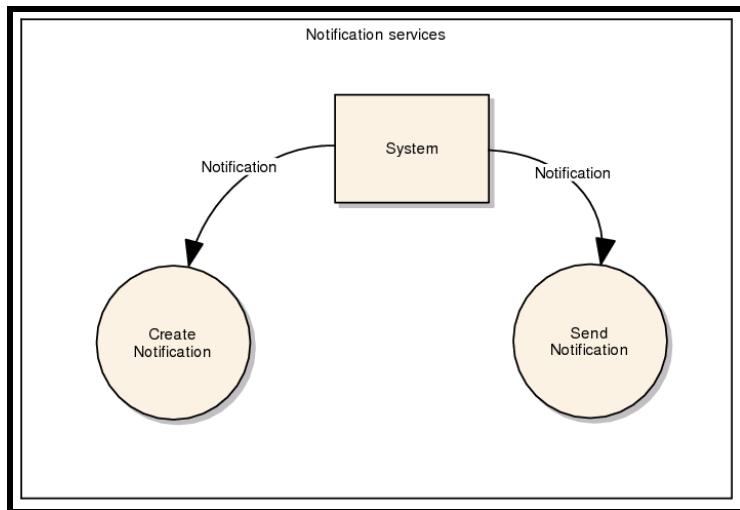


Figure 44: Data Flow for Notification Service services

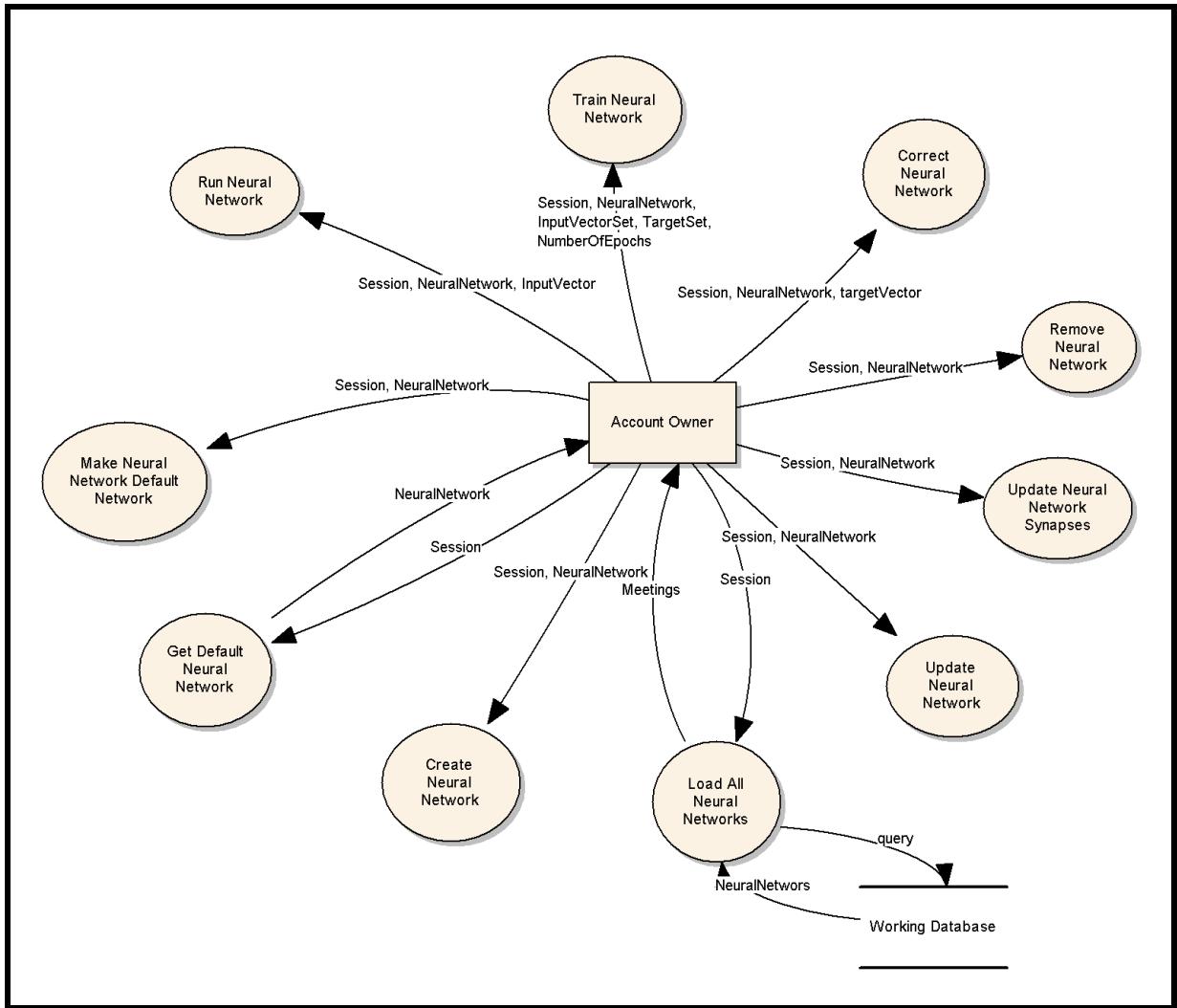


Figure 45: Data Flow for Nueral Network services

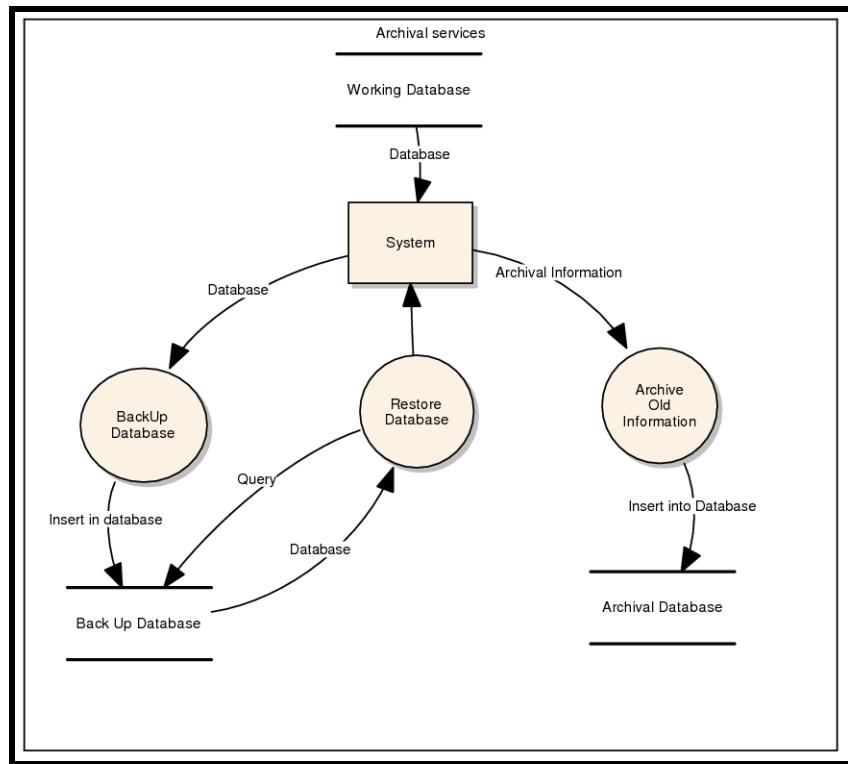


Figure 46: Data Flow for Archival Service services

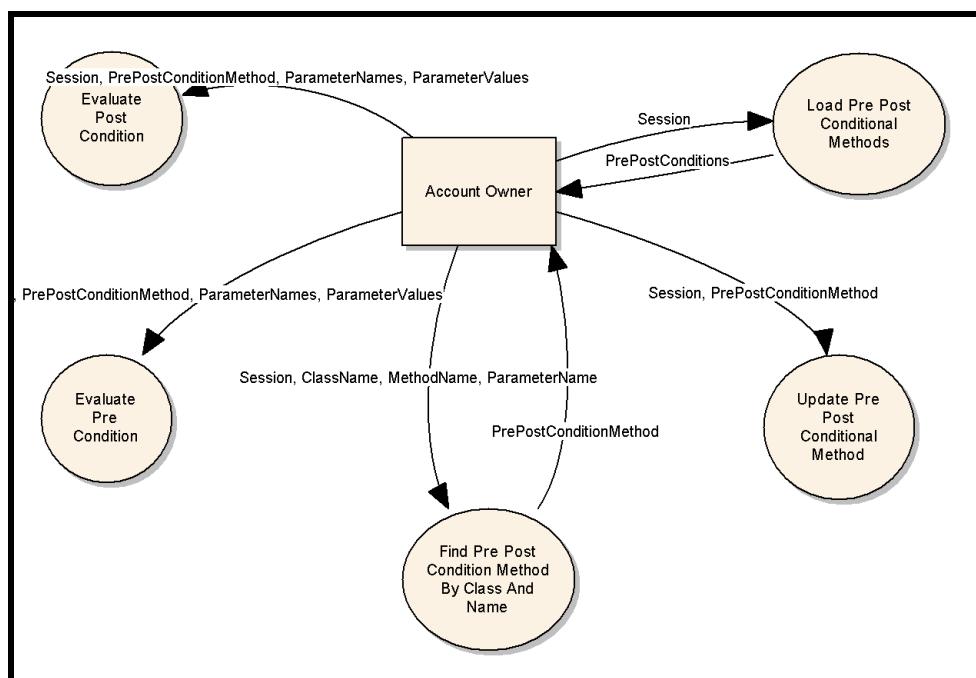


Figure 47: Data Flow for Pre Post Condition services

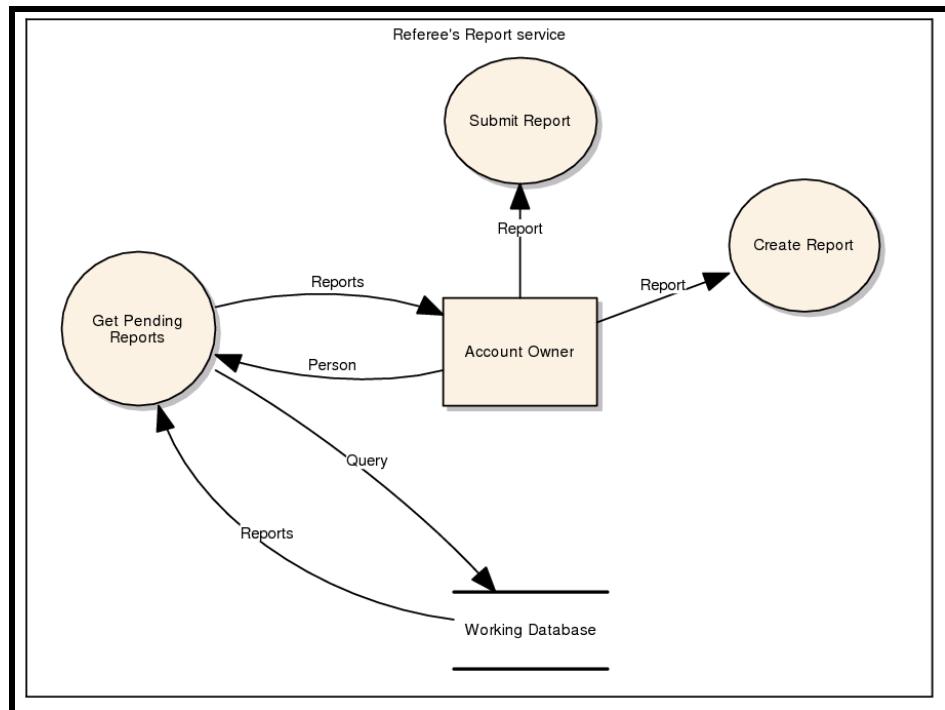


Figure 48: Data Flow for Referee Report services

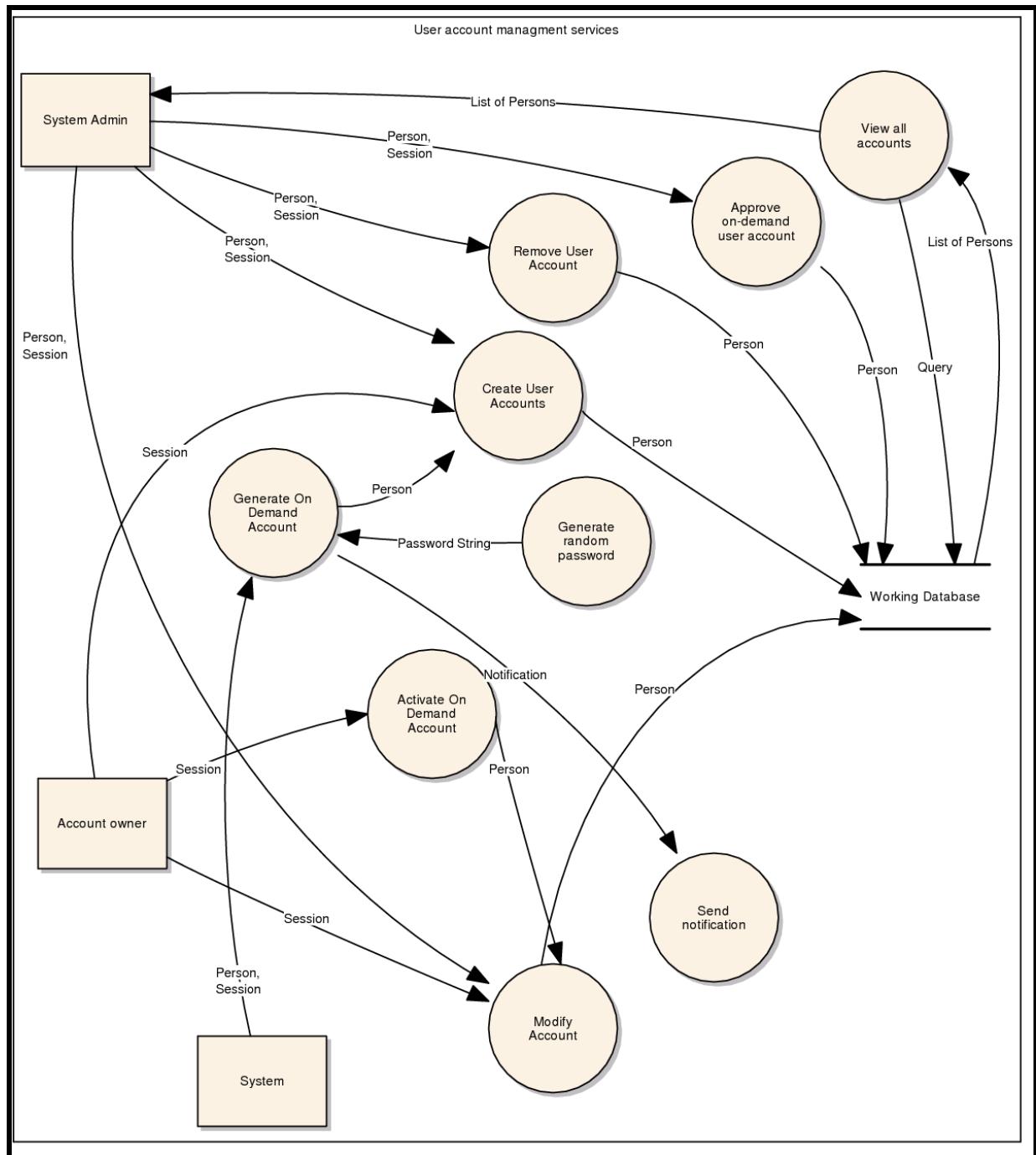


Figure 49: Data Flow for User Account Management services

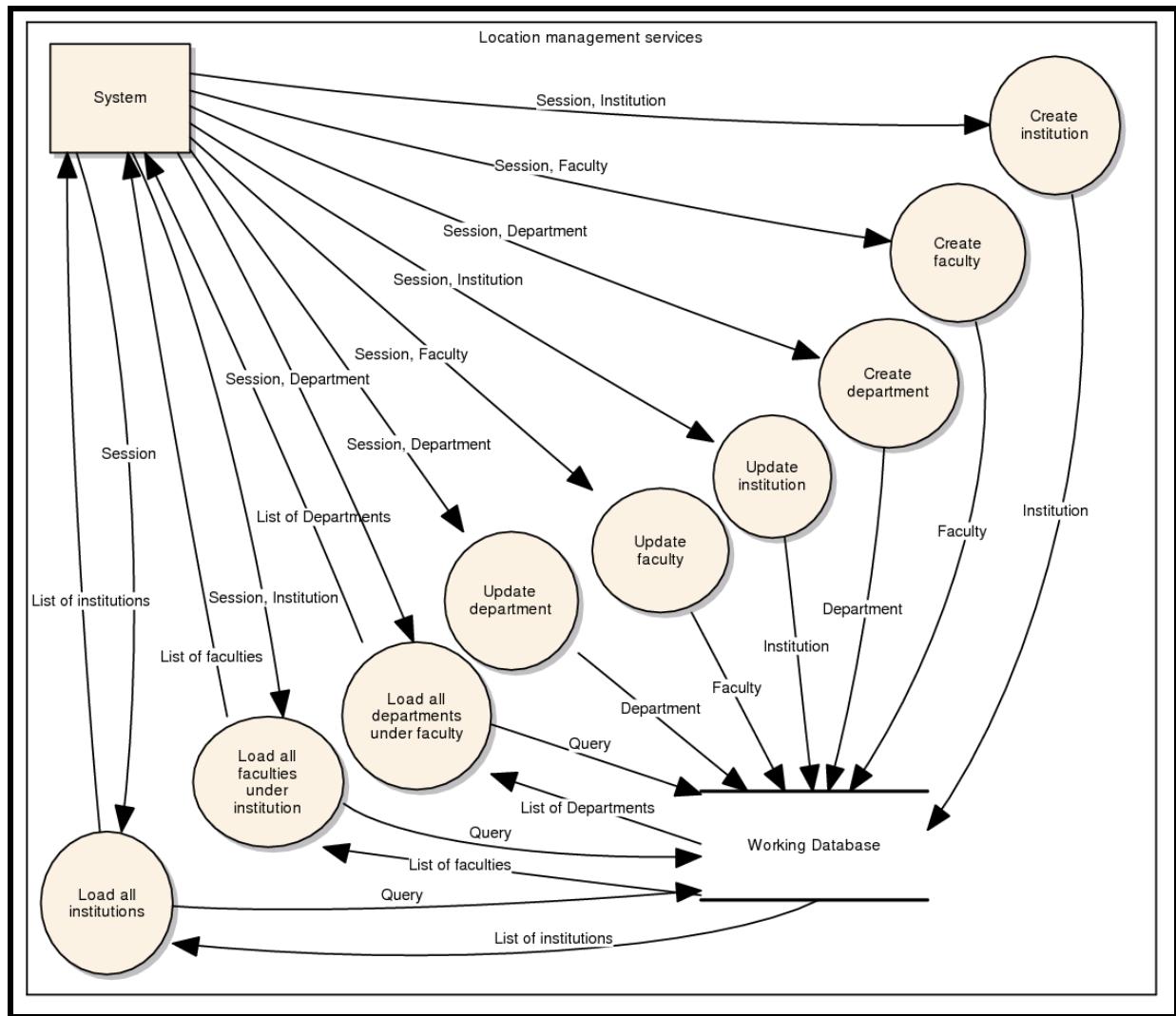


Figure 50: Data Flow for Location management services

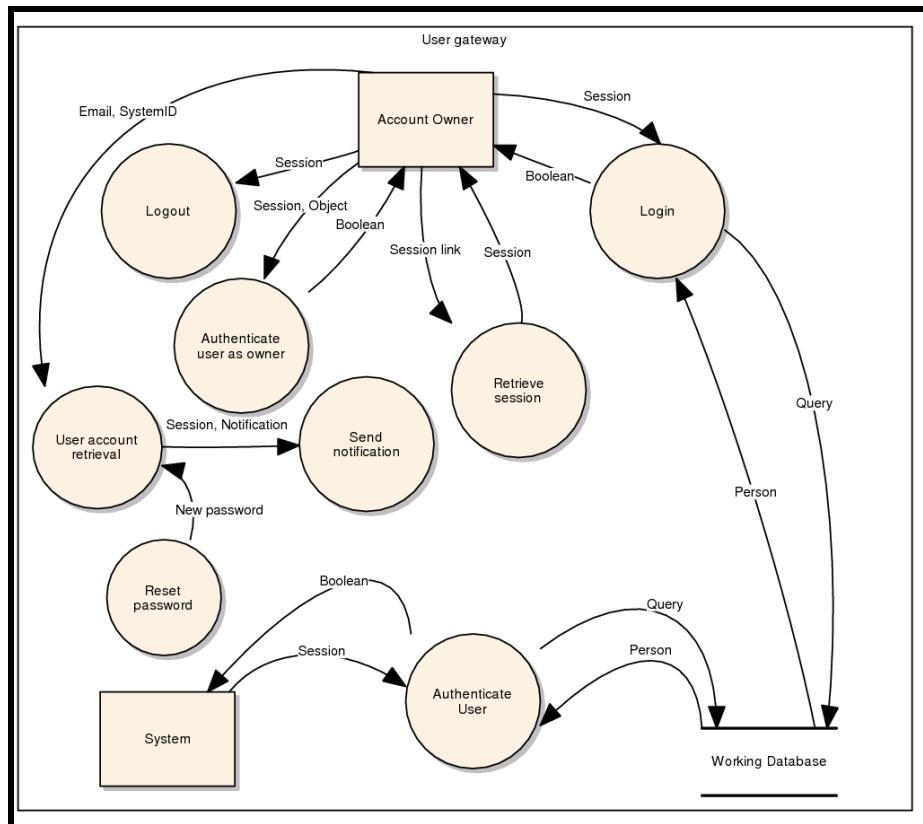


Figure 51: Data Flow for User Gateway.

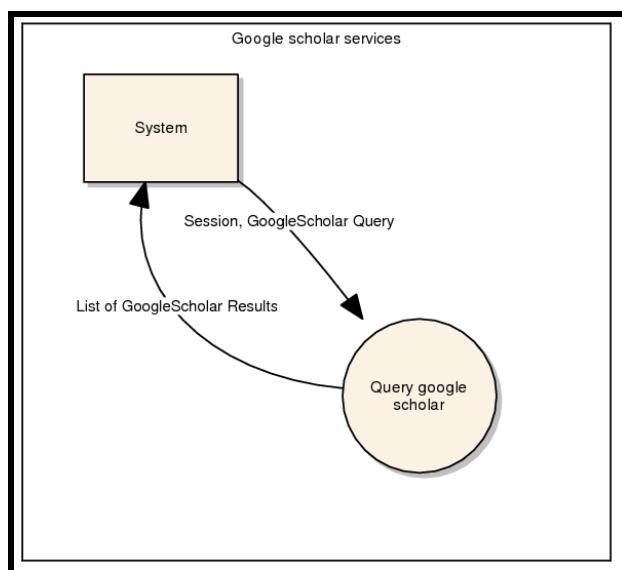


Figure 52: Data Flow for Google scholar services

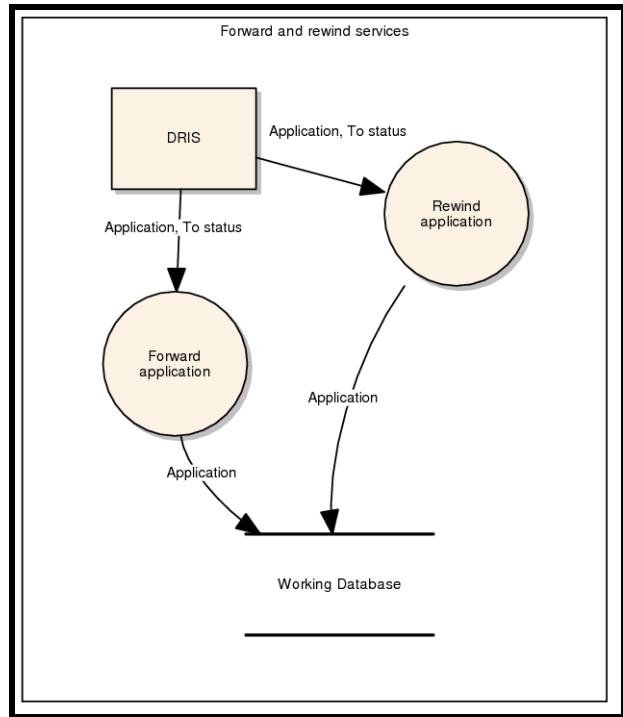


Figure 53: Data Flow for Forward and rewind services

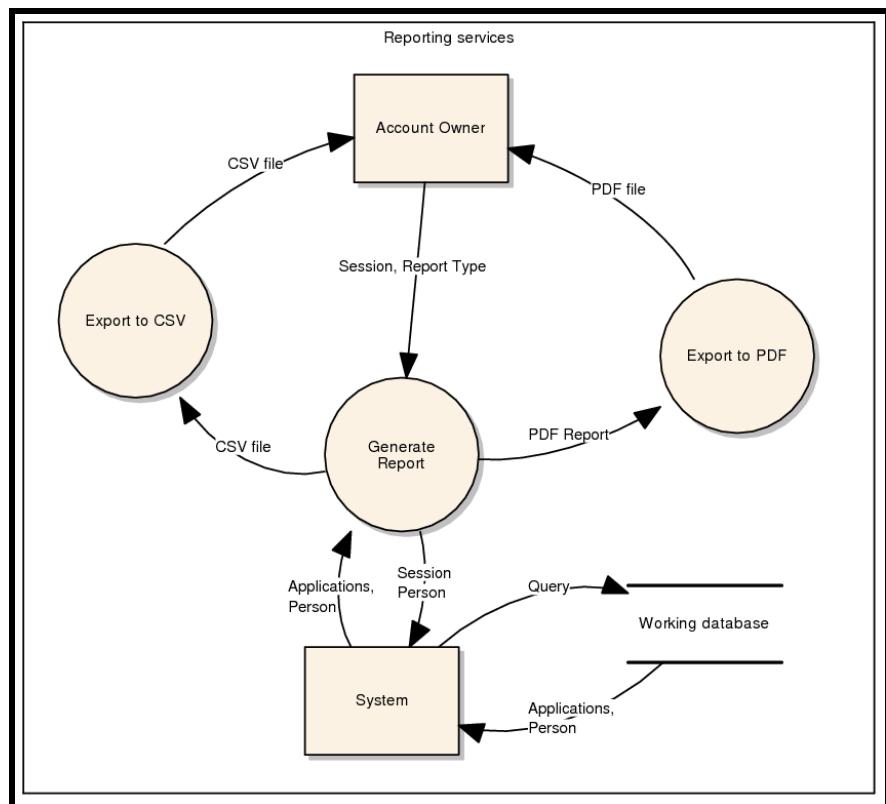


Figure 54: Data Flow for Reporting Service services

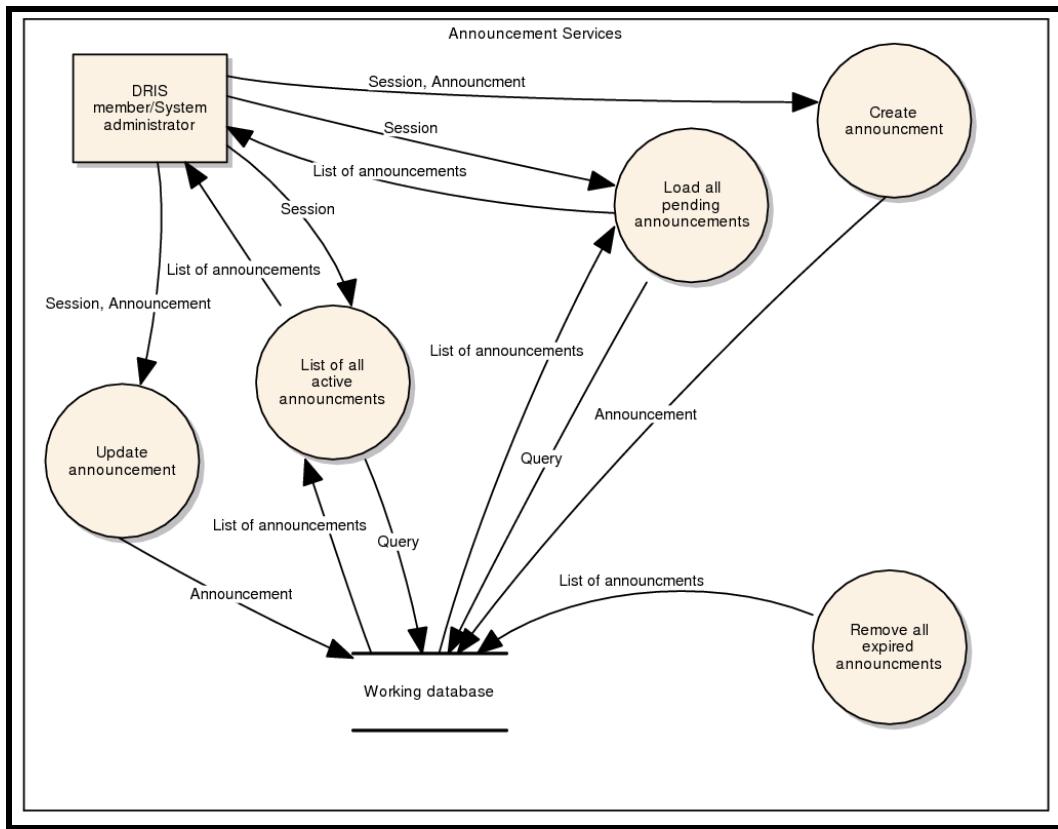


Figure 55: Data Flow for Announcement services

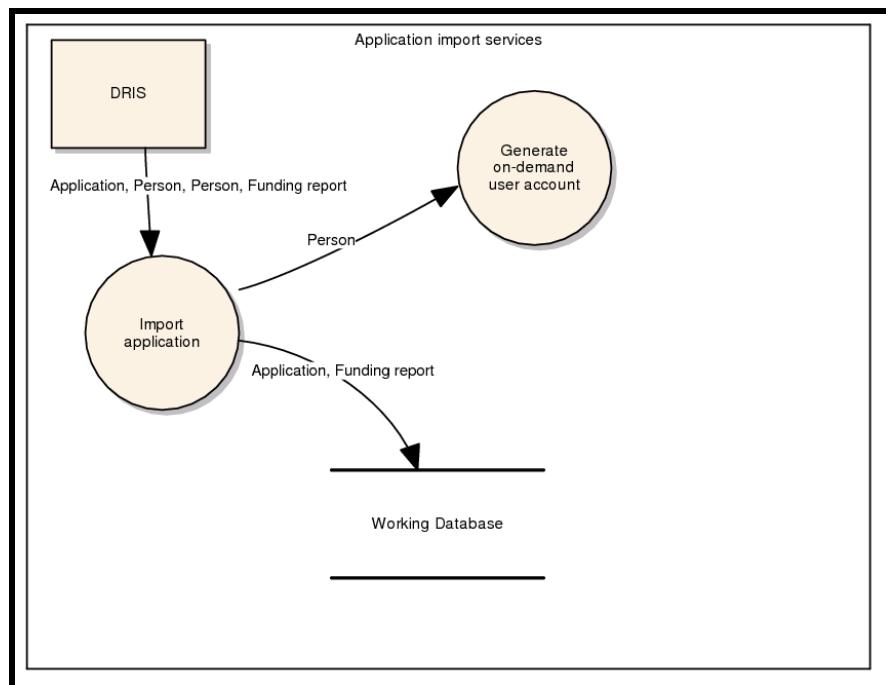


Figure 56: Data Flow for Application import services

### 3.8 Process specifications

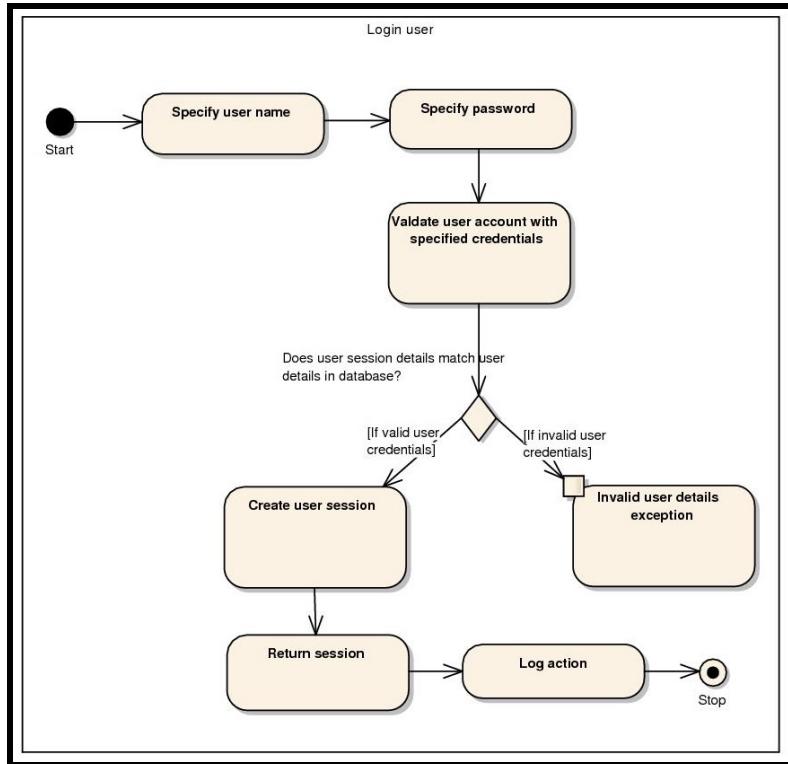


Figure 57: Activity diagram of the Login user use case.

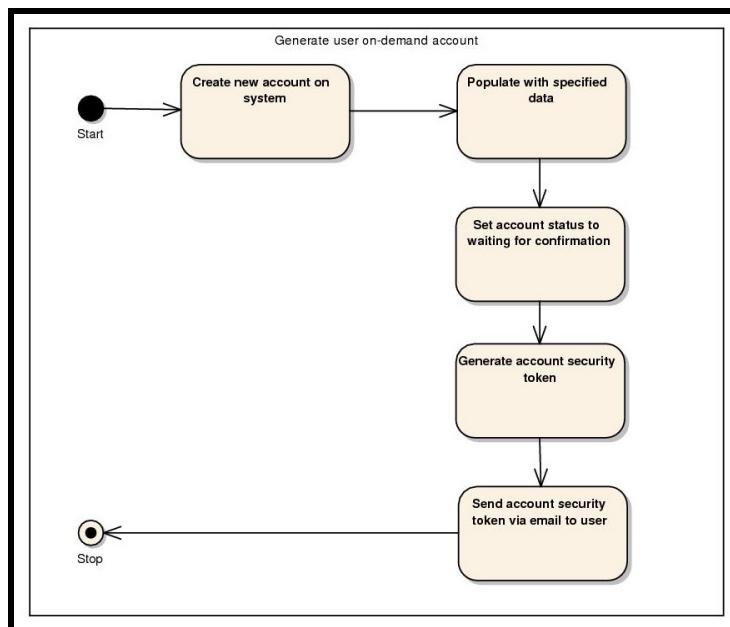


Figure 58: Activity diagram of the Generate user on-demand account use case.

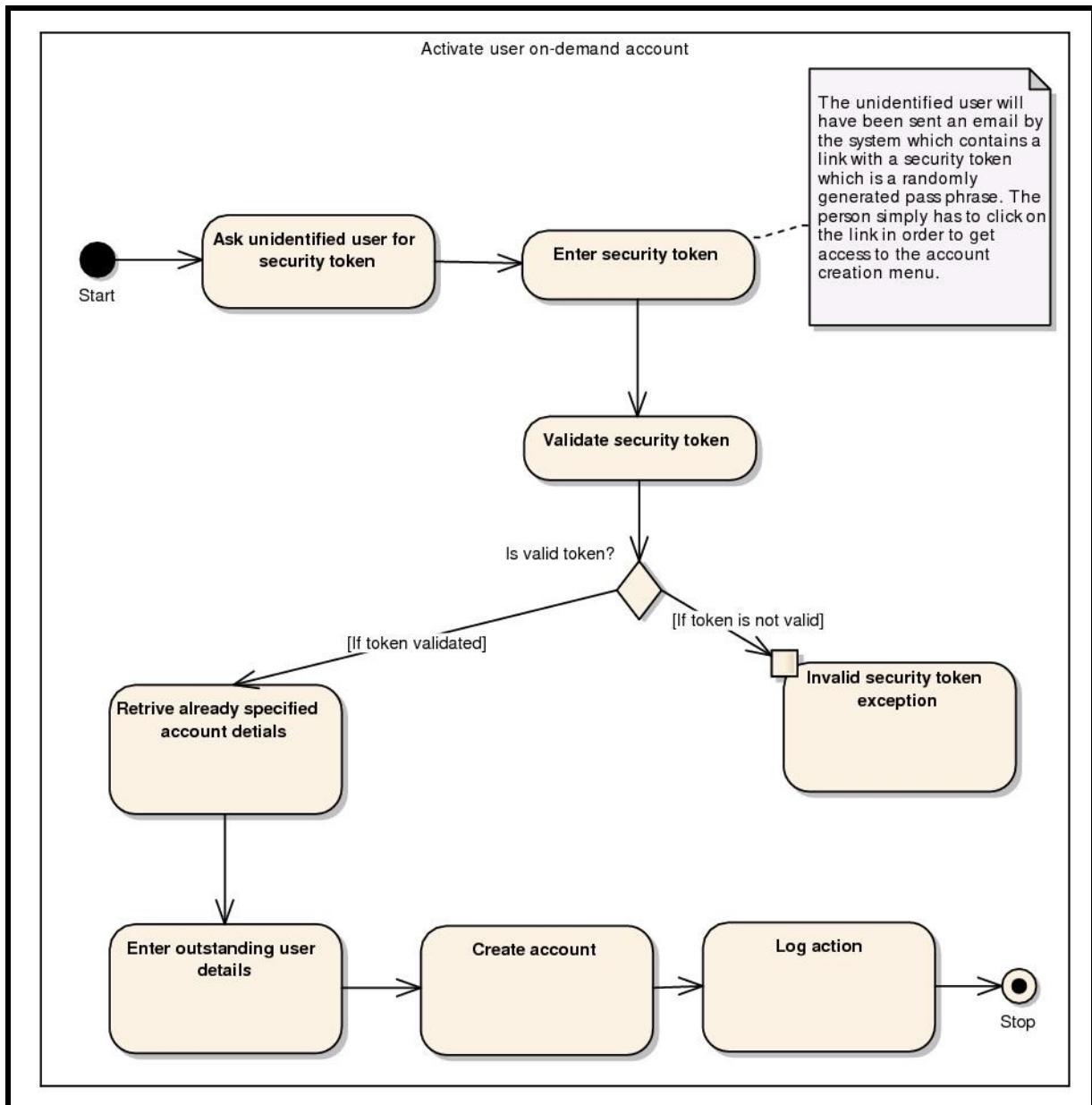


Figure 59: Activity diagram of the Activate user on-demand account use case.

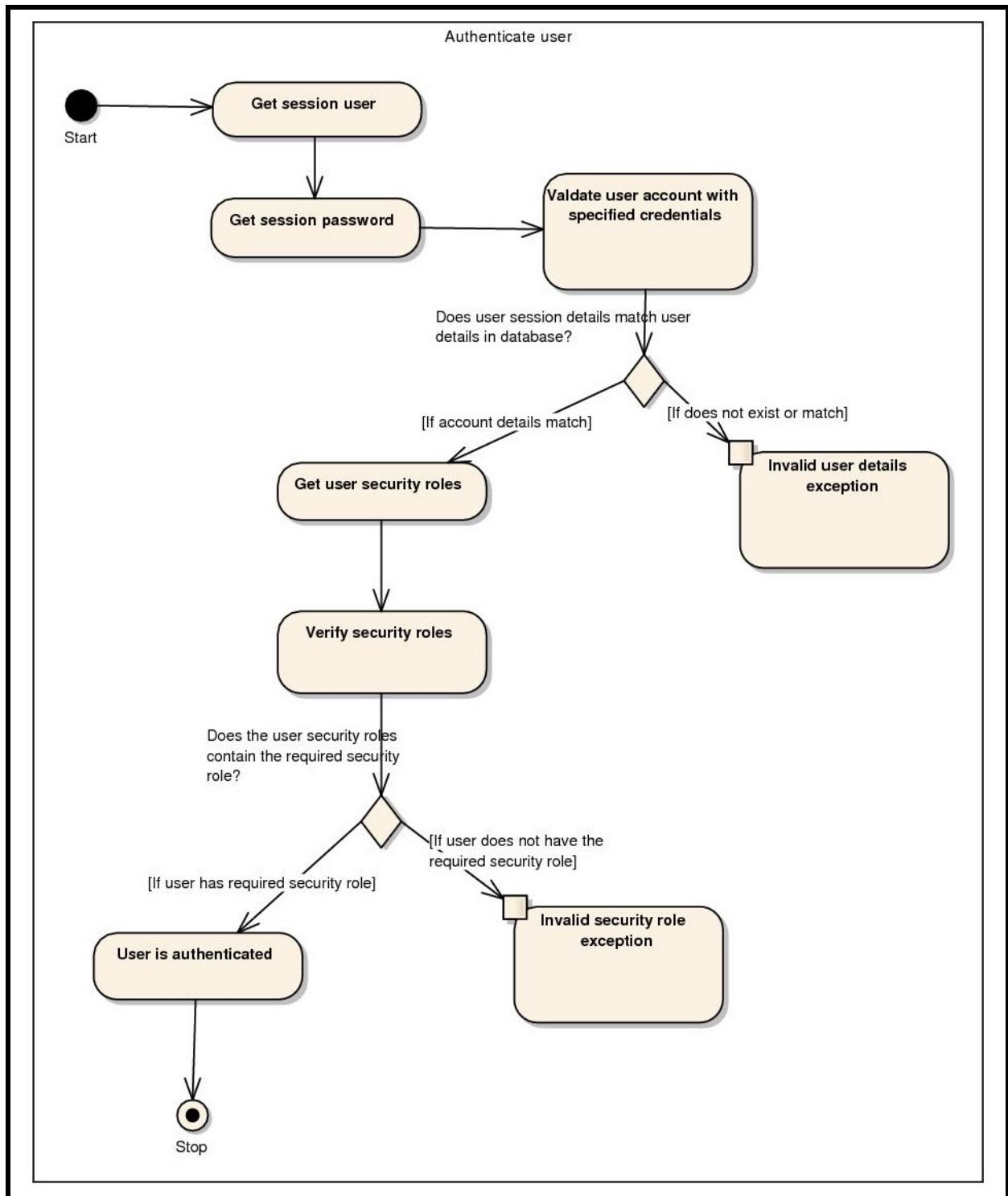


Figure 60: Activity diagram of the Authenticate user use case.

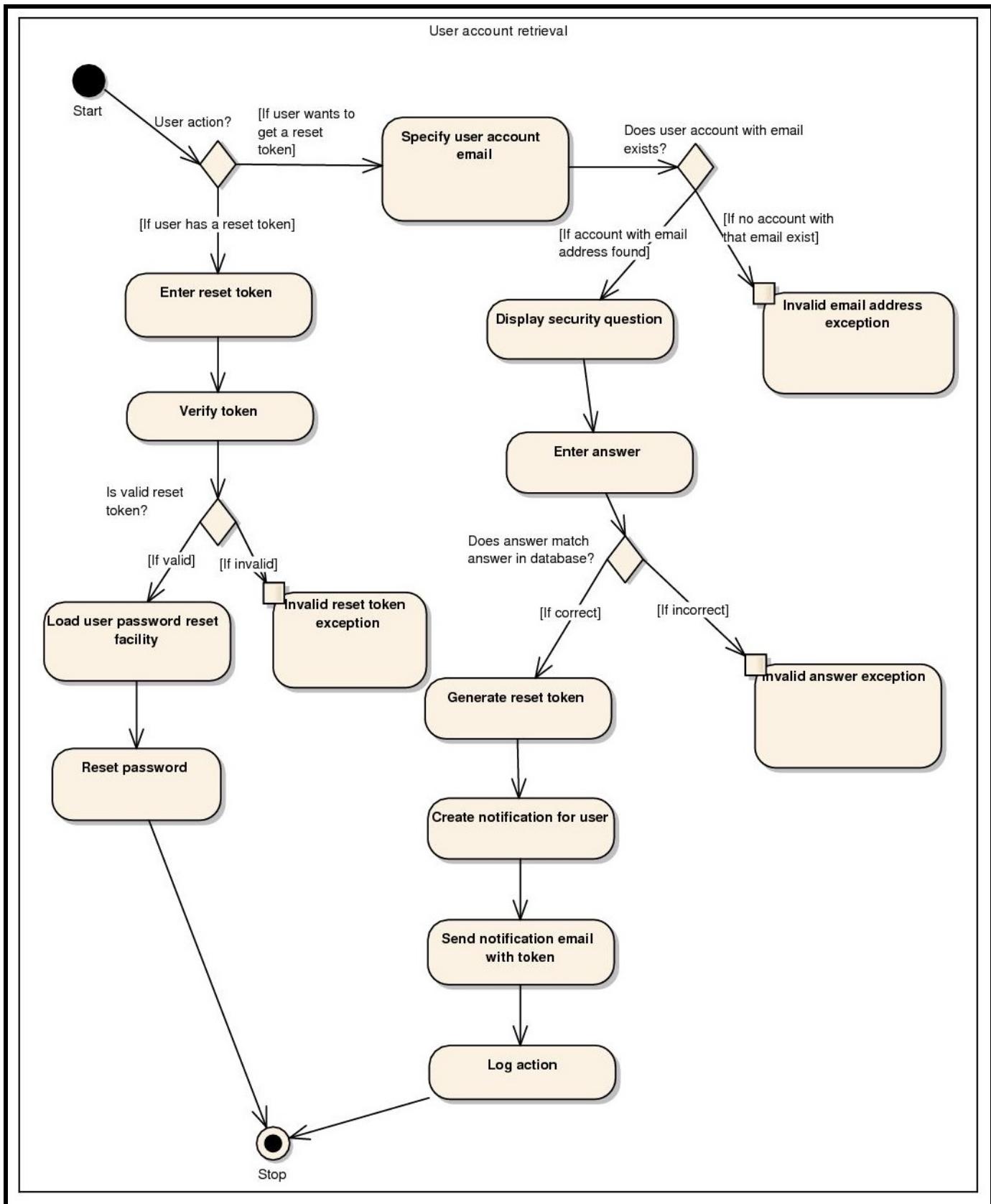


Figure 61: Activity diagram of the User account retrieval use case.

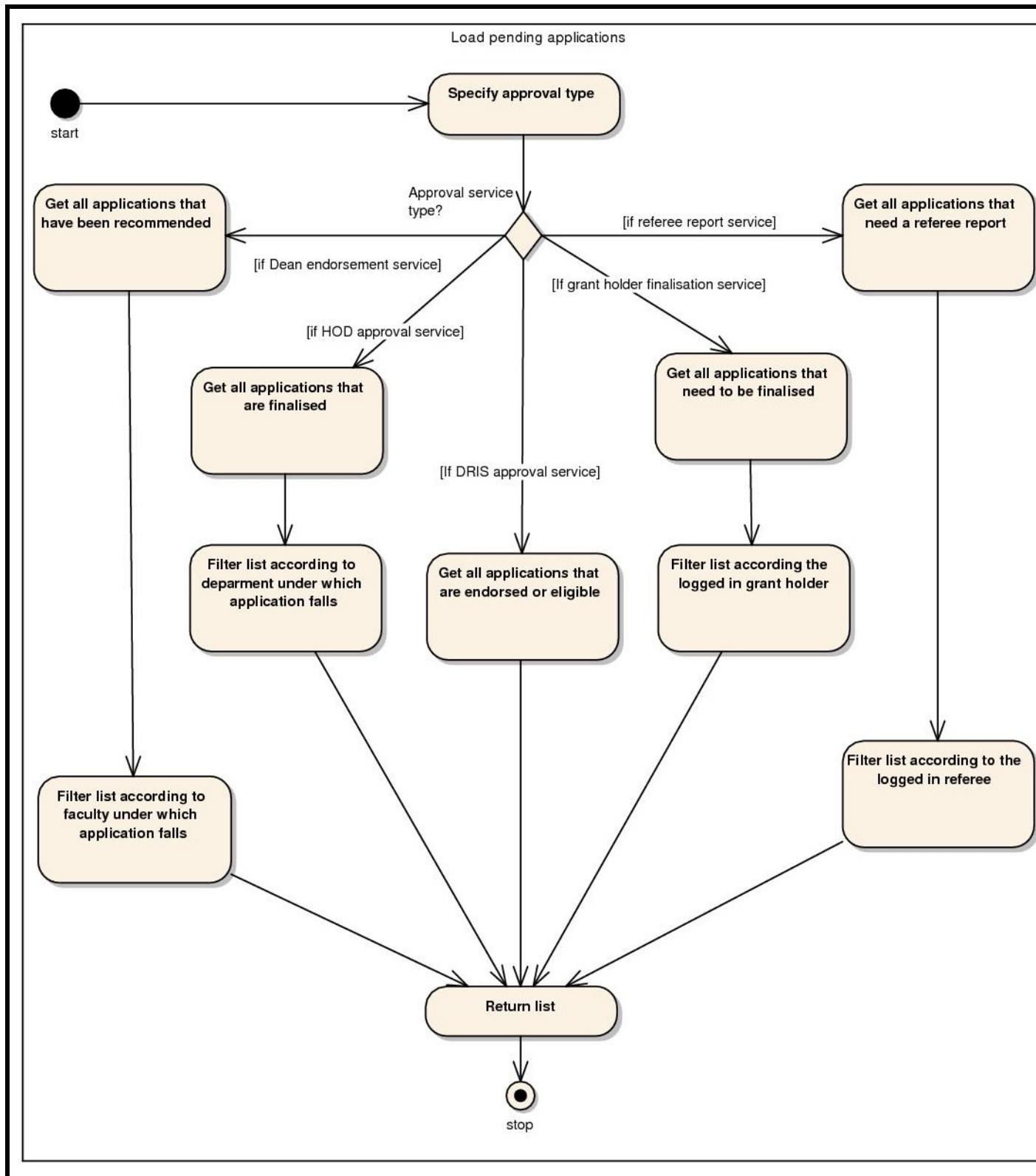


Figure 62: Activity diagram of the Load pending applications use case.

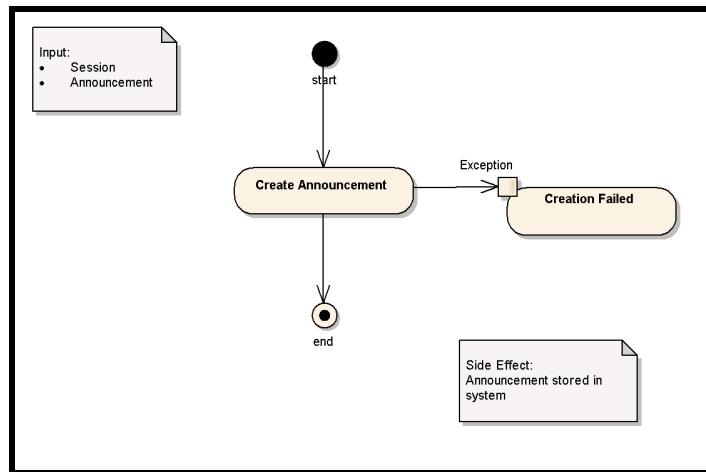


Figure 63: Activity diagram of the create announcement use case.

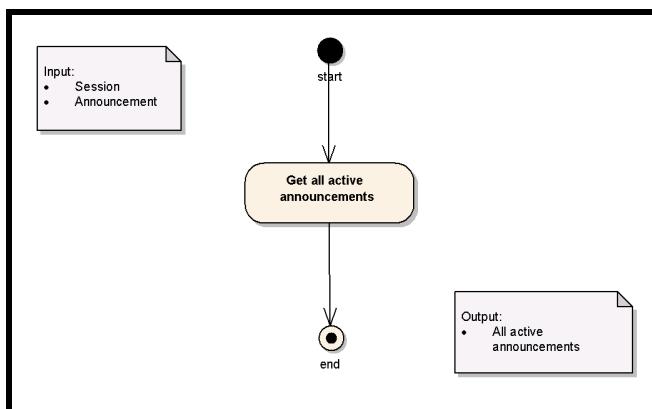


Figure 64: Activity diagram of the load all announcements use case.

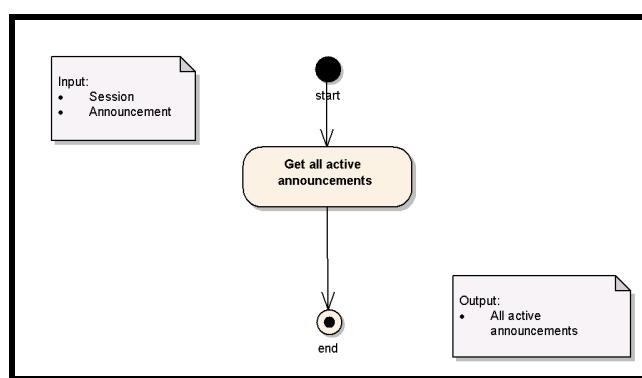


Figure 65: Activity diagram of the load all pending announcements use case.

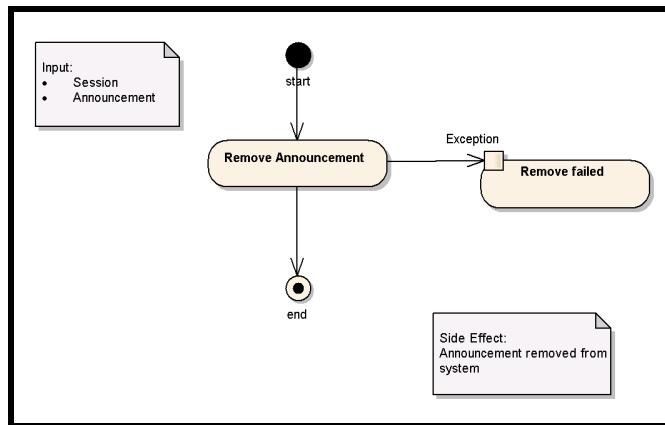


Figure 66: Activity diagram of the remove anouncement use case.

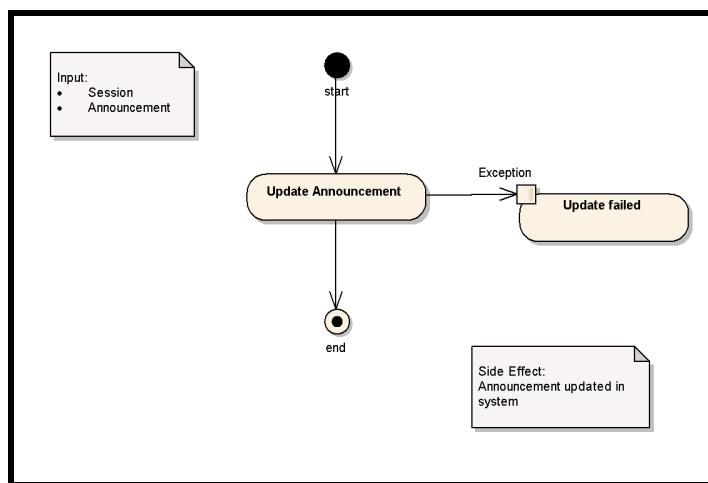


Figure 67: Activity diagram of the update anouncement use case.

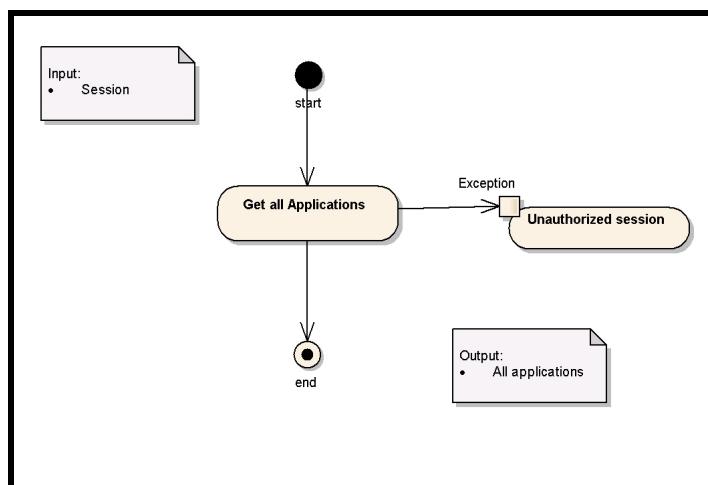


Figure 68: Activity diagram of the Get All Applications use case.

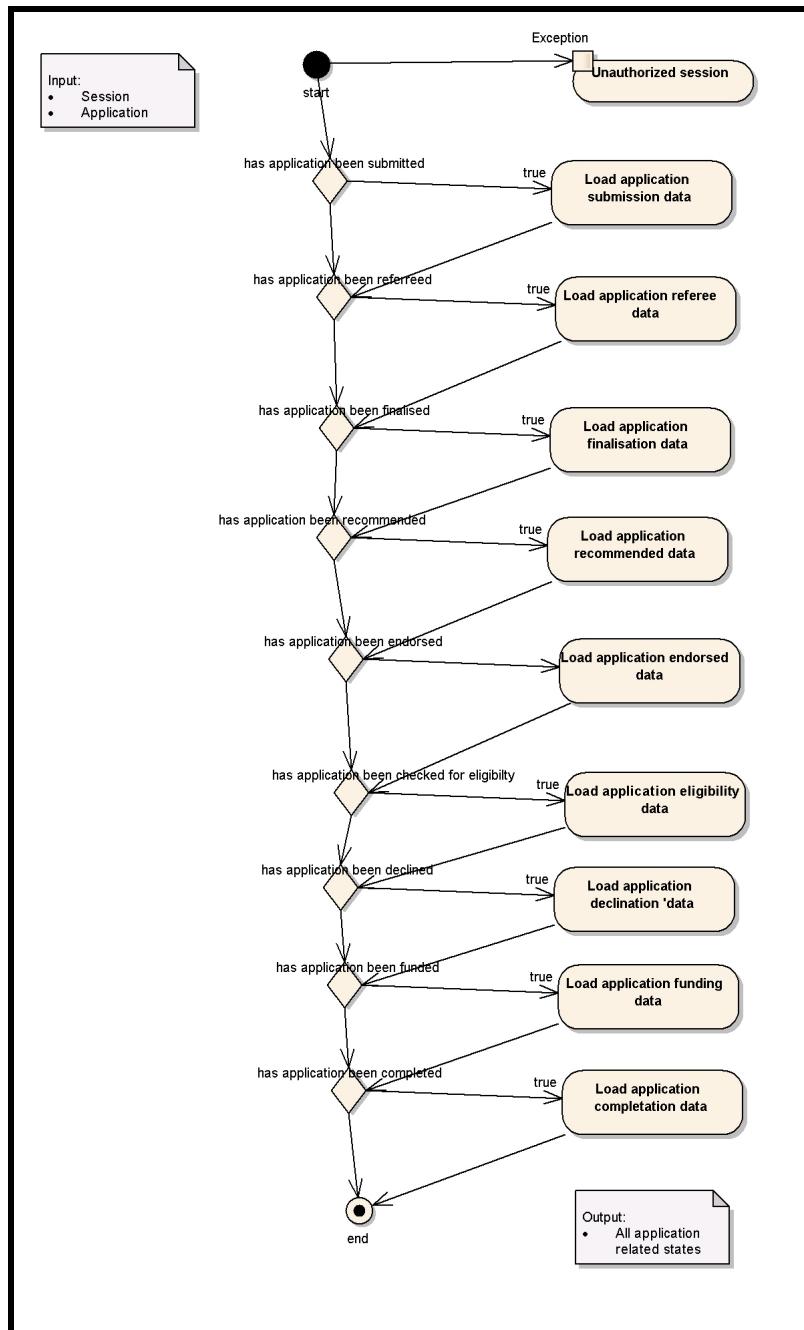


Figure 69: Activity diagram of the View application progress use case.

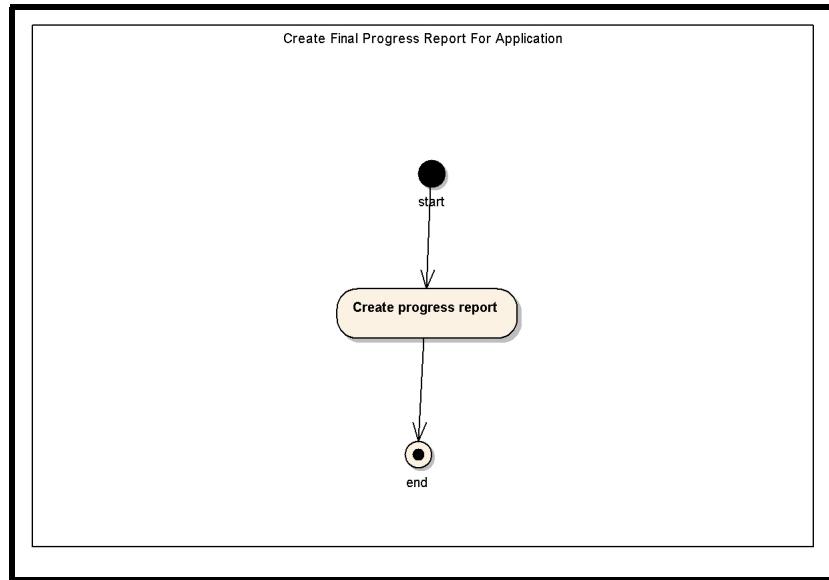


Figure 70: Activity diagram of the Create Final Progress Report For Application use case.

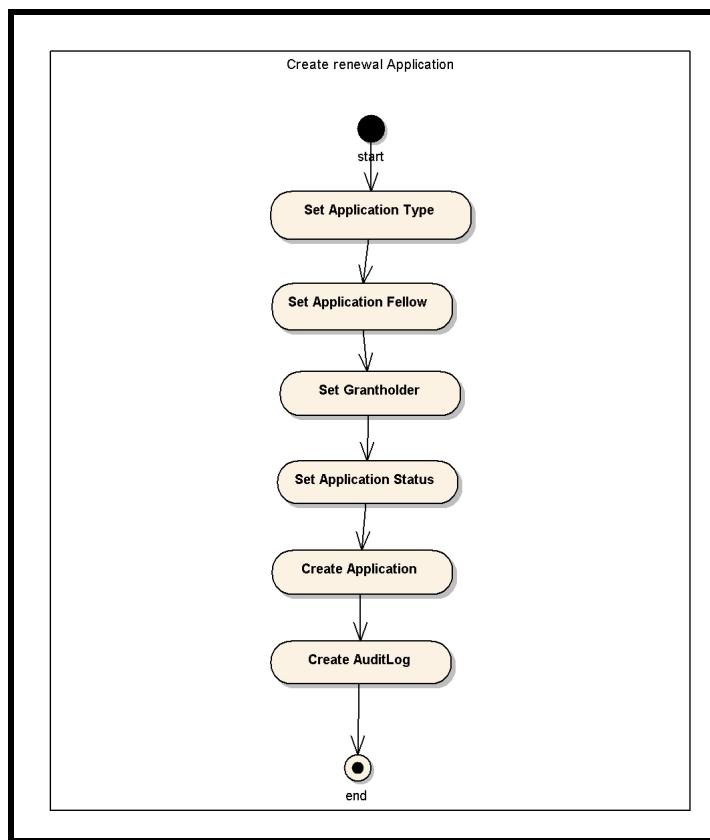


Figure 71: Activity diagram of the Create renewal Application use case.

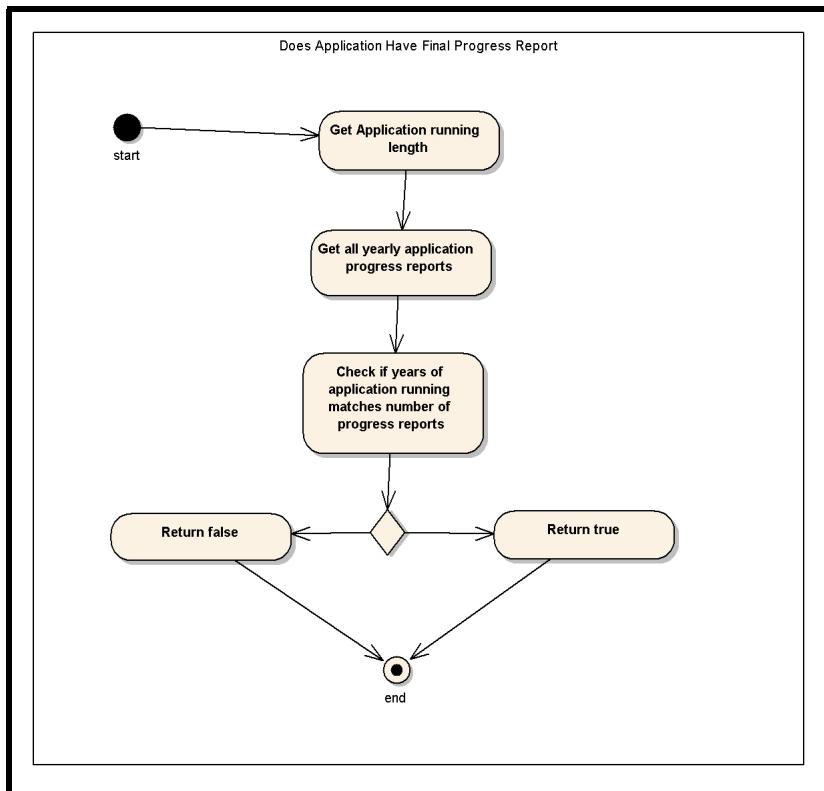


Figure 72: Activity diagram of the Does Application Have Final Progress Report use case.

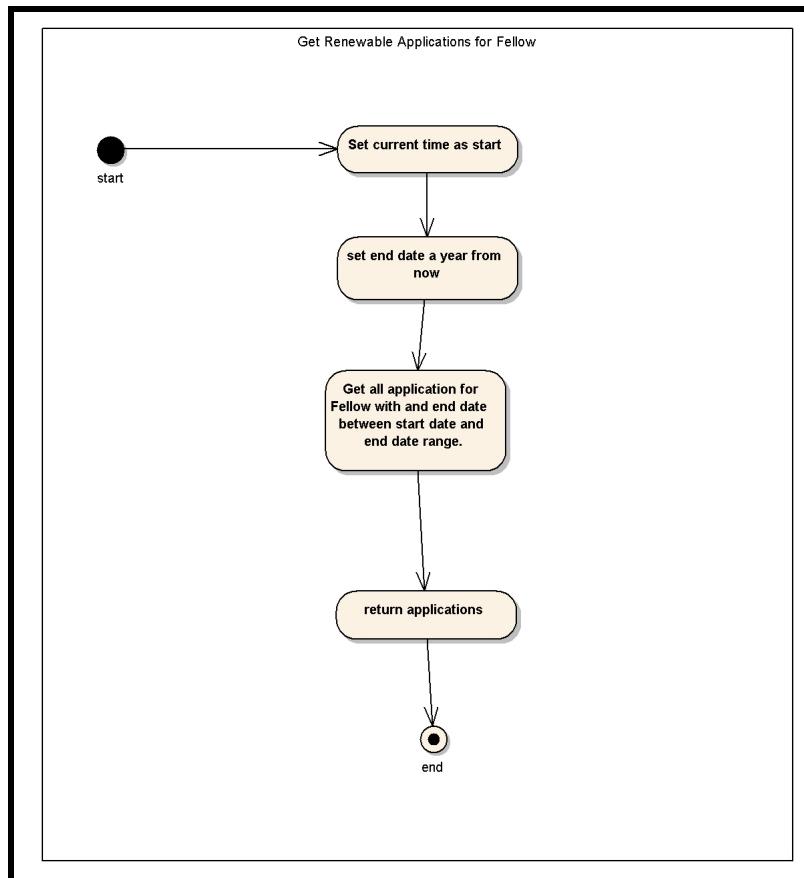


Figure 73: Activity diagram of the Get Renewable Applications for Fellow use case.

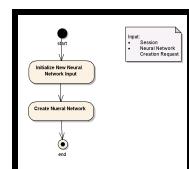


Figure 74: Activity diagram of the Create Application Success Neural Network use case.

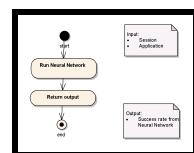


Figure 75: Activity diagram of the Get Application Success Rating use case.

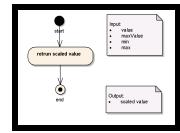


Figure 76: Activity diagram of the Scale Value use case.

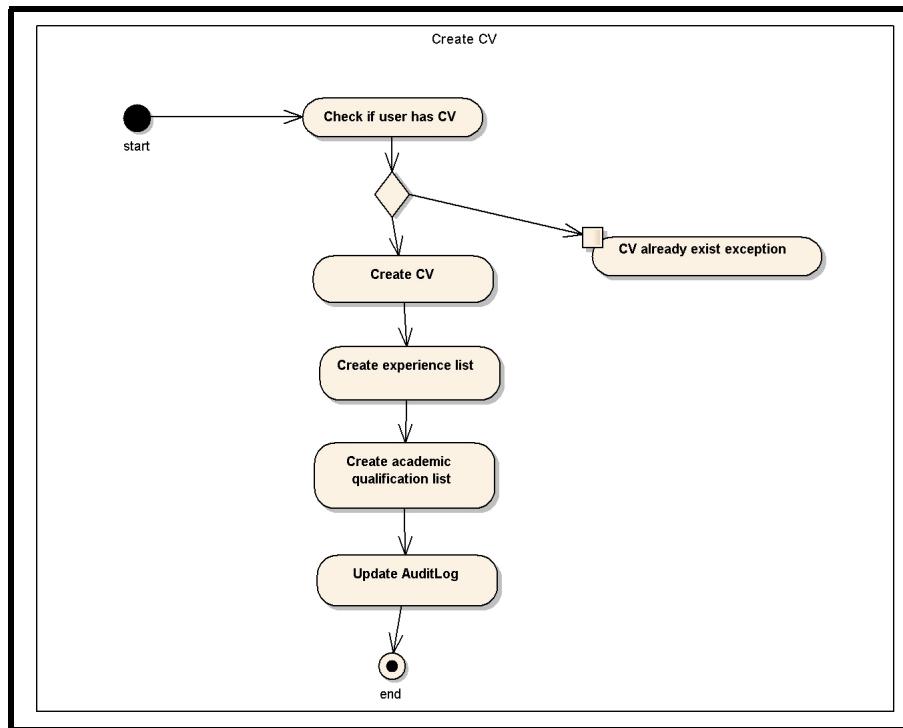


Figure 77: Activity diagram of the Create CV use case.

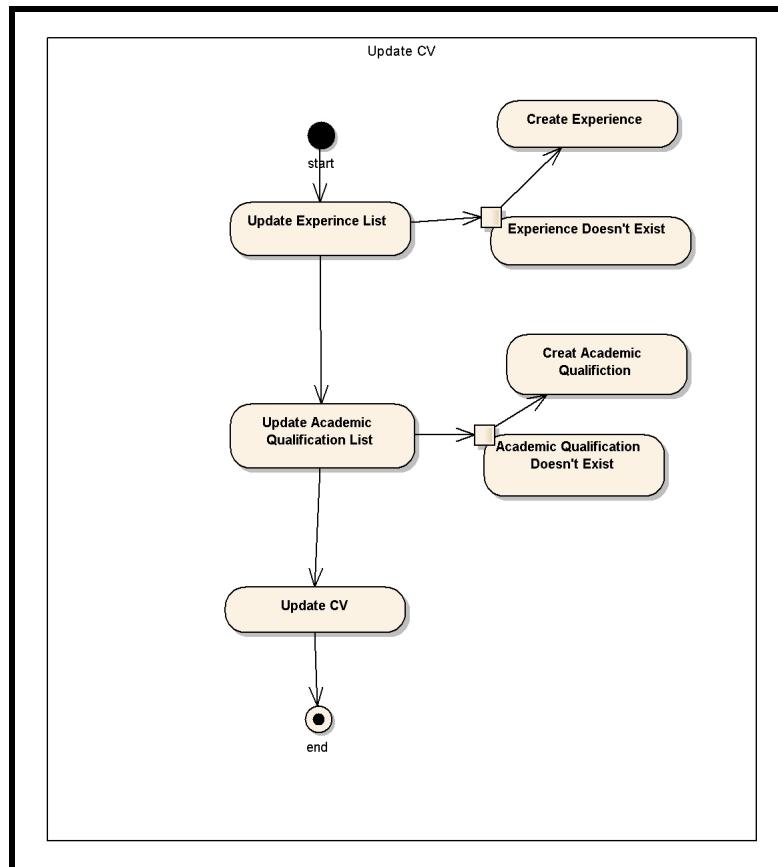


Figure 78: Activity diagram of the Update CV use case.

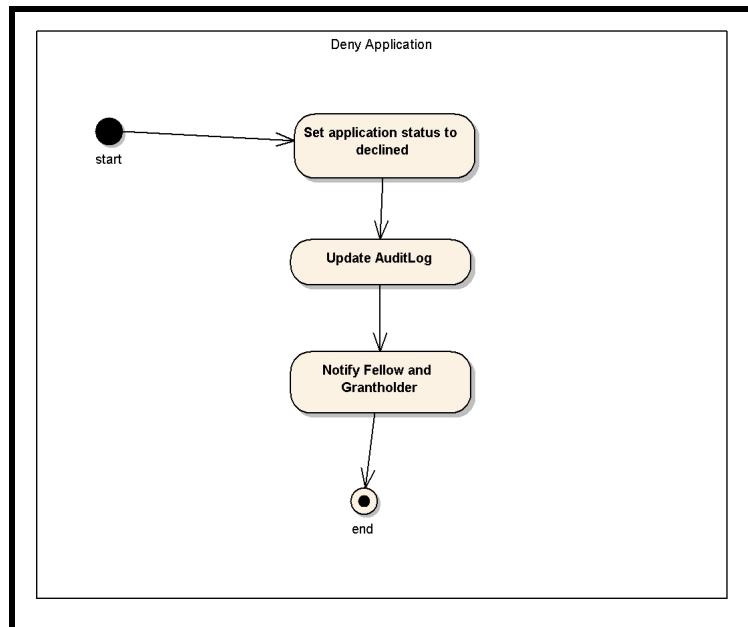


Figure 79: Activity diagram of the Deny Application use case.

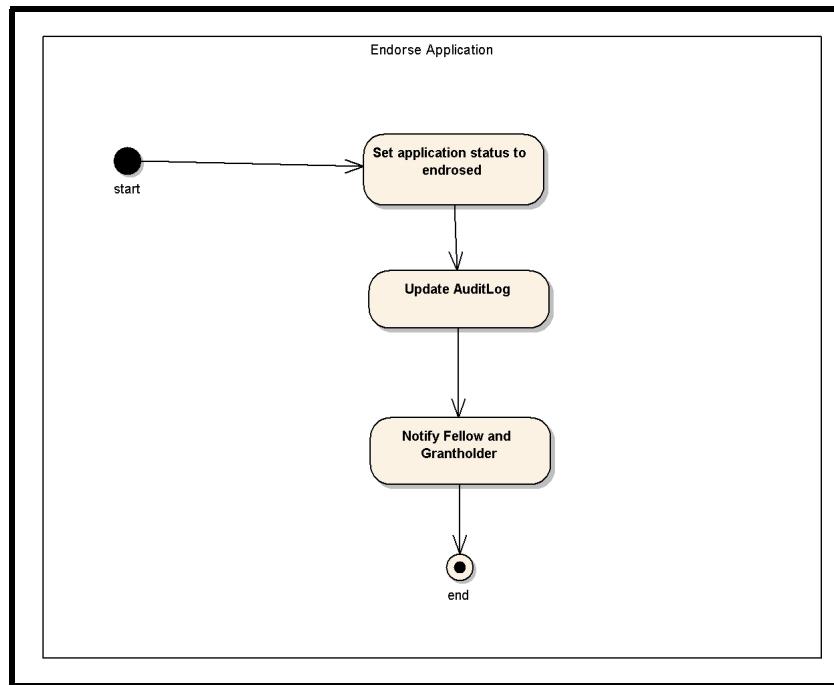


Figure 80: Activity diagram of the Endorse Application use case.

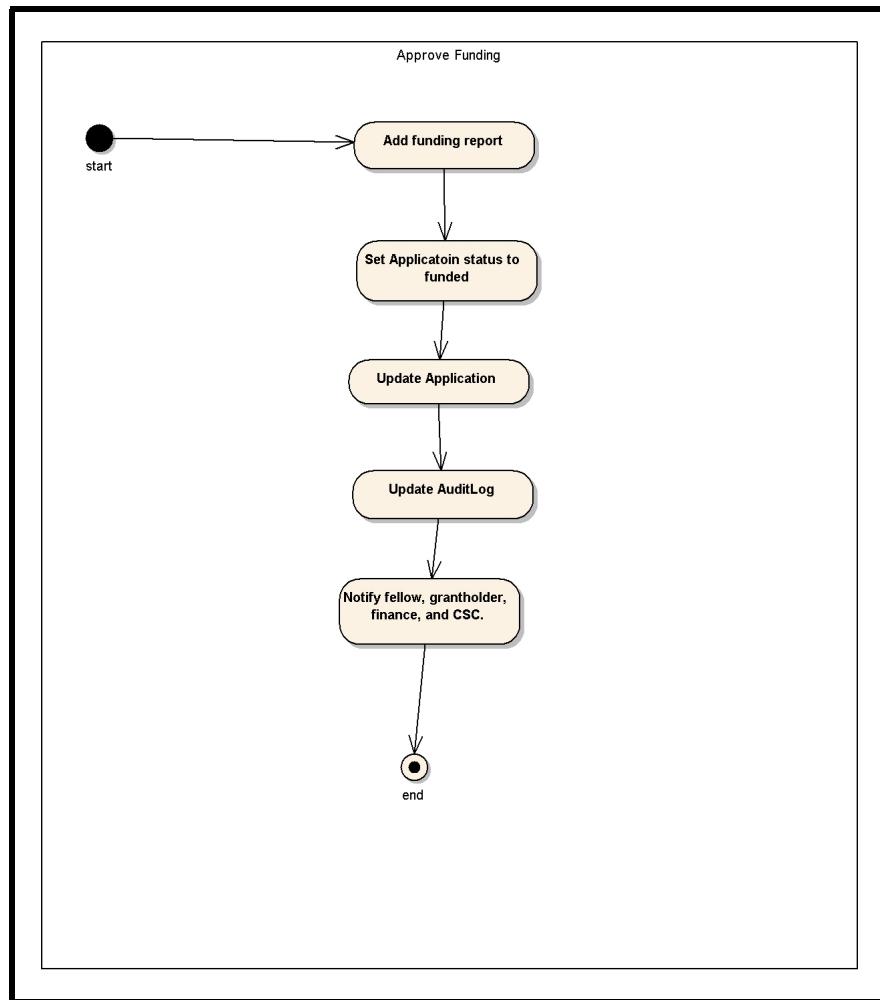


Figure 81: Activity diagram of the Approve Funding use case.

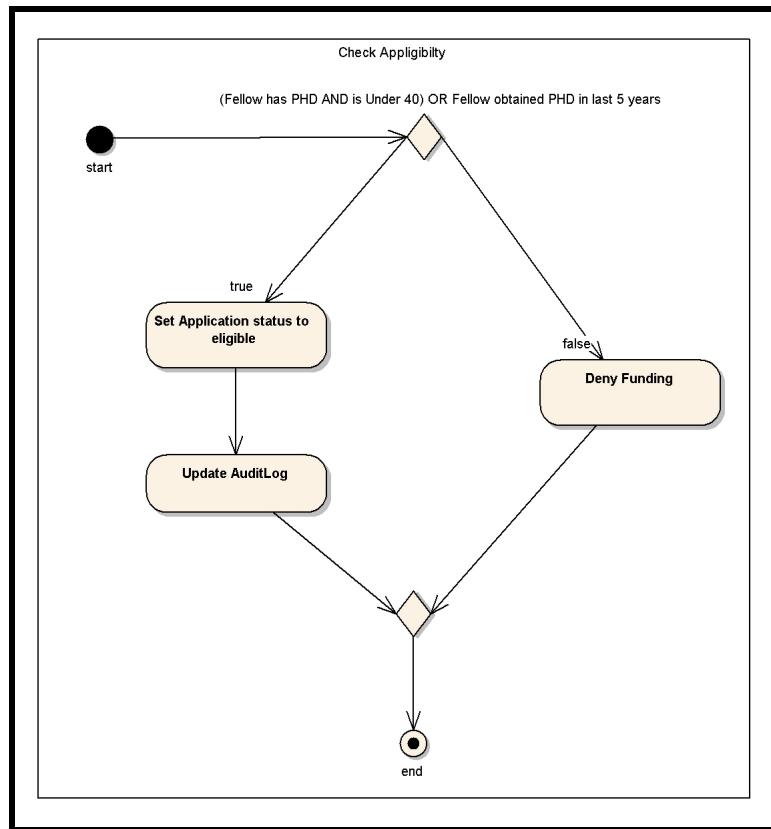


Figure 82: Activity diagram of the Check Applicability use case.

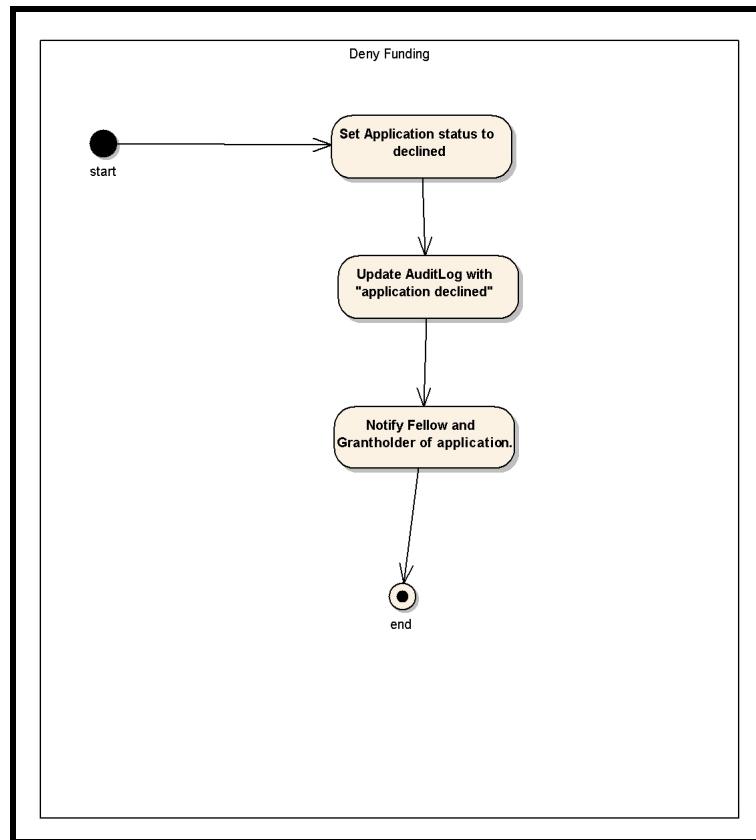


Figure 83: Activity diagram of the Deny Funding use case.

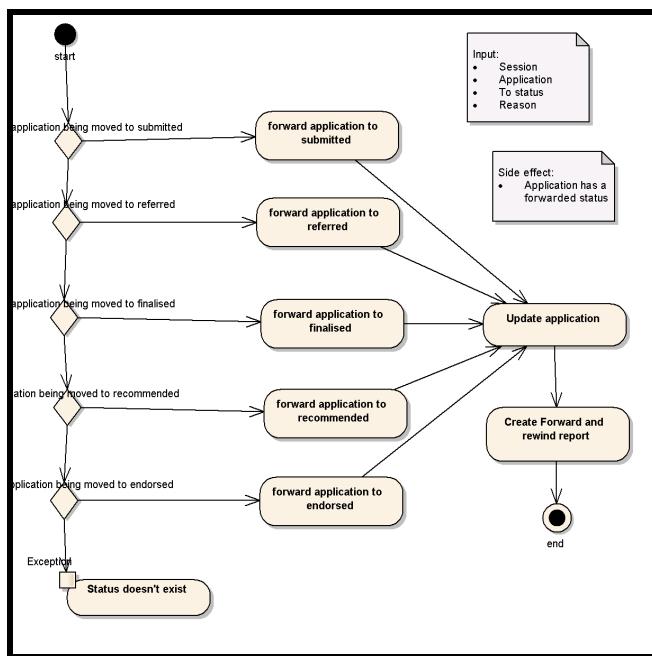


Figure 84: Activity diagram of the foward application use case.

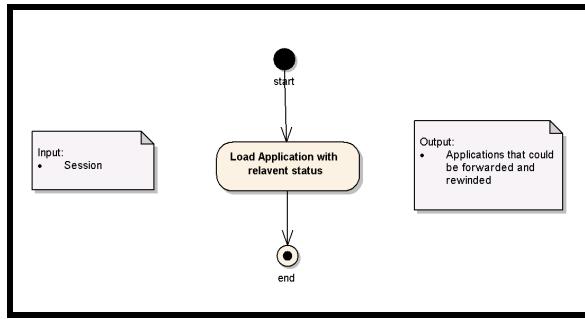


Figure 85: Activity diagram of the load movable applications use case.

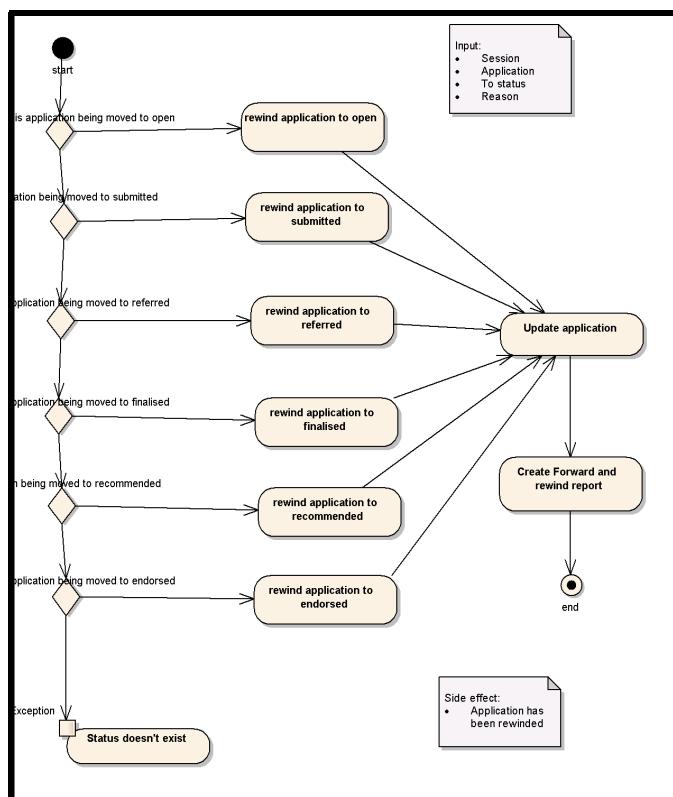


Figure 86: Activity diagram of the rewind application use case.

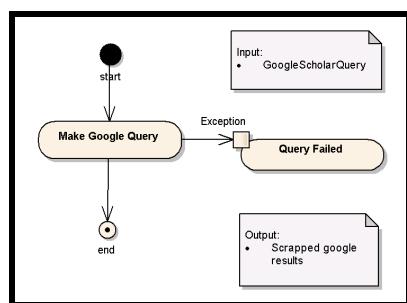


Figure 87: Activity diagram of the search Google Scholar use case.

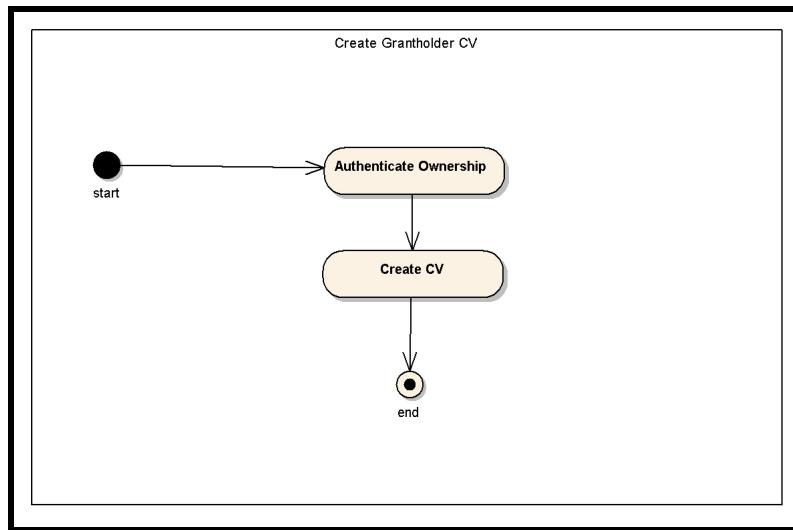


Figure 88: Activity diagram of the Create Grantholder CV use case.

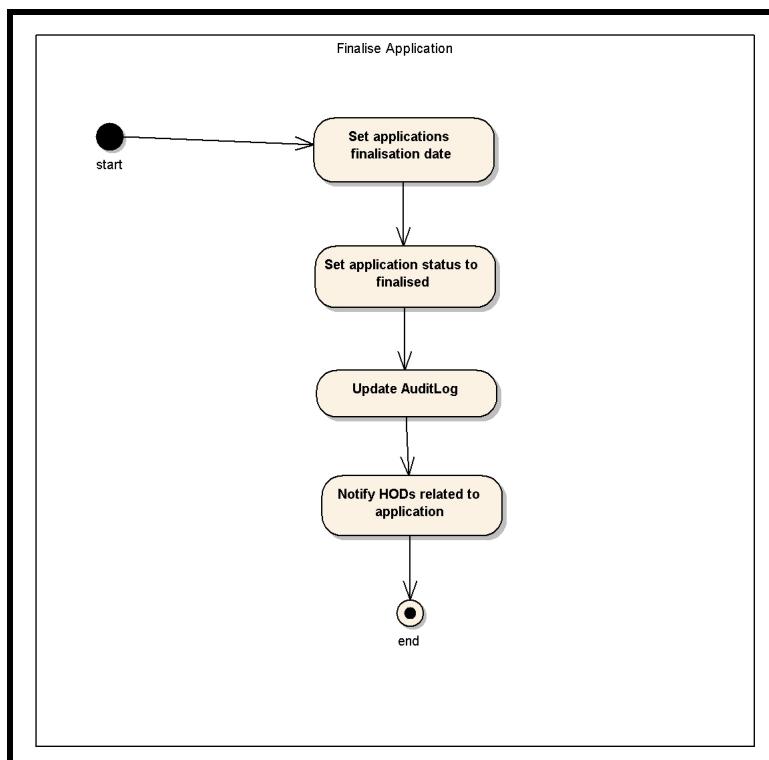


Figure 89: Activity diagram of the Finalise Application use case.

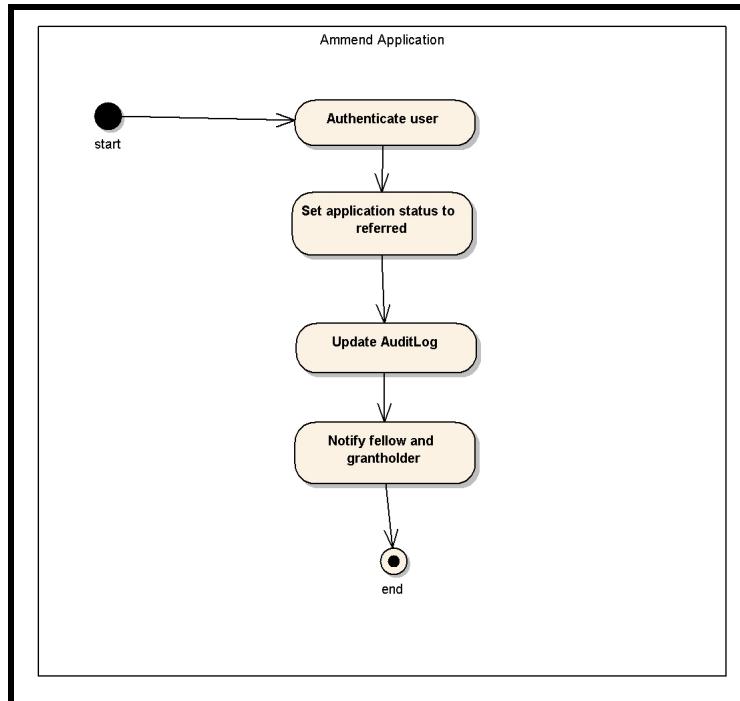


Figure 90: Activity diagram of the Ammend Application use case.

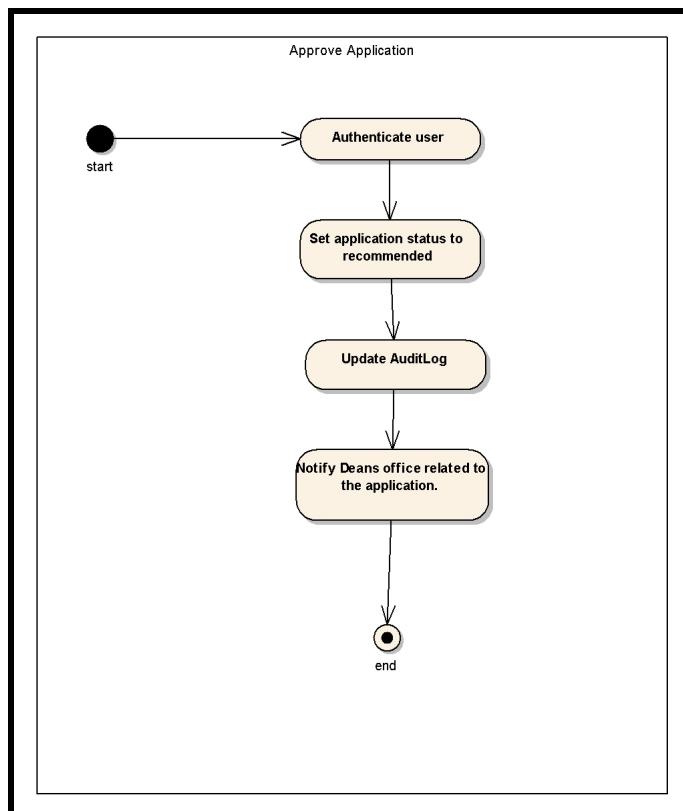


Figure 91: Activity diagram of the Approve Application use case.

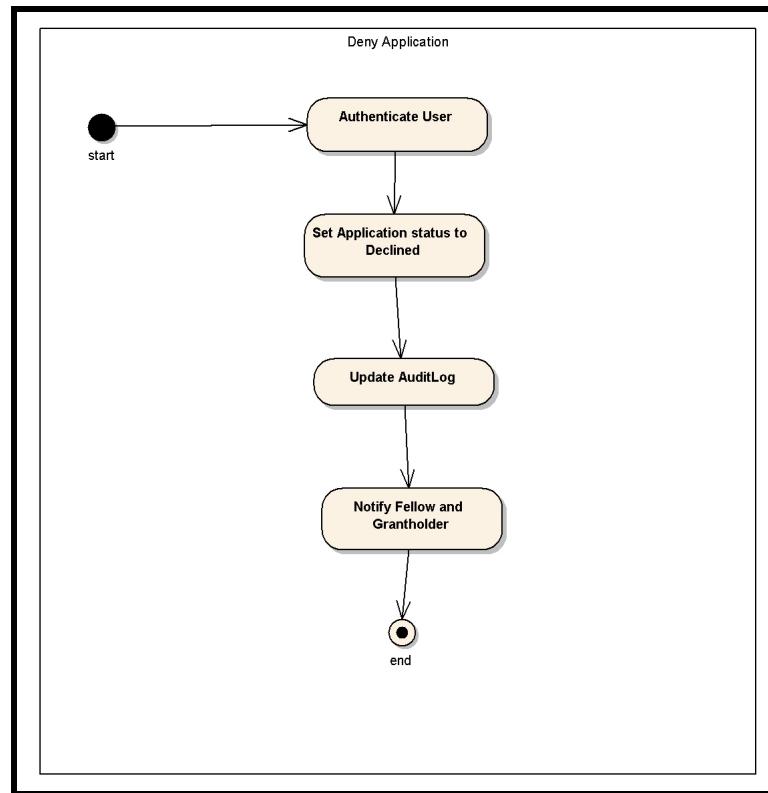


Figure 92: Activity diagram of the Deny Application use case.

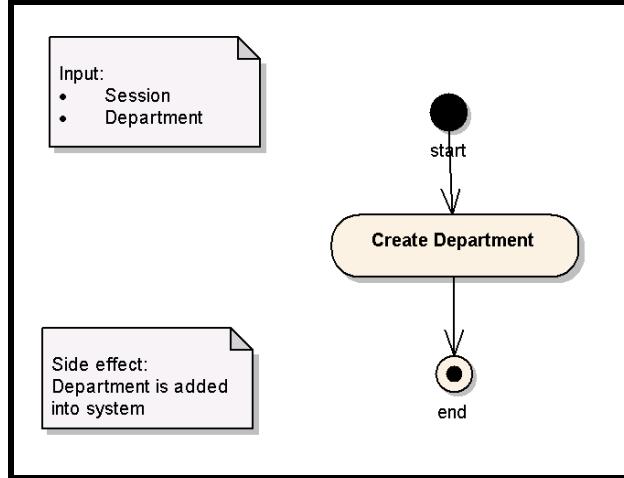


Figure 93: Activity diagram of the create department use case.

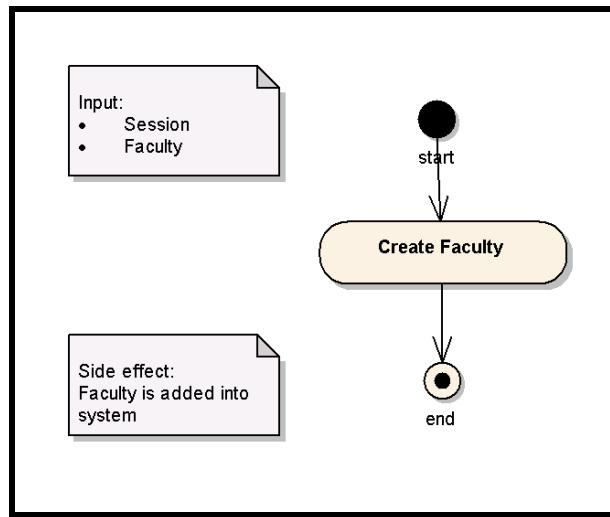


Figure 94: Activity diagram of the create faculty use case.

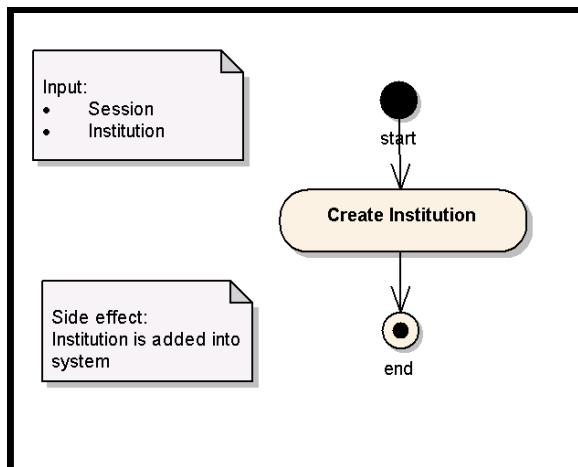


Figure 95: Activity diagram of the create institution use case.

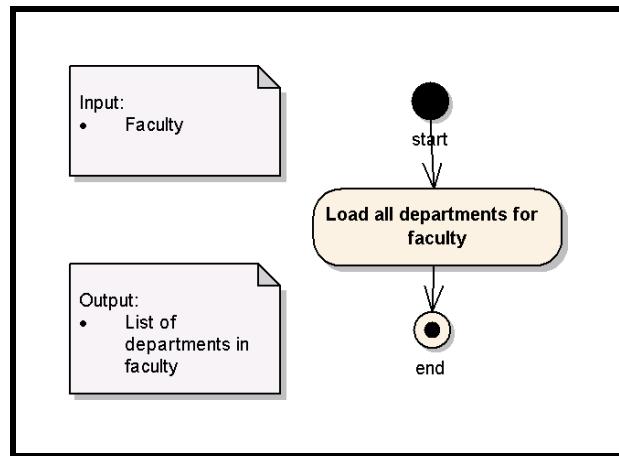


Figure 96: Activity diagram of the get all department for faculty use case.

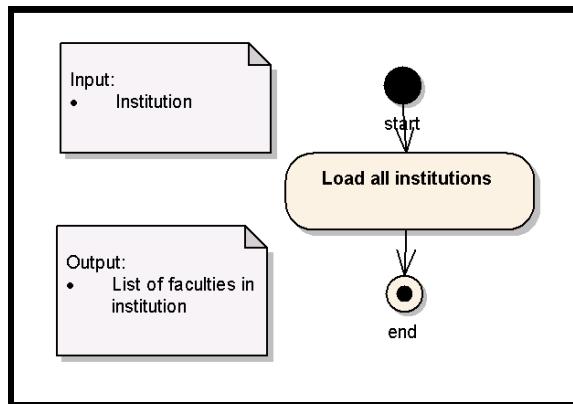


Figure 97: Activity diagram of the get all faculties institution use case.

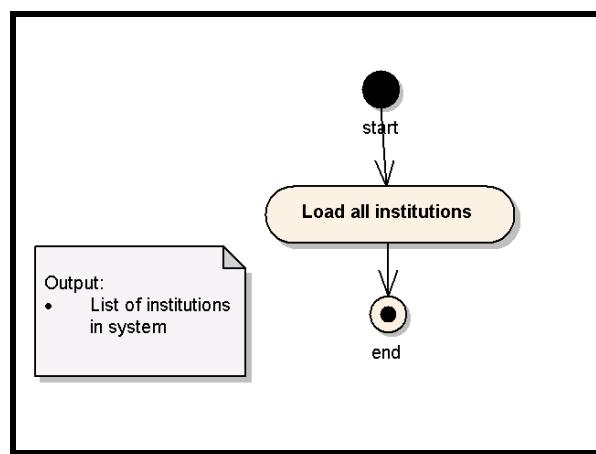


Figure 98: Activity diagram of the get all institutions use case.

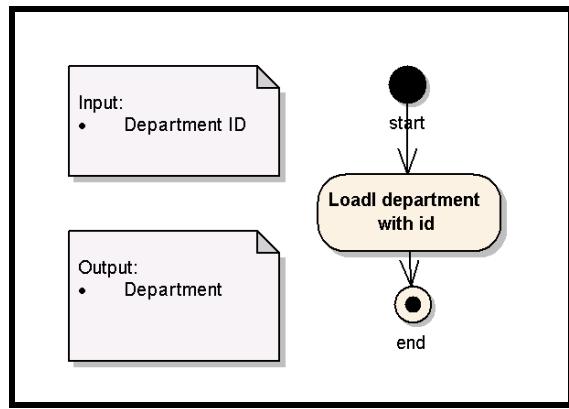


Figure 99: Activity diagram of the get department use case.

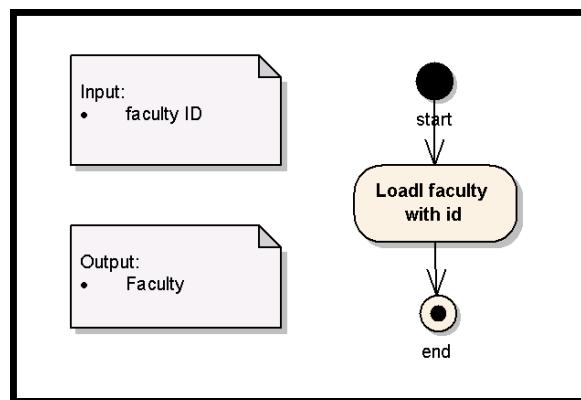


Figure 100: Activity diagram of the get faculty use case.

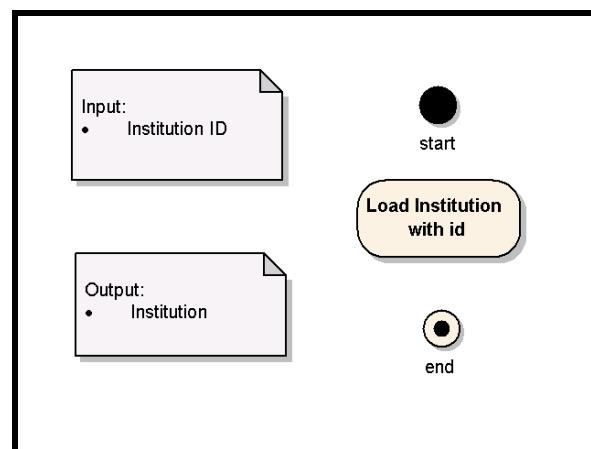


Figure 101: Activity diagram of the get institution use case.

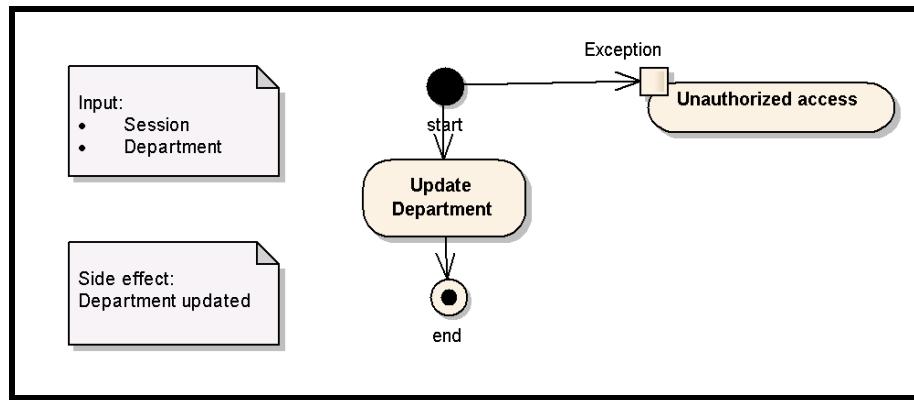


Figure 102: Activity diagram of the update department use case.

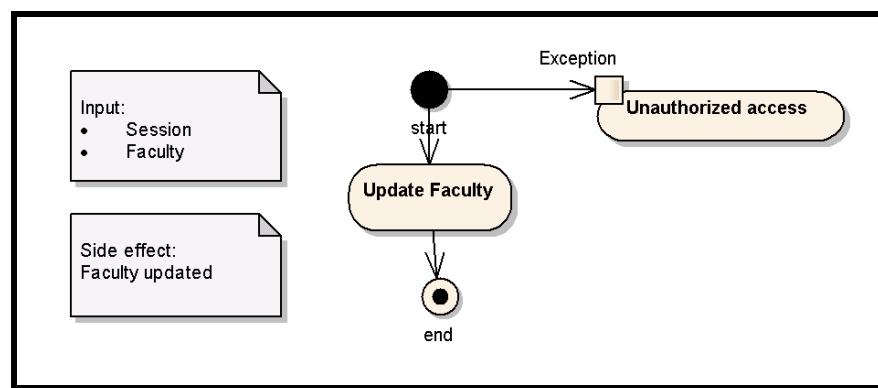


Figure 103: Activity diagram of the update faculty use case.

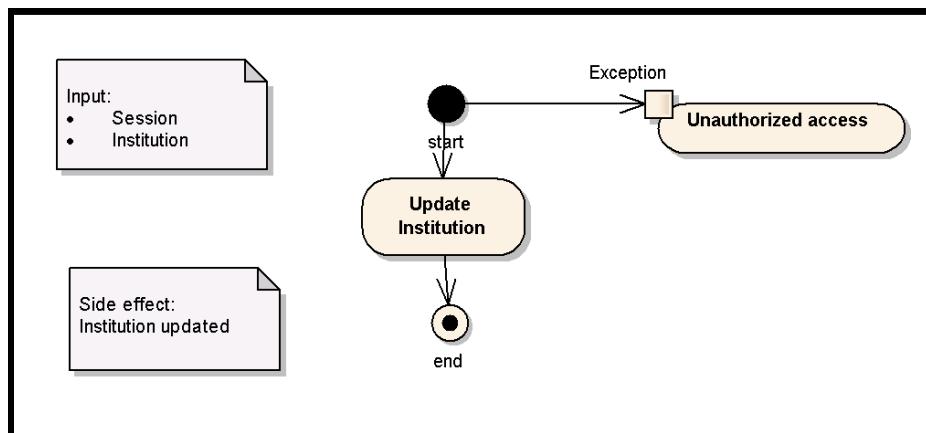


Figure 104: Activity diagram of the update institution use case.

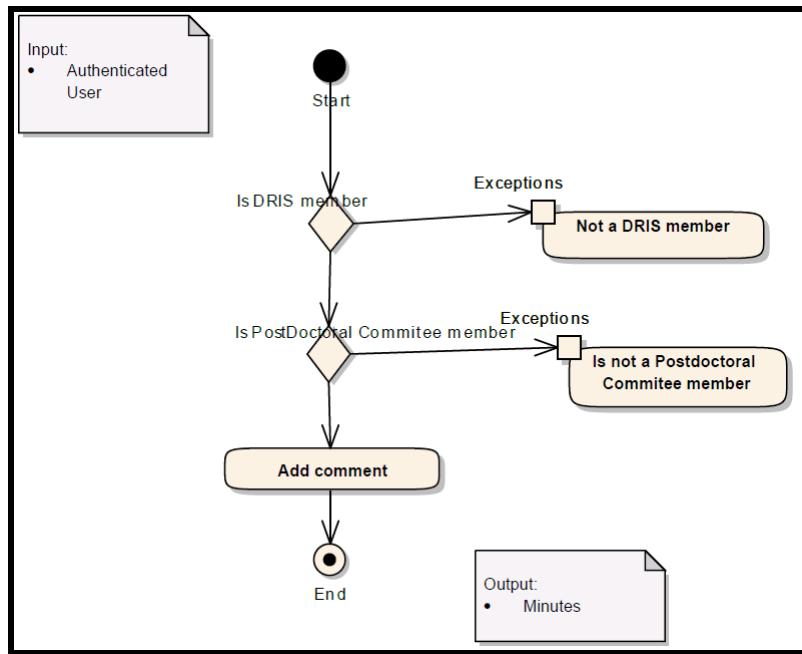


Figure 105: Activity diagram of the Add Minute Comment use case.

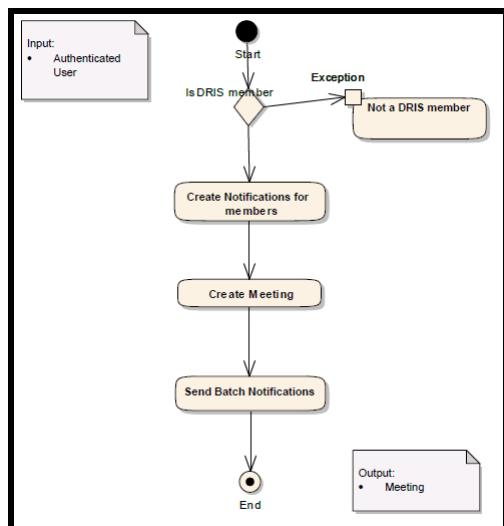


Figure 106: Activity diagram of the Create Meeting use case.

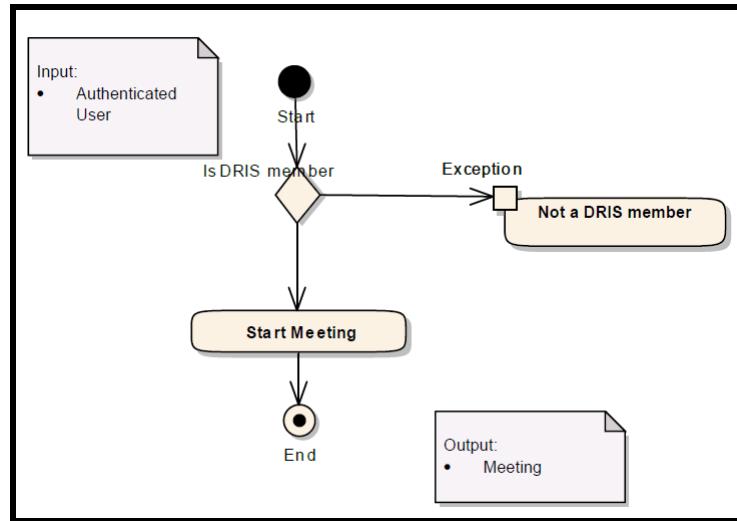


Figure 107: Activity diagram of the End Meeting use case.

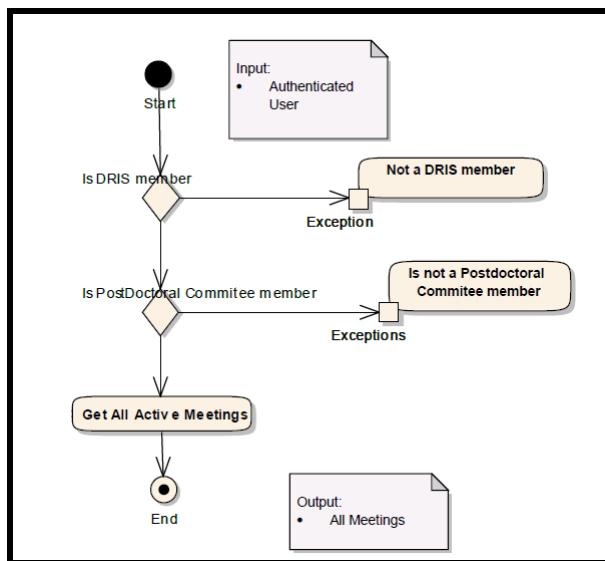


Figure 108: Activity diagram of the Get All Active Meetings use case.

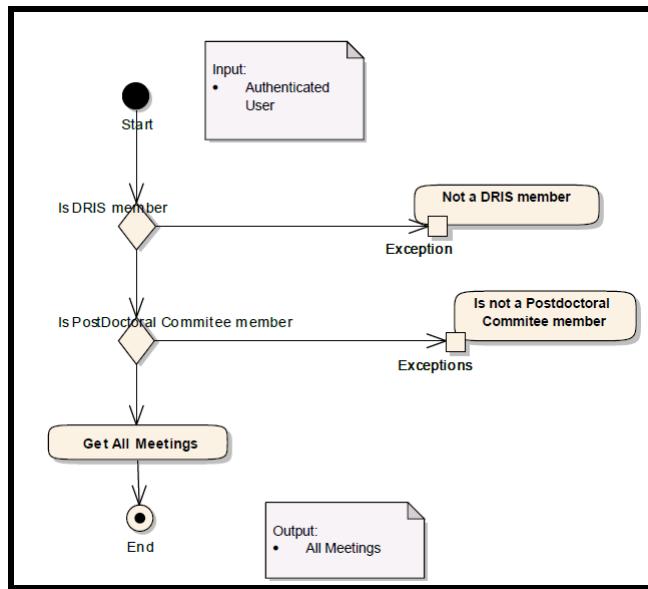


Figure 109: Activity diagram of the Get All Meetings use case.

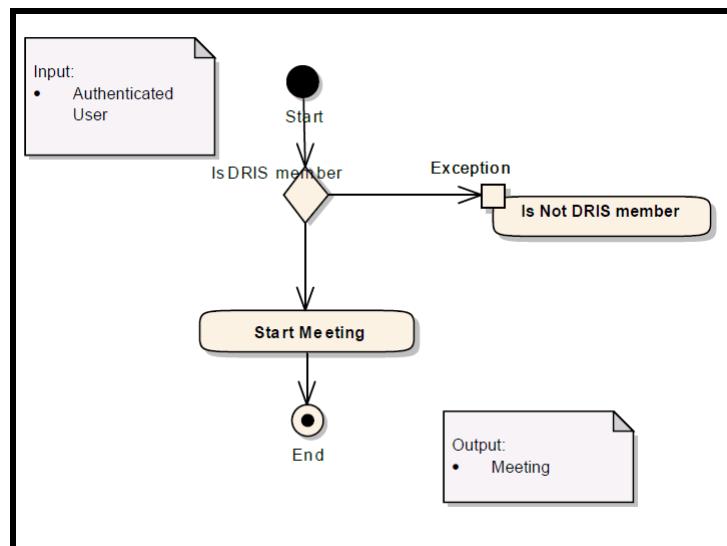


Figure 110: Activity diagram of the Start Meeting use case.

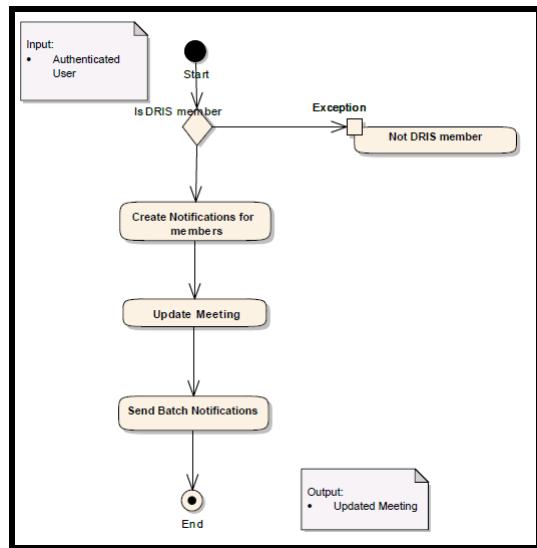


Figure 111: Activity diagram of the Update Meeting use case.

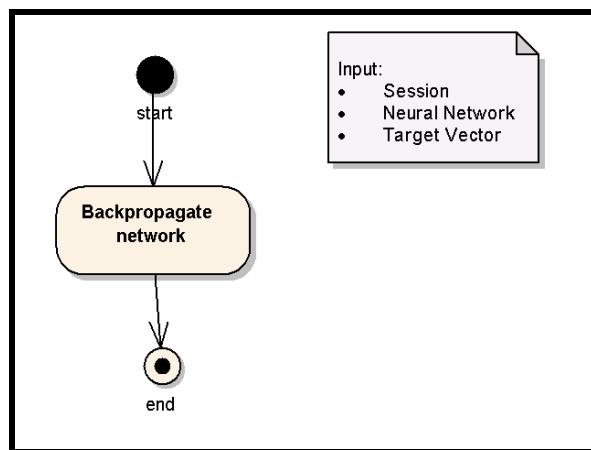


Figure 112: Activity diagram of the correctNeuralNetwork use case.

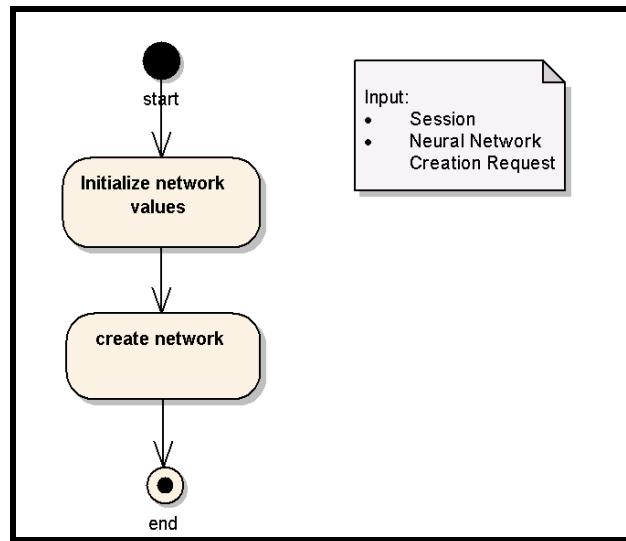


Figure 113: Activity diagram of the createNeuralNetwork use case.

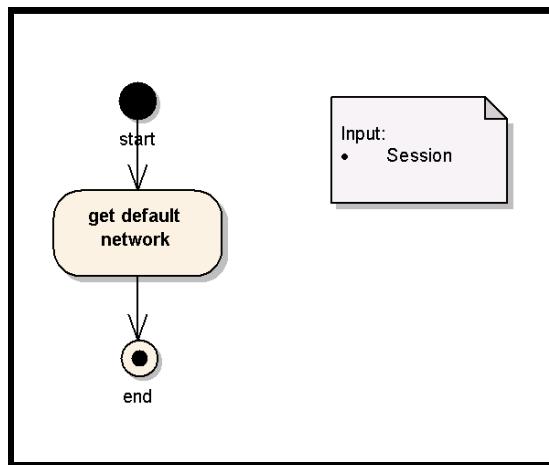


Figure 114: Activity diagram of the getDefaultNeuralNetwork use case.

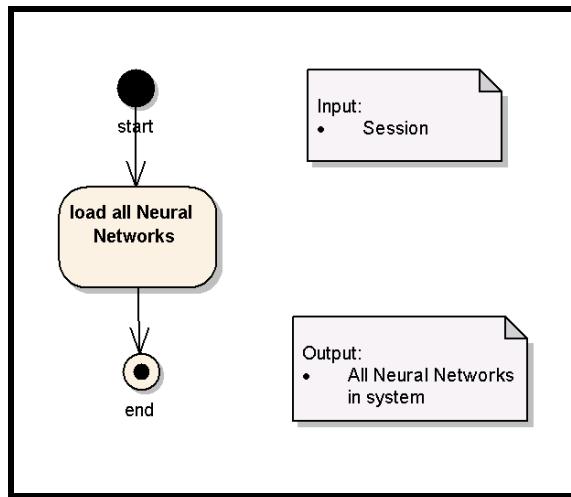


Figure 115: Activity diagram of the `loadAllNeuralNetworks` use case.

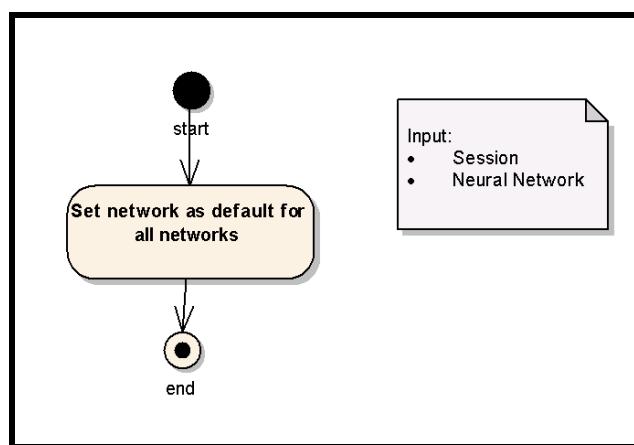


Figure 116: Activity diagram of the `makeNeuralNetworkDefaultNetwork` use case.

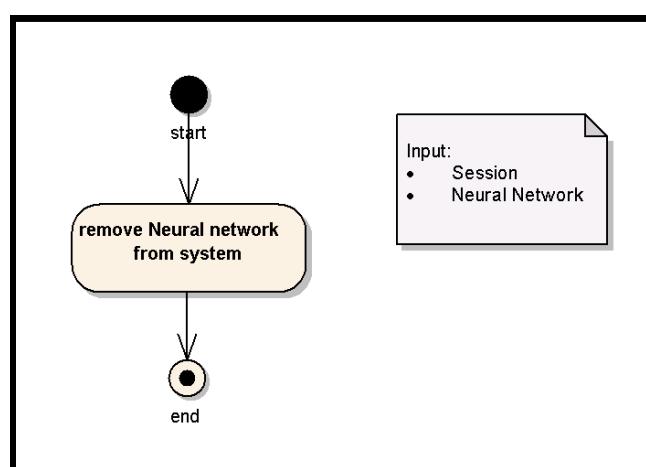


Figure 117: Activity diagram of the `removeNeuralNetwork` use case.

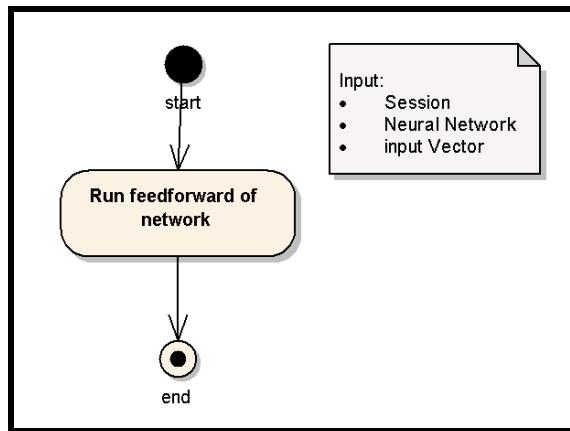


Figure 118: Activity diagram of the runNeuralNetwork use case.

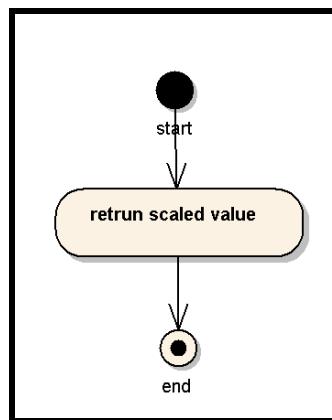


Figure 119: Activity diagram of the scaleValue use case.

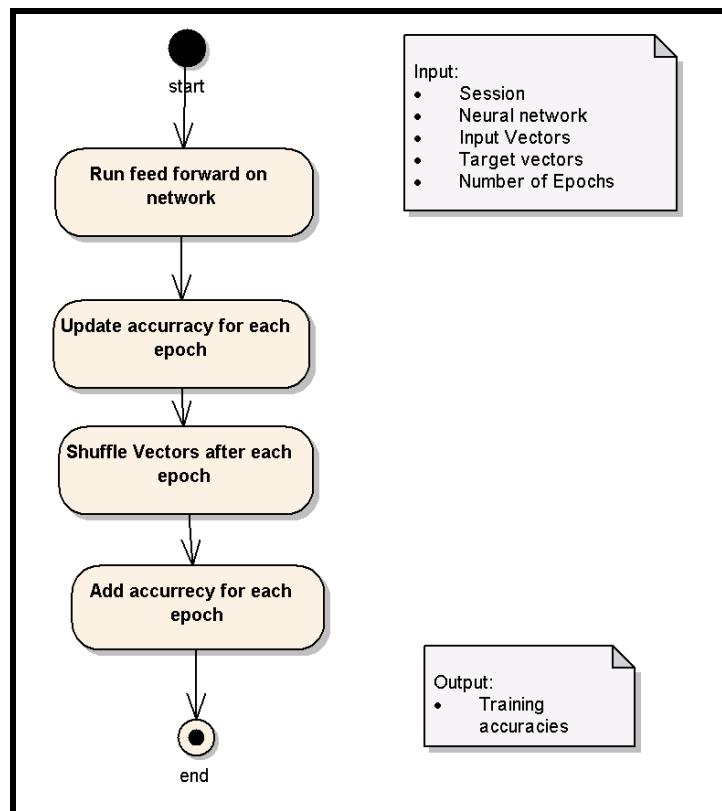


Figure 120: Activity diagram of the `trainNeuralNetwork` use case.

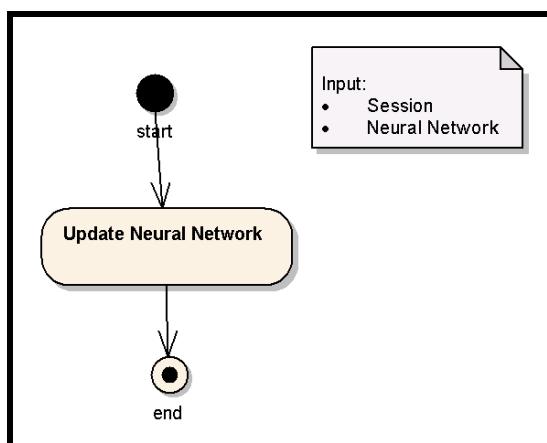


Figure 121: Activity diagram of the `updateNeuralNetwork` use case.

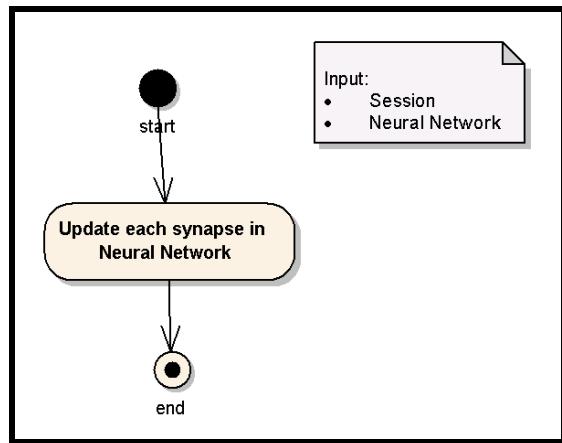


Figure 122: Activity diagram of the `updateNeuralNetworkSynapses` use case.

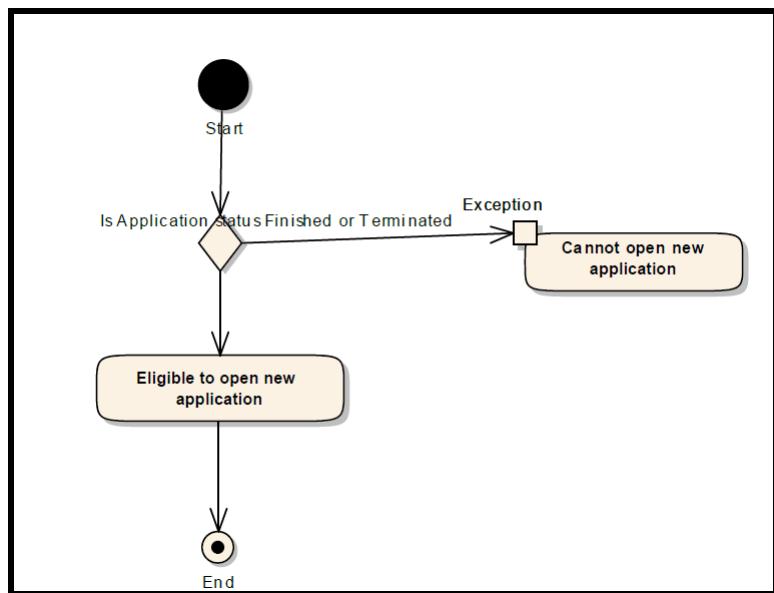


Figure 123: Activity diagram of the `Can the fellow open a New Application` use case.

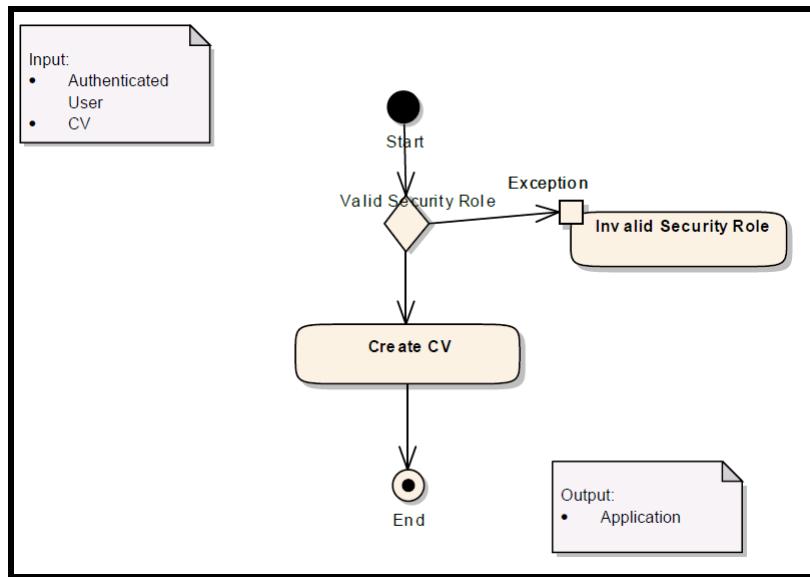


Figure 124: Activity diagram of the Create CV use case.

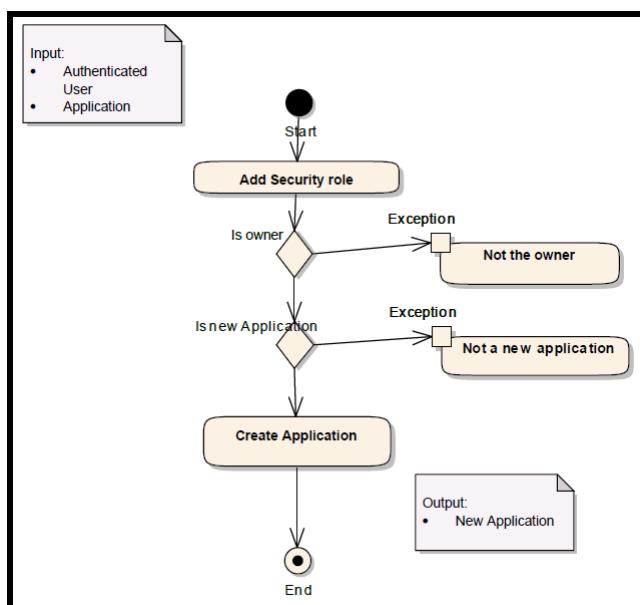


Figure 125: Activity diagram of the Create New Application use case.

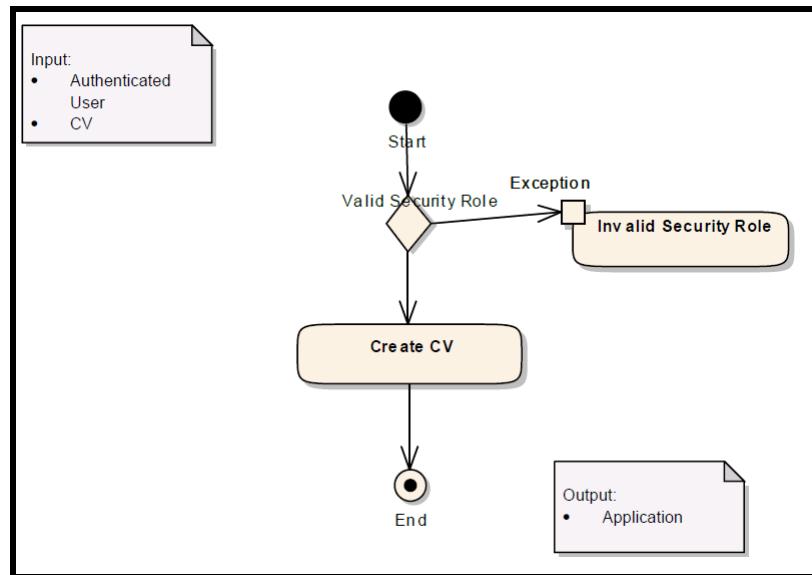


Figure 126: Activity diagram of the Get Open Application use case.

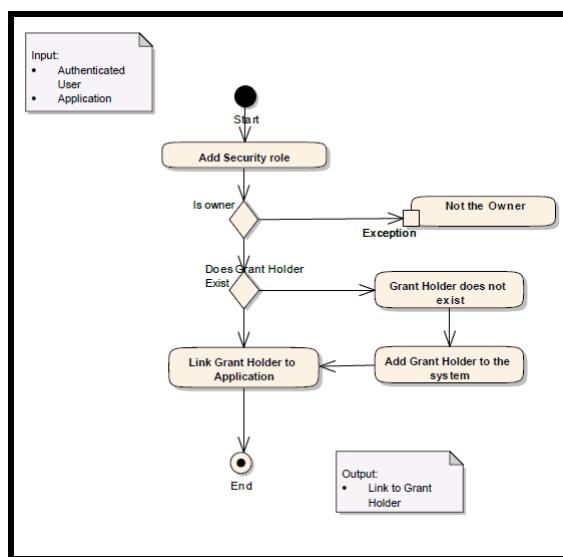


Figure 127: Activity diagram of the Link Grant Holder to Application use case.

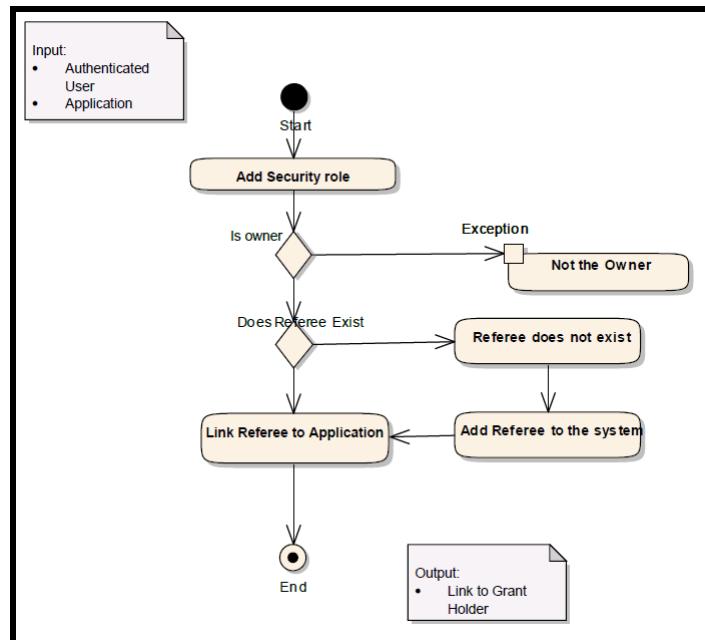


Figure 128: Activity diagram of the Link Referee to Application use case.

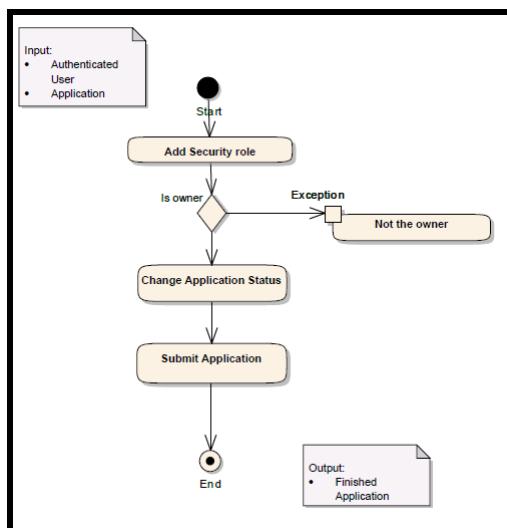


Figure 129: Activity diagram of the Submit Application use case.

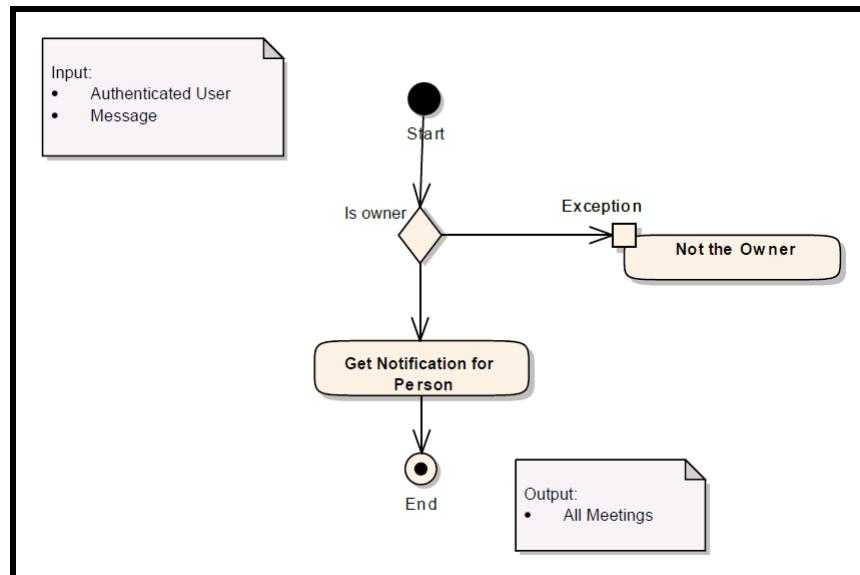


Figure 130: Activity diagram of the Get Notifications for Person use case.

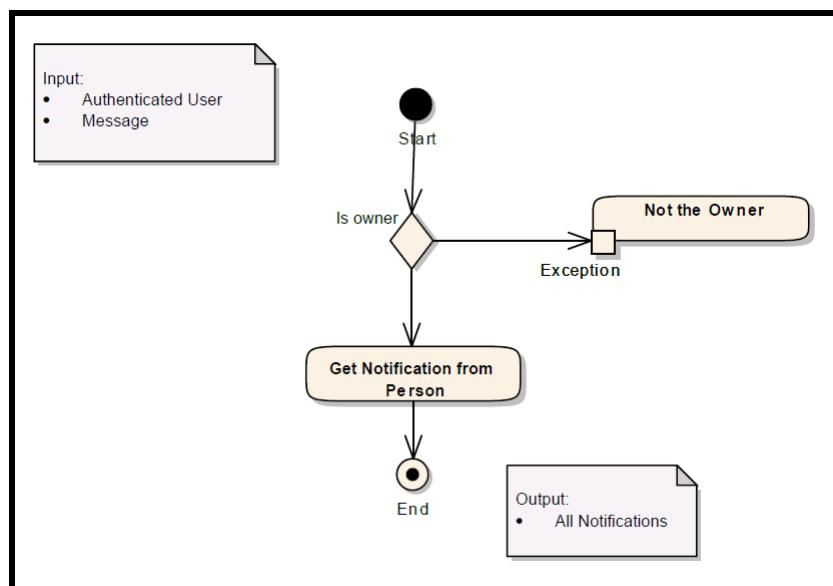


Figure 131: Activity diagram of the Get Notifications from Person use case.

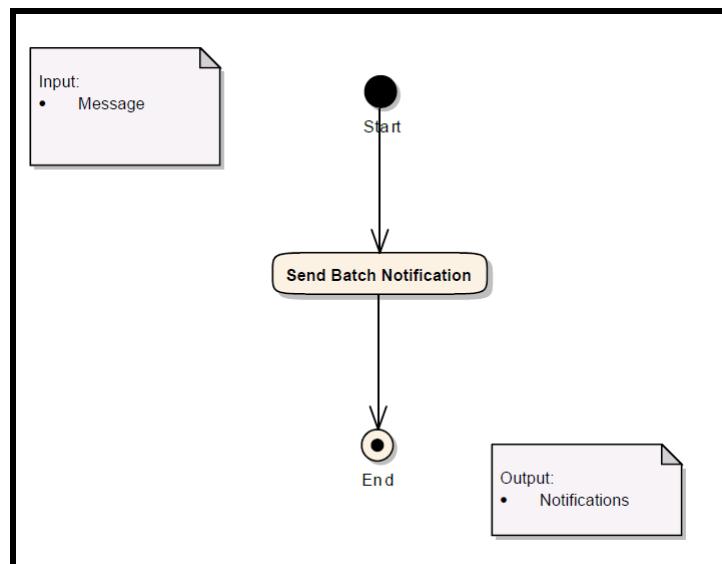


Figure 132: Activity diagram of the Send batch Notifications use case.

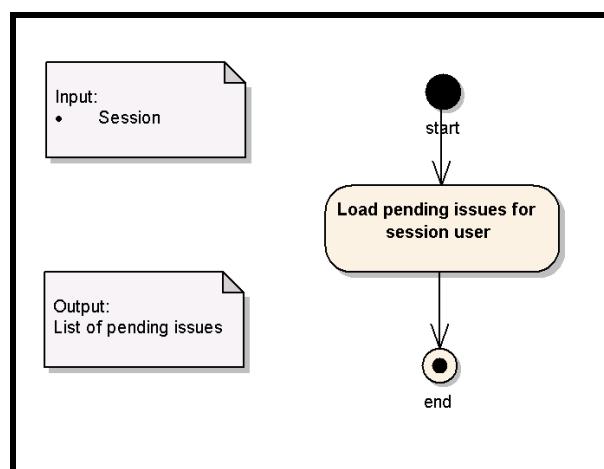


Figure 133: Activity diagram of the load all pending issues for session use case.

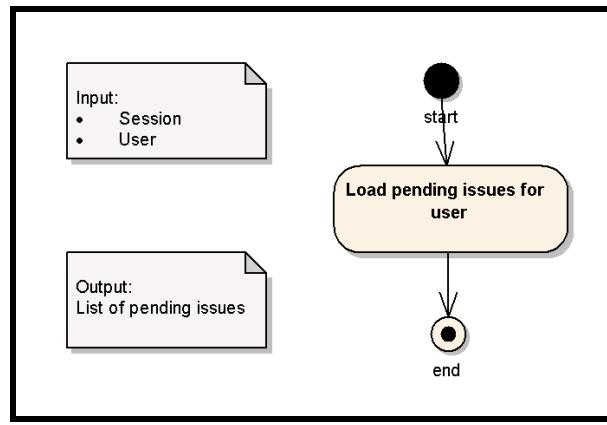


Figure 134: Activity diagram of the load all pending issues for user use case.

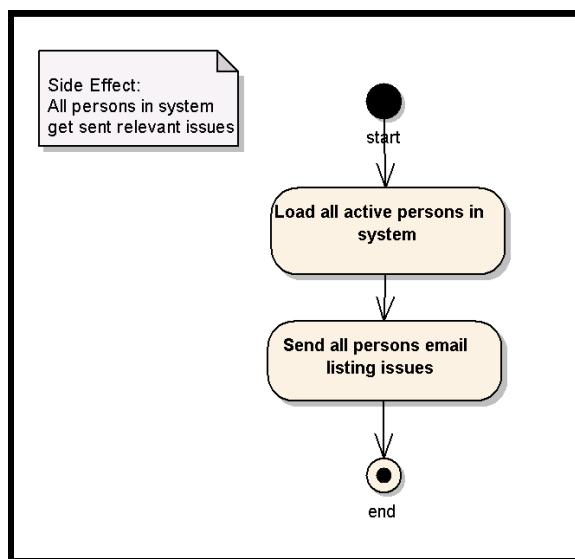


Figure 135: Activity diagram of the send daily reminders use case.

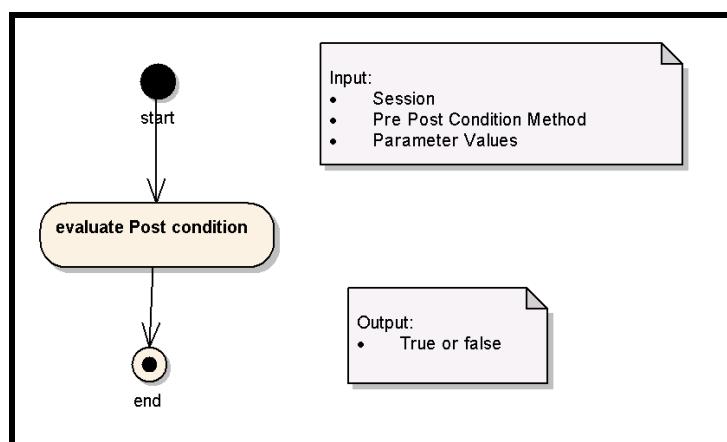


Figure 136: Activity diagram of the evaluatePostCondition use case.

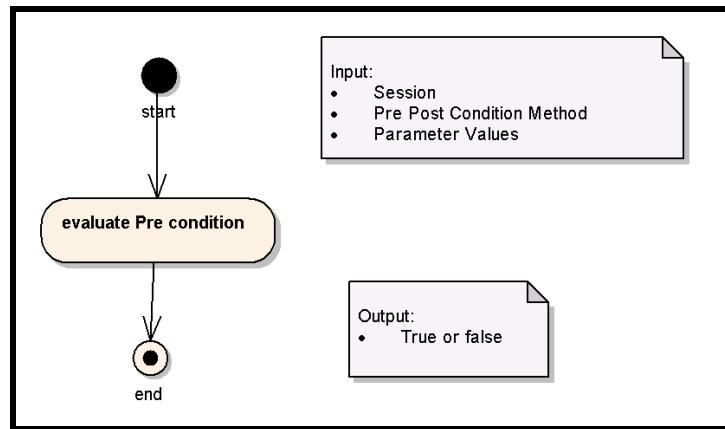


Figure 137: Activity diagram of the `evaluatePreCondition` use case.

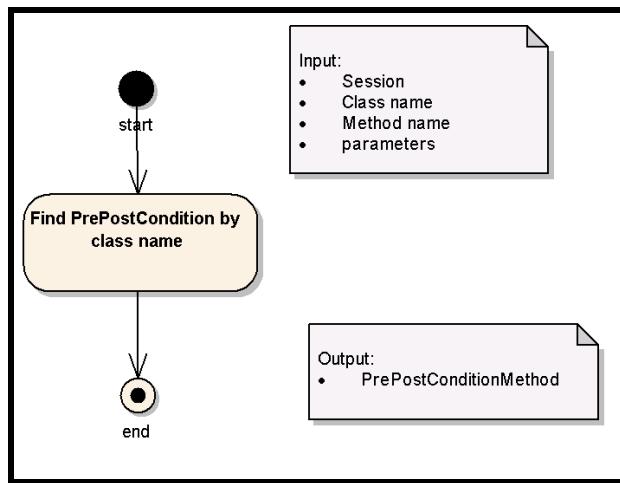


Figure 138: Activity diagram of the `findPrePostConditionalMethodByClassName` use case.

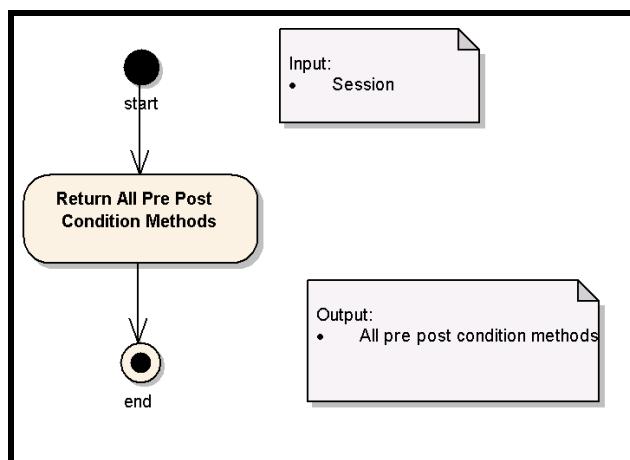


Figure 139: Activity diagram of the `loadPrePostConditionalMethods` use case.

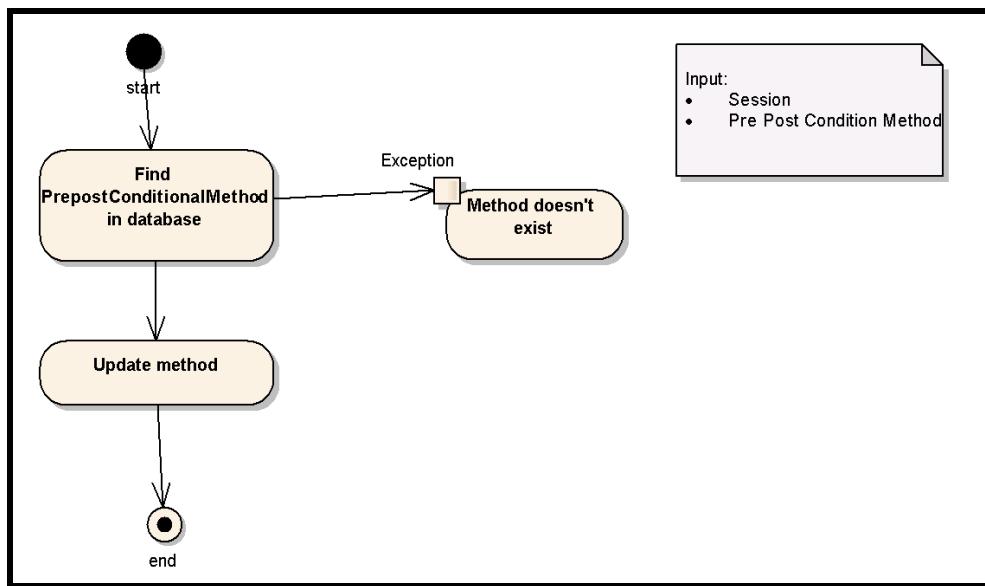


Figure 140: Activity diagram of the updatePrePostConditionalMethod use case.

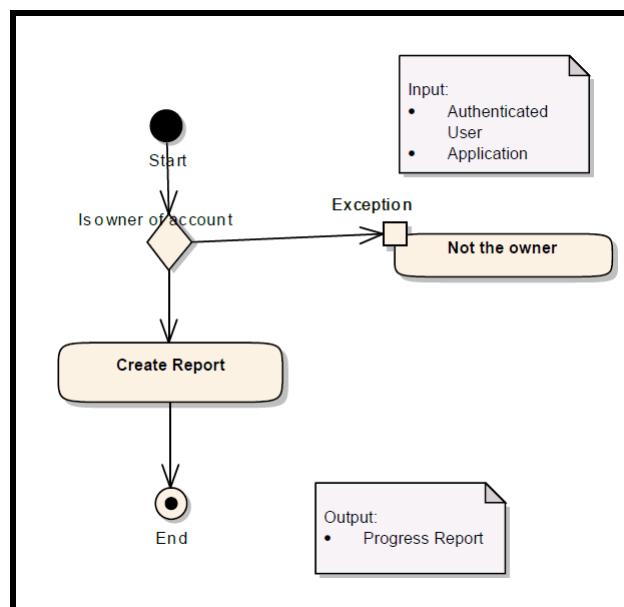


Figure 141: Activity diagram of the Create Progress Report use case.

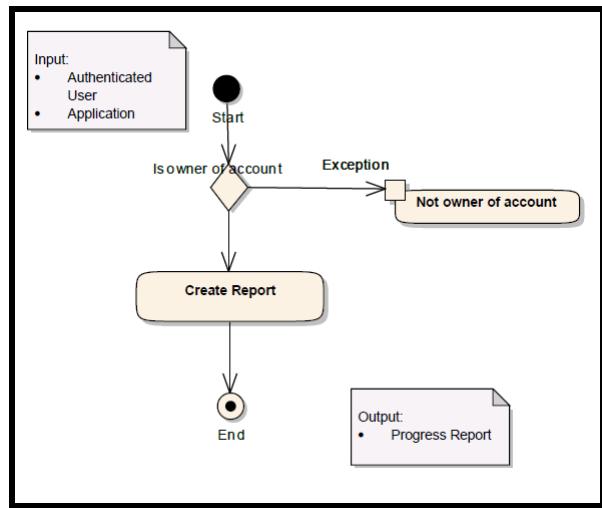


Figure 142: Activity diagram of the Update Progress Report use case.

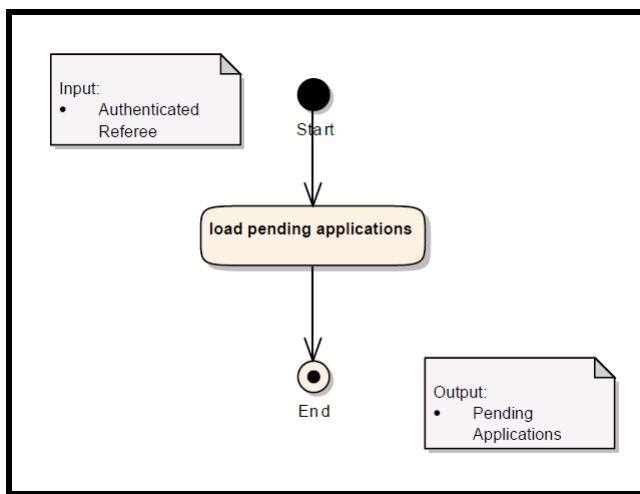


Figure 143: Activity diagram of the Create Referee Report use case.

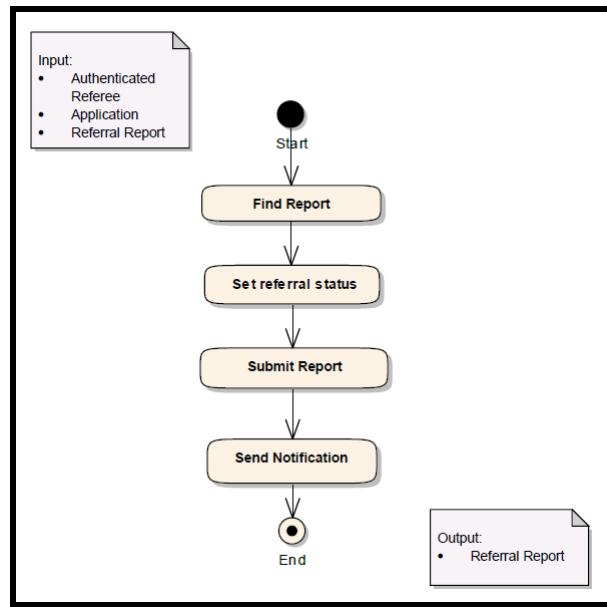


Figure 144: Activity diagram of the Submit Referee Report use case.

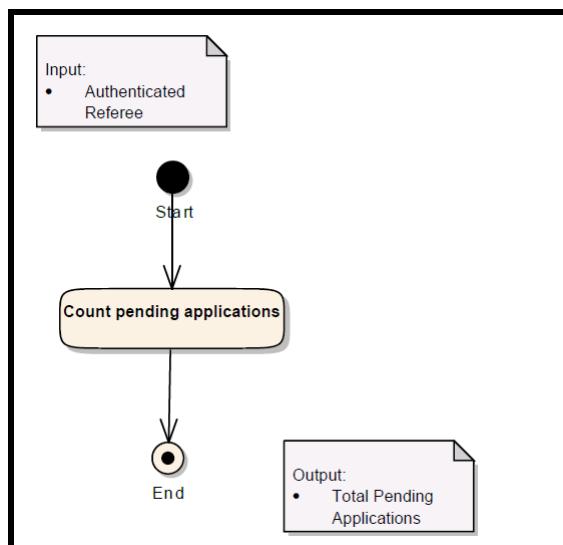


Figure 145: Activity diagram of the Count Pending Reports use case.

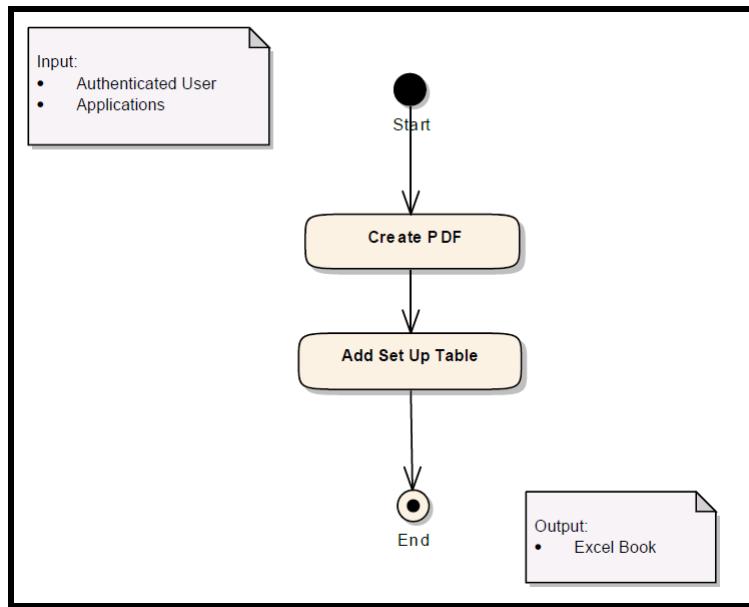


Figure 146: Activity diagram of the Export PDF Report use case.

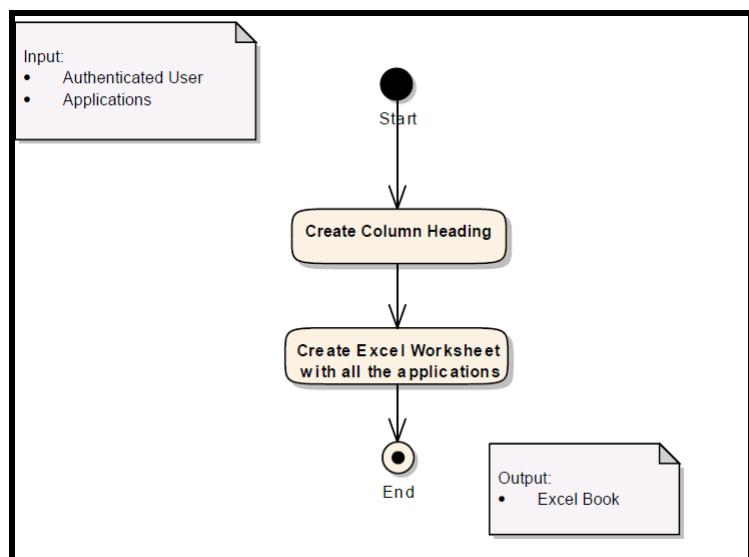


Figure 147: Activity diagram of the Export PDF Report use case.

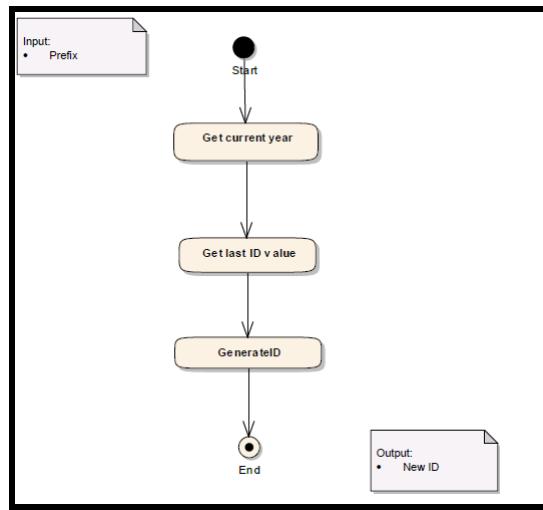


Figure 148: Activity diagram of the Generate System ID use case.

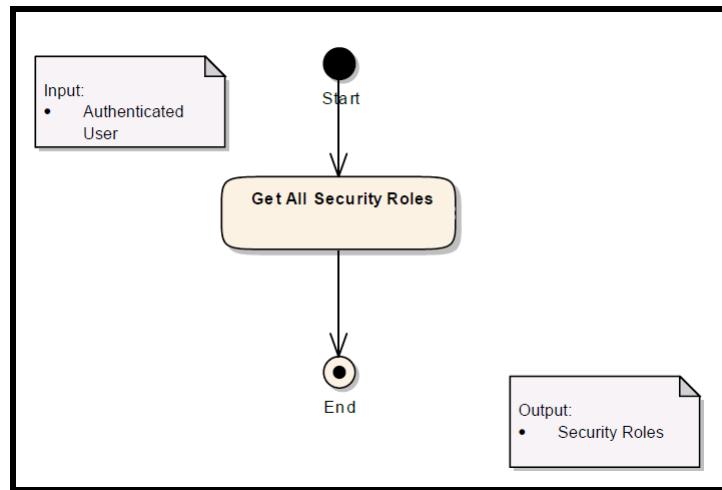


Figure 149: Activity diagram of the Get All Security Roles use case.

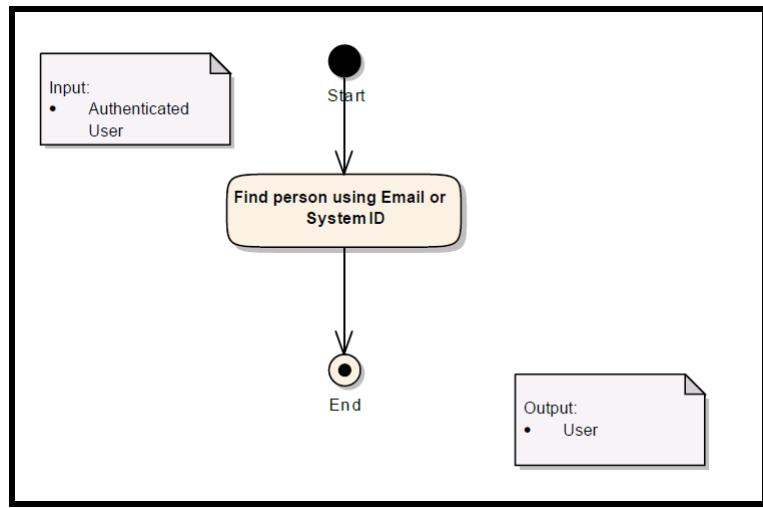


Figure 150: Activity diagram of the Get user by Email use case.

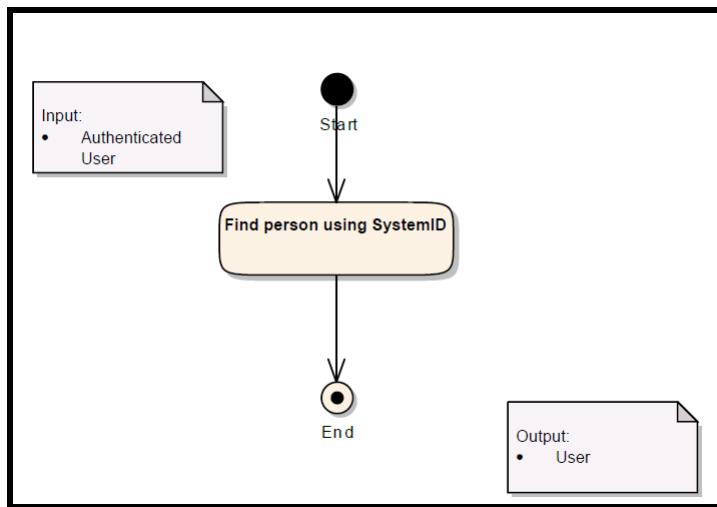


Figure 151: Activity diagram of the Get User by System ID use case.

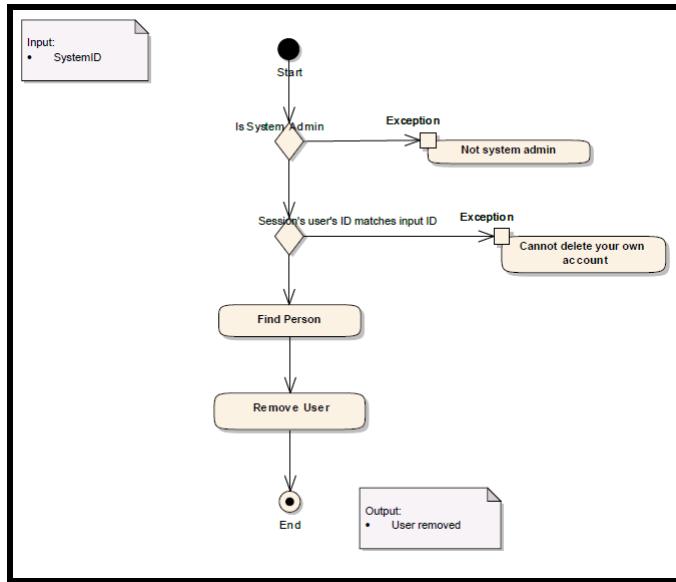


Figure 152: Activity diagram of the Remove User use case.

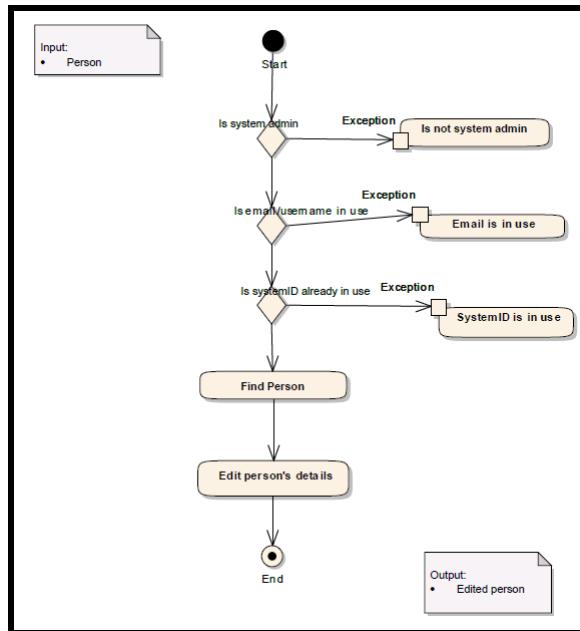


Figure 153: Activity diagram of the Update User use case.

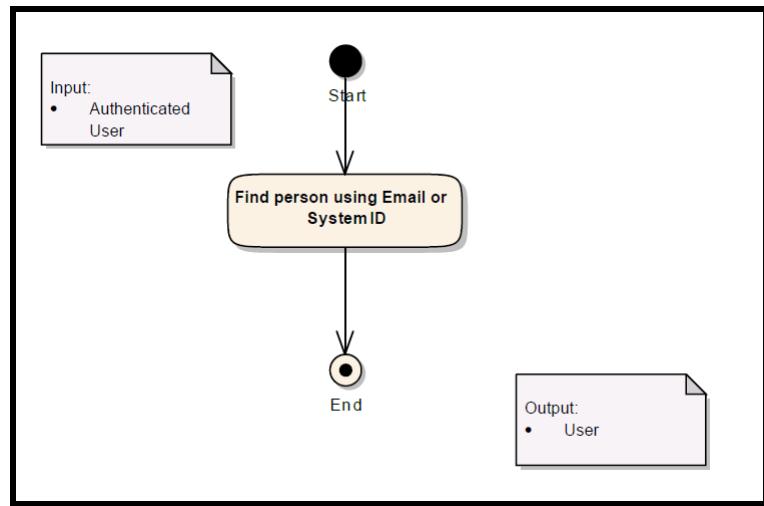


Figure 154: Activity diagram of the View All Users use case.

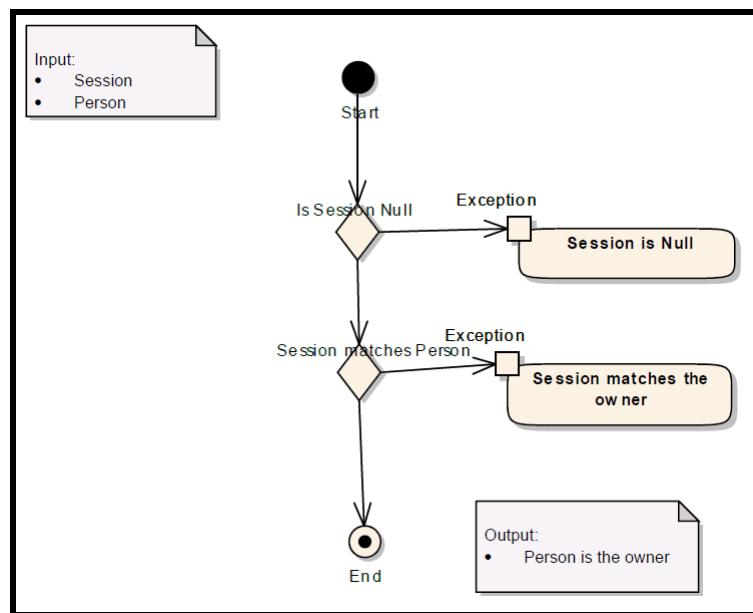


Figure 155: Activity diagram of the Authenticate User as Owner use case.

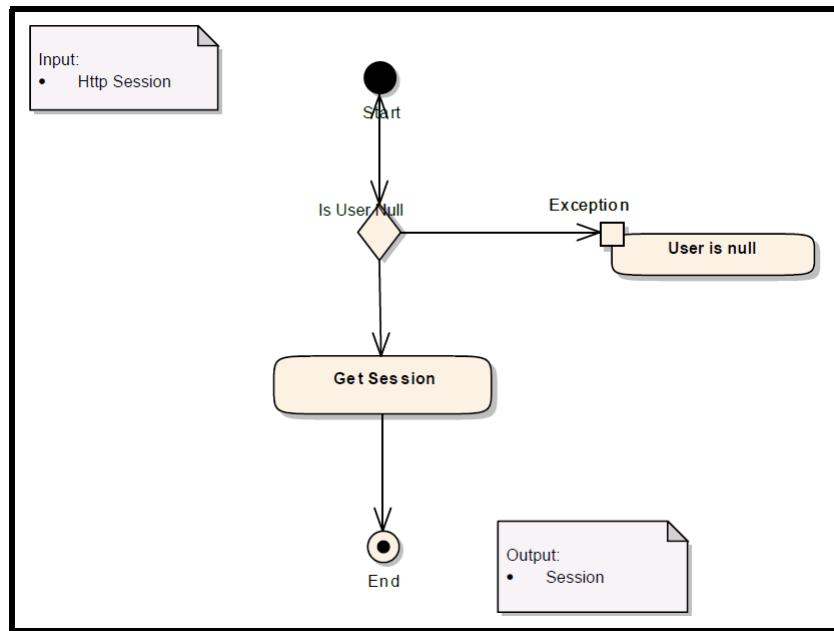


Figure 156: Activity diagram of the Get Http Session from Session use case.

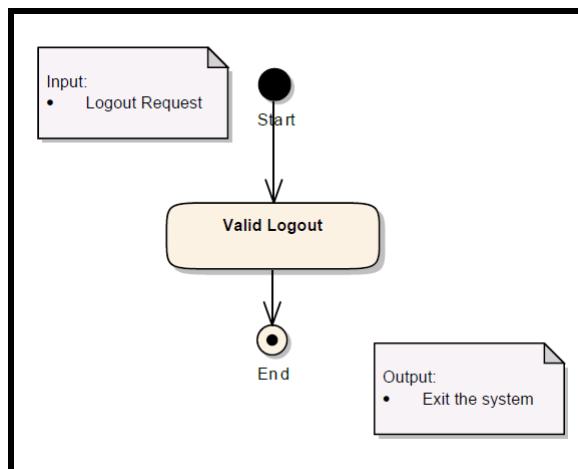


Figure 157: Activity diagram of the Logout use case.

# 4 Application Design

This section deals with actual design of the application itself and how the requirement will be implemented.

## 4.1 Overview

As seen from the various functional requirements above the application covers vast number of services and domain objects. Thus in order to break down and structure the application so that it doesn't become an complicate and unmaintainable system the application will make use of the Java EE reference framework which automatically implies a layered architectural pattern and provides many out of the box services. This is as discussed in the Software architectural document.

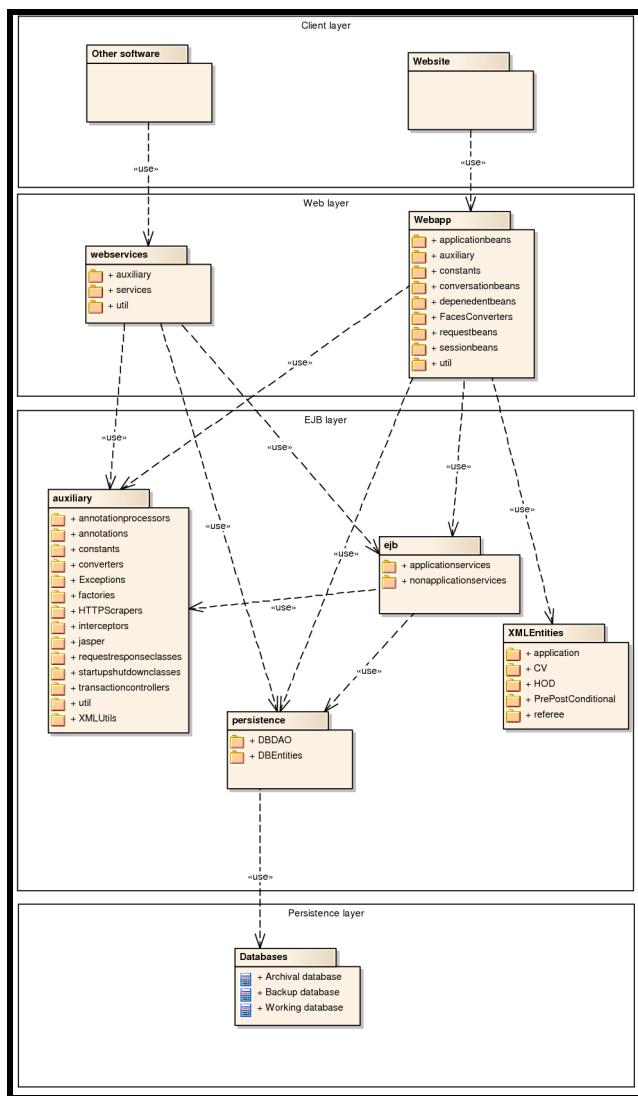


Figure 158: System overview

## 4.2 Client layer

### 4.2.1 Summary

The client layer provides a single web interface that makes use of the JSF facelet technology and Primefaces component kit in order to provide the a interactive, attractive and cohesive user interface. This web interface communicates with the JSF backing beans in the client layer through means of Java Expression language and Servelet technology as document in the Software Architecture document. Additionally this layer can be populated by other systems that can communicate other the internet using HTTPS by utilising the RESTful web services in the web layer. Some concept and template samples for the website design has been provided below:

### 4.2.2 Web page template/Style

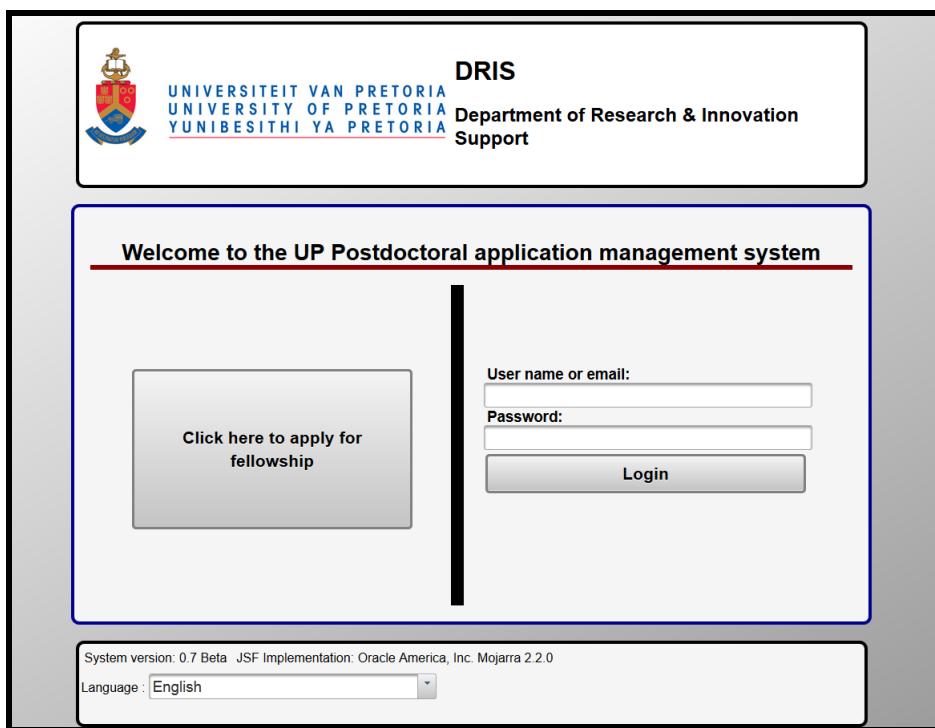


Figure 159: Website Concept Sample 1

**DRIS**  
UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA  
Department of Research & Innovation  
Support

Click to view bread crumbs

Portal Login > Prospective Fellow account creation view >

**Prospective fellow user account creation**

Please complete the fields below

**Person information**

Title:	Mr.
Full name:	John
Surname:	John
Email:	j
Password:	[redacted]
Re-Type Password:	[redacted]
Telephone number:	[redacted]
Work phone number:	[redacted]

**Error occurred!**  
Passwords do not match

Figure 160: Website Concept Sample 2

## 4.3 Client layer

### 4.3.1 Summary

The Client layer will provide JSF backing beans in order to process, validate and structure the input from the front-end web interface to the EJB layer and vice versa. The managed beans will also be used to provide the support for the front-end by using a variety of CDI scopes in order to retain important information and discard once off information:

- Application scope
- Session scope
- Conversation scope
- Request scope
- Dependent scope

Also provided by this layer are the RESTful web-services which in essence provide the same functionality as the JSF backing beans but for other systems or UIs. Further the Context and Dependency Injection (CDI) Technology will be used provide communication between managed beans and web services themselves and the EJBs.

## 4.4 EJB Layer

### 4.4.1 Summary

The EJB layer will provide the back bone for the business logic and capture all the core non UI-related logic as seen by the class diagrams of the various interfaces below. Majority of the functions provided by the EJB layer will be expecting domain objects as stipulated above and only in some rare instances will expect primitives or non-domain objects. All the EJBs are stateless and all provide a local interface. Also in order to perform user authentication and auditing custom interceptors designed for these purposes were designed as well as accompanying custom method annotations to specify parameters for audit able and secured methods by the EJBs. Also in order to conform to the layered architecture no communication with Persistence layer is allowed without using the EJB layer. Also since the system will make use of Restful API in order to integrate with other software this Layer needs to be completely secure.

### 4.4.2 Annotations

This section provides an overview of all the custom annotations used by the system.

- **AuditableMethod** - Used to indicate the method needs to be audited and what audit message should be used.
- **SecuredMethod** - Used to indicate the method needs authentication, the allowed security roles, if to test for ownership and what is the object to test ownership of.
- **PrePostConditionalMethod** - Used to indicate the method needs to be evaluated by the script in the database to see if it can proceed.

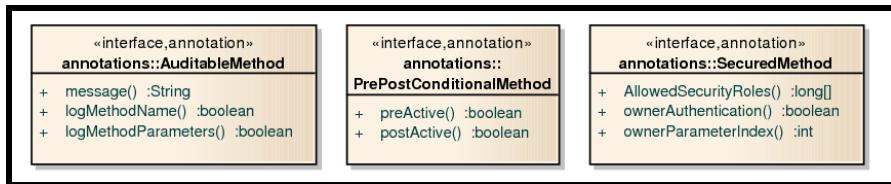


Figure 161: Annotation interfaces

#### 4.4.3 Class diagrams of interfaces

This section handles the



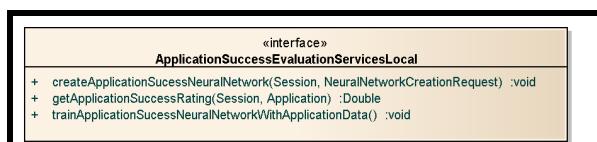
(a) Class diagram of interface for the Announcement Management service.



(b) Class diagram of interface for the Application Progress Viewer service.



(c) Class diagram of interface for the Application Renewal service.



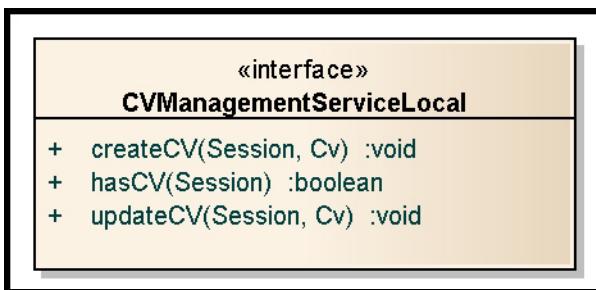
(d) Class diagram of interface for the Application Success Evaluation service.



(e) Class diagram of interface for the Archival service.



(f) Class diagram of interface for the Archival service.



(g) Class diagram of interface for the CV Management service.



(h) Class diagram of interface for the Deans Endorsement service.



(a) Class diagram of interface for the DRIS Approval service.



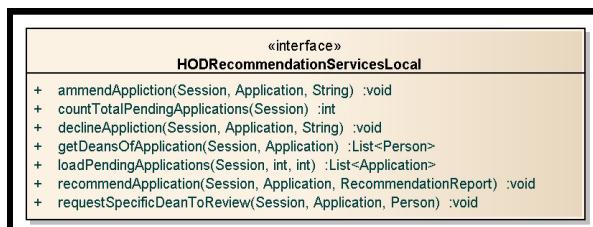
(b) Class diagram of interface for the Forward and Rewind service.



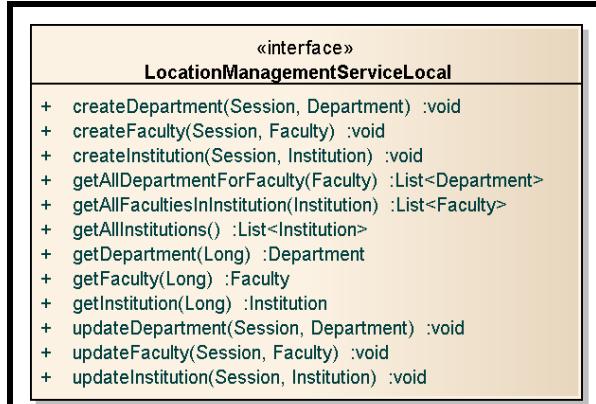
(c) Class diagram of interface for the Google Scholar service.



(d) Class diagram of interface for the Grant Holder Finalisation service.



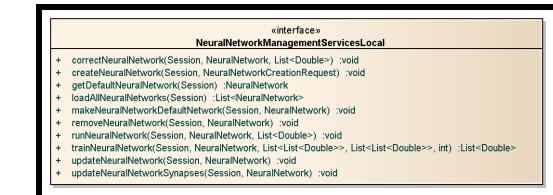
(e) Class diagram of interface for the HOD Recommendation service.



(f) Class diagram of interface for the Location Management service.



(g) Class diagram of interface for the Meeting Management service.



(h) Class diagram of interface for the Neural Network Management service.



(a) Class diagram of interface for the New Application service.



(b) Class diagram of interface for the Notification service.



(c) Class diagram of interface for the Notifier service.



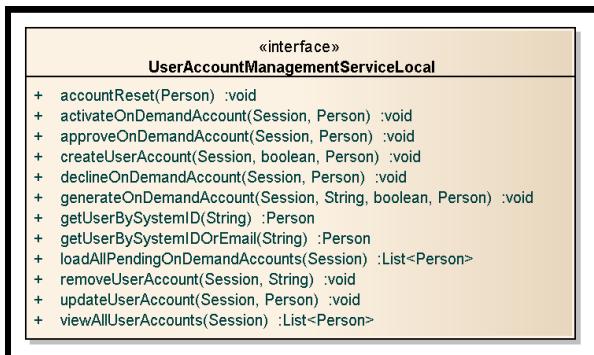
(d) Class diagram of interface for the Pre Post Conditional Management service.



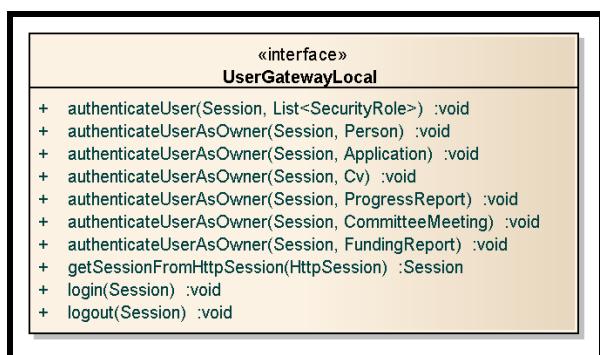
(e) Class diagram of interface for the Progress Report Management service.



(f) Class diagram of interface for the Referee Report service.



(g) Class diagram of interface for the User Account Management service.



(h) Class diagram of interface for the User Gateway service.

## 4.5 Persistence Layer

### 4.5.1 Summary

The persistence layer is made up of three Relation MySQL databases each of which has precisely the same entity structure, as specified in the ERD diagram. These entities are the technological specific mapping of the persistence domain objects found above. The databases are as follows:

- Working database - Used to store all the current data used by the application.
- Backup database - Used to store the latest full backup of the working database.
- Archive database - Used to store any archived data that the application identified and moved from the working database.

Ideally these databases should be stored on separate database services to improve redundancy and fault tolerance. Though in the current situation this is not possible due to financial constraints.

In order to convert the relation entities to object representation for the EJB layer, the Eclipse Top-link JPA object relational mapper is used. Further to efficiently use storage space and increase maintainability of the large text data in the database certain text sections are in XML format. This string of text is converted to and from Objects using JAXB technology. As stipulated in the Software architecture document this the RDBMS can be replaced with ObjectDB if financial support is provided this would increase the performance, maintainability and cohesiveness of the data structures in the persistence layer.

#### 4.5.2 ERD diagram of database

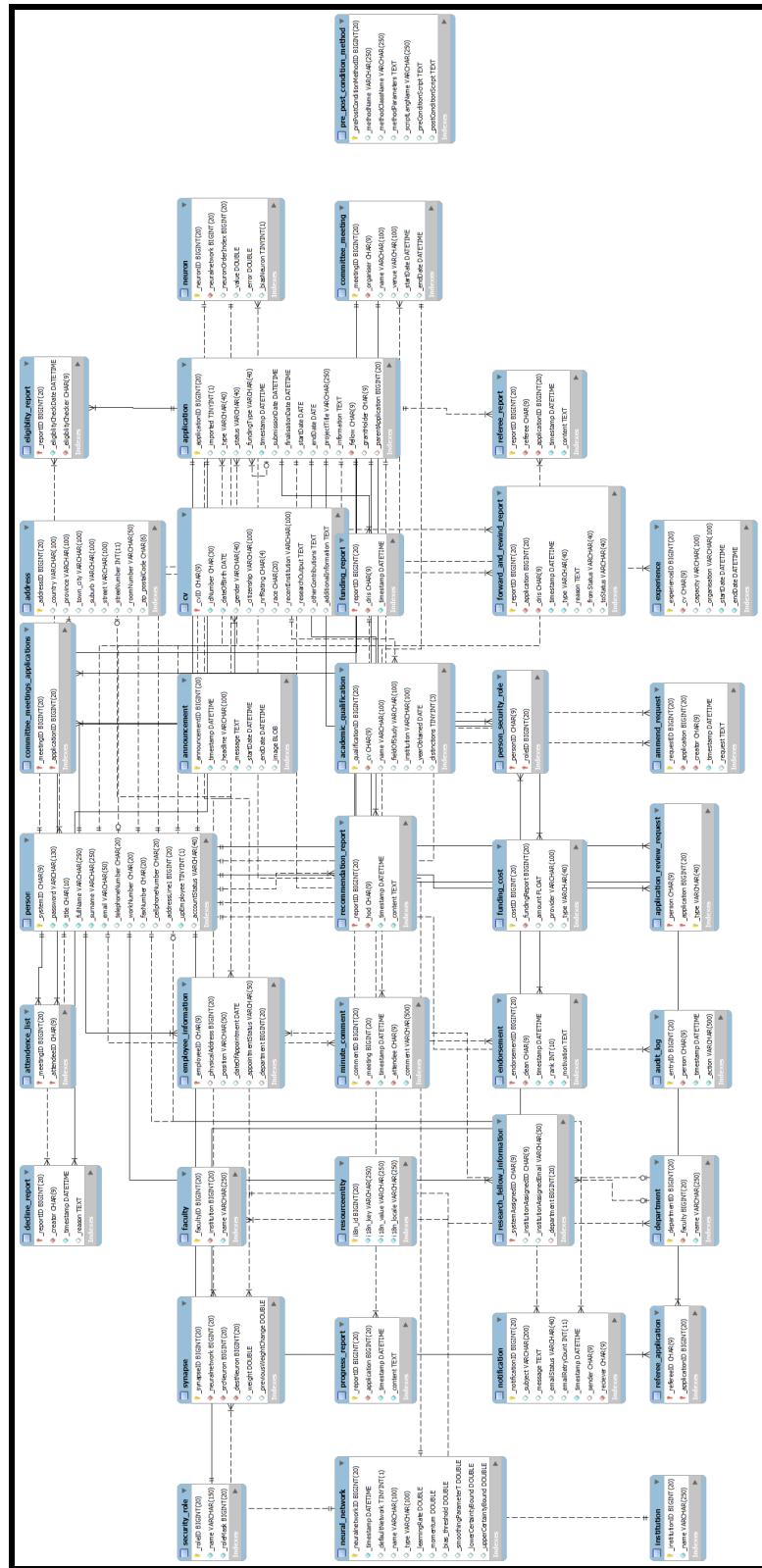


Figure 165: ERD diagram of MySQL database

## 5 Glossary:

- **Activity diagram** - A UML diagram that depicts the flow of actions or activities in the process.
- **API** - Application Programming Interface
- **Application** - Both renewal applications or new fellowship applications are seen as applications by this project.
- **CV** - Curriculum Vita
- **Domain objects** - Are the objects that are present in the system being modelled.
- **EAI** - Enterprise Application Integration
- **NRF** - National Research Foundation
- **Spreadsheet** - A special type of computer document that is used to represent data in rows and columns.
- **GlassFish** - GlassFish is a web server software package that is very flexible and compatible with Java EE applications.
- **HTML** - Hyper Text Mark-up Language
- **HTML Scrapping** - technique of extracting information from websites
- **HTTPS** - Hyper Text Transfer Protocol Secure is a higher level network oriented communication rule set that is highly secure and is used by all web browsers.
- **Java EE** - Java Enterprise Edition
- **MySQL** - Is a relational persistence database package that provides all the necessary management tools to run and manage a database server.
- **Object-Oriented** - A programming language style that encapsulates everything as an object instance of a particular class of attributes and methods.
- **JDBC** - Java Database Connection
- **MVC** - Model View Controller
- **PDF** - Portable Document Format file
- **Peoplesoft** - A management system designed by oracle.

- **Spreadsheet** - A special type of digital document that is used to represent data in rows and columns
- **UI** - User Interface
- **Use case diagram** - A UML diagram that gives a visual depiction of a service or group of services.
- **UML** - Unified modelling language. A commonly used model standard to provide technology neutral models of different aspects of software.
- **UP** - University of Pretoria
- **Work Flow** - Describes the tasks, procedural steps and tools needed for each step in a business process