COS 301 Final year project

Post-Doctoral management System

Functional testing document

October 17, 2014

Version 1.0 (Final)

Iteration 6

Prepared for

Client: Ms. Cathy Sandis (UP DRIS)

Supervisor: Prof. Stefan Gruner (SSFM Group)

by

SoftServe 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
05/03/2014	v 0.3	Added Introduction, Vision, Background	Carlo Machaba
06/03/2014	v 0.4	Added open issues. Modified some sections	Alfred Ngako
06/03/2014	v 0.5	Added methodology, scope and limitations	Mathys Ellis
08/03/2014	v 0.6	Added some wrapping to the change log which is now a table	Alfred Ngako
16/03/2014	v 0.8	Did some restructuring and document formatting	Mathys Ellis
17/03/2014	v 0.8	Also added to the glossary	Mathys Ellis
12/05/2014	v 0.9	Created new vision and scope document. Transferred necessary content from old SRS document. Performed editing and restructuring of document. Added exclusions	Mathys Ellis

Contents

List of Figures

1 Document description:

1.1 Document purpose:

This functional testing document serves the purpose of providing a detailed testing plan and test cases for functional testing of the system. Therefore the usability, integration and unit testing of the system is covered by the document. 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 functional quality assurance.

1.2 Documentation methodology

The documentation and software development methodology used by the project adhere to the guidelines set out by the agile method. Thus this document has undergone and will undergo various iterations that may extend or reduce the contents of the document.
The document was formualated using techniques described by Ammann P. and Offutt J.. Also used were techniques and skills obtained from the COS 301 course.

1.3 Document conventions:

- Documentation formulation tool: LaTeX

1.4 References:

- Ammann P., Offutt J., 2009, *Introduction to software testing*, Cambridge university press, New york.
- Yvonne Rogers, Helen Sharp, Jenny Preece, *Interaction Design: Beyond Human–Computer Interaction, 3rd ed.*, John Wiley, Sons Ltd., 2011 ISBN 0-470-66576-9

2 Testing Methodology

The tests were obtained through the means of graphs which followed all the possible paths that can be followed by using the service or unit alongside this the pre and post conditions stated in the Functional Requirements Document where also used. As a result of this process mentioned above it can be assumed that the tests cover all the functions as stated by the Functional Requirement Documentation. The section below lists all the unit tests as well as the a description of what the function does. Since the project adheres to the V-Model for testing these tests were documented at the time of architectural design and implementation.

3 Unit Testing

3.1 ApplicationProgressViewer

Name:	testGetApplicationProgressWithUserAsOwnerAndApplicationOpen
Level:	Unit
TestCode:	v APV
Module:	Application Progress Viewer
Function	APV
Description:	To test EJB's function to get the progress of an open application which belongs to the current user.

Name:	testGetApplicationProgressWithUserAsOwnerAndApplicationOpen
Level:	Unit
TestCode:	v APV
Module:	Application Progress Viewer
Function	APV
Description:	To test EJB's function to get the progress of an open application which belongs to the current user.

3.2 Application Services Util

Name:	testLoadPendingApplications
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to get the all pending applications that belong to current user.

Name:	testGetTotalNumberOfPendingApplicationsWithStatusAndReferre
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to get the number all pending applications that have specified person as referre.

Name:	testGetTotalNumberOfPendingApplicationsWithStatusAndGrantholder
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to get the number of all pending applications that have the specified person as grantholder.

Name:	testGetTotalNumberOfPendingApplicationsWithStatusAndDepartment
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to get the number of all pending applications that fall under specified department.

Name:	testGetTotalNumberOfPendingApplicationsWithStatusAndFaculty
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to get the number of all pending applications that fall under the specied faculty.

Name:	testGetTotalNumberOfPendingApplications
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to get the number of all pending applications that belong to specified user.

Name:	testDeclineAppliction
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to allow authorized user to decline an application.

Name:	testDeclineApplictionAlreadyDeclined
Level:	Unit
TestCode:	v APSU
Module:	Application Progress Viewer
Function	APSU
Description:	To test the EJB's function to behave when an application has already been declined.

3.3 Audit Trail

Name:	testLogAction
Level:	Unit
TestCode:	v APV
Module:	Application Progress Viewer
Function	APV
Description:	To test EJB's function to Log an action to the system.

3.4 CV Management

Name:	testGetApplicationProgressWithUserAsOwnerAndApplicationOpen
Level:	Unit
TestCode:	v CV
Module:	Application Progress Viewer
Function	CV
Description:	To test EJB's function to get the progress of an open application which belongs to the current user.

Name:	testCreateCVButHasCV
Level:	Unit
TestCode:	v APV
Module:	Application Progress Viewer
Function	CV
Description:	Test EJB's function to behave when a CV trying to create a CV that already exists.

Name:	testUpdateCV
Level:	Unit
TestCode:	v CV
Module:	CV Management
Function	CV
Description:	Test EJB's function to update an existing CV.

3.5 DRIS Approval

Name:	testLoadPendingEndorsedApplications
Level:	Unit
TestCode:	v DA
Module:	DRIS Approval
Function	DA
Description:	To test EJB's function to get the all pending applications that fall under a specified range.

Name:	testCountTotalPendingEndorsedApplications
Level:	Unit
TestCode:	v DA
Module:	DRIS Approval
Function	DA
Description:	To test EJB's function to get the number of all pending applications.

Name:	testLoadPendingEligibleApplications
Level:	Unit
TestCode:	v DRIS
Module:	Application Progress Viewer
Function	DRIS
Description:	To test EJB's function to get the all pending applications that fall under a specified range.

3.6 Deans Endorsement

Name:	testLoadPendingEligibleApplications
Level:	Unit
TestCode:	v DE
Module:	DeansEndorsement
Function	DE
Description:	To test EJB's function to get the all applications which fall under a specied range.

Name:	testCountTotalPendingApplications
Level:	Unit
TestCode:	v DE
Module:	DeansEndorsement
Function	DE
Description:	To test EJB's function to get the nubmer of all applications which a pending approval.

Name:	testDenyApplication
Level:	Unit
TestCode:	v DE
Module:	DeansEndorsement
Function	DE
Description:	To test EJB's function to decline a specific application.

3.7 Fast Forward And Rewind Service

Name:	testForwardApplication
Level:	Unit
TestCode:	v FFR
Module:	Fast Forward And Rewind Service
Function	FFR
Description:	Test EJB's function to Forward an application

Name:	testRewindApplication
Level:	Unit
TestCode:	v FFR
Module:	Fast Forward And Rewind Service
Function	FFR
Description:	Test EJB's function to Rewind an application

Name:	testLoadMovableApplication
Level:	Unit
TestCode:	v FFR
Module:	Fast Forward And Rewind Service
Function	FFR
Description:	Test EJB's function to show the applications which can be moved

3.8 Grant Holder Finalisation

Name:	testCreateGrantHolderCV
Level:	Unit
TestCode:	v GHF
Module:	Grant Holder Finalisation
Function	GHF
Description:	Test EJB's function to create a new CV for the grantholder.

Name:	testCreateGrantHolderCVNotValid
Level:	Unit
TestCode:	v GHF
Module:	Grant Holder Finalisation
Function	GHF
Description:	Test EJB's function behavoiur to create a new CV for the grantholder when a CV already exist.

Name:	testFinaliseApplication
Level:	Unit
TestCode:	v GHF
Module:	Grant Holder Finalisation
Function	GHF
Description:	Test EJB's function to finalise an application.

Name:	testFinaliseApplicationWithNotifications
Level:	Unit
TestCode:	v GHF
Module:	Grant Holder Finalisation
Function	GHF
Description:	Test EJB's function to finalise an application and send notifications in the same procedure.

3.9 HOD Recommendation

Name:	testDenyAppliction
Level:	Unit
TestCode:	v HOD
Module:	HODRecommendation
Function	HOD
Description:	To test EJB's function to allow a HOD to deny a grant holder.

Name:	testApproveApplicationWithoutDeansToEndorse
Level:	Unit
TestCode:	v HOD
Module:	HODRecommendation
Function	HOD
Description:	To test EJB's function to allow a HOD approve an application.

Name:	testAmmendAppliction
Level:	Unit
TestCode:	v HOD
Module:	HODRecommendation
Function	HOD
Description:	To test EJB's function to allow a HOD to ammend an application.

3.10 Location Management

Name:	testCreateInstitution
Level:	Unit
TestCode:	v LM1
Module:	Location Management
Function	LM1
Description:	To test out the ability to create an institution

Name:	testCreateFaculty
Level:	Unit
TestCode:	v LM2
Module:	Location Management
Function	LM2
Description:	To test out the ability to create an institution

Name:	testCreateDepartment
Level:	Unit
TestCode:	v LM3
Module:	Location Management
Function	LM3
Description:	To test out the ability to create a department

Name:	testUpdateInstitution
Level:	Unit
TestCode:	v LM4
Module:	Location Management
Function	LM4
Description:	To test out the ability to update an institution

Name:	testUpdateDepartment
Level:	Unit
TestCode:	v LM5
Module:	Location Management
Function	LM5
Description:	To test out the ability to update a department

Name:	testGetInstitution
Level:	Unit
TestCode:	v LM6
Module:	Location Management
Function	LM6
Description:	To test out the ability to get an institution

Name:	testGetAllFacultiesAtAnInstitution
Level:	Unit
TestCode:	v LM7
Module:	Location Management
Function	LM7
Description:	To test out the ability to get all the faculties at an institution

Name:	testGetAllDepartmentForAnInstitution
Level:	Unit
TestCode:	v LM8
Module:	Location Management
Function	LM8
Description:	To test out the ability to get all departments at an institution

Name:	testGetInstitution
Level:	Unit
TestCode:	v LM9
Module:	Location Management
Function	LM9
Description:	To test out the ability to get an institution

Name:	testGetFaculty
Level:	Unit
TestCode:	v LM10
Module:	Location Management
Function	LM10
Description:	To test out the ability to get a faculty

Name:	testGetDepartment
Level:	Unit
TestCode:	v LM11
Module:	Location Management
Function	LM11
Description:	To test out the ability to get a department

3.11 Meeting Management

Name:	testAddMeetingComments
Level:	Unit
TestCode:	v MM1
Module:	Meeting Management
Function	MM1
Description:	To test out the ability to add meeting comments

Name:	testCreateMeeting
Level:	Unit
TestCode:	v MM2
Module:	Meeting Management
Function	MM2
Description:	To test out the ability to create meetings

Name:	testEndMeeting
Level:	Unit
TestCode:	v MM3
Module:	Meeting Management
Function	MM3
Description:	To test out the ability to end meetings

Name:	testStartMeeting
Level:	Unit
TestCode:	v MM4
Module:	Meeting Management
Function	MM4
Description:	To test out the ability to start meetings

Name:	testUpdateMeetingWithoutAttendance
Level:	Unit
TestCode:	v MM5
Module:	Meeting Management
Function	MM5
Description:	To test out the ability to update meetings without any attendance

3.12 New Application

Name:	testCanFellowOpenApplication
Level:	Unit
TestCode:	v NA1
Module:	New Application
Function	NA1
Description:	To test out see if the fellow can open a new application

Name:	testCreateNewApplication
Level:	Unit
TestCode:	v NA2
Module:	New Application
Function	NA2
Description:	To test out see if the fellow can new application

Name:	testCreateNewApplicationNull
Level:	Unit
TestCode:	v NA3
Module:	New Application
Function	NA3
Description:	To test out see if the fellow can new application with a null value and if the exception is thrown

Name:	testCreateProspectiveFellowCV
Level:	Unit
TestCode:	v NA4
Module:	New Application
Function	NA2
Description:	To test out see if the fellow can create a CV

Name:	testLinkToGrantHolder
Level:	Unit
TestCode:	v NA5
Module:	New Application
Function	NA5
Description:	To test out see if the grant holder has been added to the application

Name:	testLinkToGrantHolderNull
Level:	Unit
TestCode:	v NA6
Module:	New Application
Function	NA6
Description:	To test out see if the grant holder has been added to the application with a null grant holder.

Name:	testLinkToReferee
Level:	Unit
TestCode:	v NA7
Module:	New Application
Function	NA7
Description:	To test out see if the referee has been added to the application

Name:	testLinkToRefereeNull
Level:	Unit
TestCode:	v NA8
Module:	New Application
Function	NA8
Description:	To test out see if the referee has been added to the application with null referee

Name:	testSubmitApplicationNull
Level:	Unit
TestCode:	v NA9
Module:	New Application
Function	NA9
Description:	To test out see if the application has been submitted with null value

Name:	testSubmitApplication
Level:	Unit
TestCode:	v NA10
Module:	New Application
Function	NA10
Description:	To test out see if the application has been submitted

3.13 Neural Networks

Name:	testLoadAllNeuralNetworks
Level:	Unit
TestCode:	v NN1
Module:	Neural Networks
Function	NN1
Description:	To test out the ability to load all the neural networks

Name:	testGetDefaultlNeuralNetwork
Level:	Unit
TestCode:	v NN2
Module:	Neural Networks
Function	NN2
Description:	To test out the ability to get the default neural network

Name:	testCreateNeuralNetworks
Level:	Unit
TestCode:	v NN3
Module:	Neural Networks
Function	NN3
Description:	To test out the ability to create a neural network

Name:	testMakeNeuralNetworkDefaultNetwork
Level:	Unit
TestCode:	v NN4
Module:	Neural Networks
Function	NN4
Description:	To test out the ability to make the neural network the default network

Name:	testUpdateNeuralNetwork
Level:	Unit
TestCode:	v NN5
Module:	Neural Networks
Function	NN5
Description:	To test out the ability to update the neural network

Name:	testUpdateNeuralNetworkSynapses
Level:	Unit
TestCode:	v NN6
Module:	Neural Networks
Function	NN6
Description:	To test out the ability to update the neural network synapses

Name:	testRemoveNeuralNetwork
Level:	Unit
TestCode:	v NN7
Module:	Neural Networks
Function	NN7
Description:	To test out the ability to remove a neural network

Name:	testCorrectNeuralNetwork
Level:	Unit
TestCode:	v NN8
Module:	Neural Networks
Function	NN8
Description:	To test out the ability to correct a neural network

Name:	testTrainNeuralNetwork
Level:	Unit
TestCode:	v NN9
Module:	Neural Networks
Function	NN9
Description:	To test out the ability to train a neural network

3.14 Notification

Name:	testFindAll
Level:	Unit
TestCode:	v NT1
Module:	Notification
Function	NT1
Description:	To test out see if all the notification can be found

Name:	testFindByTimeStamp
Level:	Unit
TestCode:	v NT2
Module:	Notification
Function	NT2
Description:	To test out see if all the notification can be found by timestamp

Name:	testFindByDate
Level:	Unit
TestCode:	v NT3
Module:	Notification
Function	NT3
Description:	To test out see if all the notification can be found by date

Name:	testFindByID
Level:	Unit
TestCode:	v NT4
Module:	Notification
Function	NT4
Description:	To test out see if all the notification can be found by ID

Name:	testFindByRange
Level:	Unit
TestCode:	v NT5
Module:	Notification
Function	NT5
Description:	To test out see if all the notification can be found within a range

Name:	testFindBySubject
Level:	Unit
TestCode:	v NT6
Module:	Notification
Function	NT6
Description:	To test out see if all the notification can be found by subject

Name:	testSendNotificationByEmail
Level:	Unit
TestCode:	v NT7
Module:	Notification
Function	NT7
Description:	To test out send notifications by email

Name:	testSendBatchNotification
Level:	Unit
TestCode:	v NT8
Module:	Notification
Function	NT8
Description:	To test out send batch notifications by email

Name:	testSendNotificationWithoutEmail
Level:	Unit
TestCode:	v NT9
Module:	Notification
Function	NT9
Description:	To test out send notifications without emails.

3.15 Progress Reports

Name:	testCreateProgressReport
Level:	Unit
TestCode:	v PR1
Module:	Progress Report
Function	PR1
Description:	To test out if a porgress report has been created

Name:	testSubmitProgressReport
Level:	Unit
TestCode:	v PR2
Module:	Progress Report
Function	PR2
Description:	To test out if a porgress report has been submitted

3.16 Referee Report

Name:	testCreateRefereeReport
Level:	Unit
TestCode:	v RR1
Module:	Report Report
Function	RR1
Description:	To test out if a referee report has been created

Name:	testSubmitRefereeReport
Level:	Unit
TestCode:	v PR1
Module:	Referee Report
Function	PR1
Description:	To test out if a referee report has been submitted

3.17 User Account Management

Name:	testCreateUser
Level:	Unit
TestCode:	v UA1
Module:	User Account Management
Function	UA1
Description:	To test out if a new user has been created

Name:	testCreateUserFalse
Level:	Unit
TestCode:	v UA2
Module:	User Account Management
Function	UA1
Description:	To test out if a new user has been created without systemID known

Name:	testGenerateOnDemandAccount
Level:	Unit
TestCode:	v UA3
Module:	User Account Management
Function	UA3
Description:	To test out if an on demand account has been created

Name:	testGetRemoveUserTrue
Level:	Unit
TestCode:	v UA4
Module:	User Account Management
Function	UA4
Description:	To test out if a user has been removed

Name:	testUpdateUser
Level:	Unit
TestCode:	v UA5
Module:	User Account Management
Function	UA5
Description:	To test out if a user has been updated

Name:	testViewAllAccounts
Level:	Unit
TestCode:	v UA7
Module:	User Account Management
Function	UA7
Description:	To test out if the system admin can view all accounts

Name:	testGetRemoveUserFail
Level:	Unit
TestCode:	v UA8
Module:	User Account Management
Function	UA8
Description:	To test out if a user has been removed because an exception as expected

Name:	testGenerateOnDemandAccountTrue
Level:	Unit
TestCode:	v UA9
Module:	User Account Management
Function	UA9
Description:	To test out if an on demand for the specified user

Name:	testAddresses
Level:	Unit
TestCode:	v UA10
Module:	User Account Management
Function	UA10
Description:	To test out if the address for the specified user exists

3.18 User Gateway

Name:	testAuthenticateUser
Level:	Unit
TestCode:	v UG1
Module:	User Gateway
Function	UG1
Description:	To test out if a user has been authenticated

Name:	testGetSessionFromHttpSession
Level:	Unit
TestCode:	v UG1
Module:	User Gateway
Function	UG1
Description:	To test out if it

Name:	testLogin
Level:	Unit
TestCode:	v UG2
Module:	User Gateway
Function	UG2
Description:	To test out if a user has been logged in

Name:	testLogout
Level:	Unit
TestCode:	v UG3
Module:	User Gateway
Function	UG3
Description:	To test out if a user has been logged out

Name:	testAuthenticateUserAsOwner
Level:	Unit
TestCode:	v UG4
Module:	User Gateway
Function	UG4
Description:	To test out if a user has been authenticated as the owner

4 Integretation Testing

4.1 User Accounts and Notification Service

Name:	testNewUserAndNotification
Level:	Integration
TestCode:	INUN
Module:	User Account Management and Notification Service
Function	INUN
Description:	This test ensures the integration between the User Account Management Service and Notification Service.

4.2 Referral Reports and Notification Service

Name:	testReferralAndNotification
Level:	Integration
TestCode:	IRAN
Module:	Referee Reports and Notification Service
Function	ICNA
Description:	This test ensures the integration between the Referee Report Service and Notification Service.

4.3 Meeting Management and Notification Service

Name:	testMeetingAndNotification
Level:	Integration
TestCode:	ICNA
Module:	Meeting Management and Notification Service
Function	ICNA
Description:	This test ensures the integration between the Meeting Management Service and Notification Service.

4.4 Creating A New Application

Name:	testWorkFlow
Level:	Integration
TestCode:	ICNA
Module:	New Application, Grant Holder's Report, Referee Report, HOD Recommendation, Dean's Endorsement, DRIS Approval
Function	ICNA
Description:	To test out the ability of a new application to move through the work flow required for the whole system to function.

4.5 Applying for Renewal

Name:	testWorkFlowWithRenewal
Level:	Integration
TestCode:	IAR
Module:	New Application, Grant Holder's Report, Referee Report, HOD Recommendation, Dean's Endorsement, DRIS Approval
Function	IAR
Description:	To test out the ability of a renewal application to move through the work flow required for the whole system to function.

4.6 Applying for Fellowship with Fast Forwarding and Rewind Service

Name:	testWorkFlowWithNewApplicationWithFastForward
Level:	Integration
TestCode:	IAR
Module:	New Application, Grant Holder's Report, Referee Report, HOD Recommendation, Dean's Endorsement, DRIS Approval
Function	IAR
Description:	To test out the ability of a renewal application to move through the work flow, but while fast forwarding and rewinding the application.

4.7 Google Scholar

Name:	testGoogleScholarAPI
Level:	Integration
TestCode:	IAR
Module:	New Application, Grant Holder's Report, Referee Report, HOD Recommendation, Dean's Endorsement, DRIS Approval
Function	IAR
Description:	To test out the ability of tehe EJB's to use scrapping to retrieve academic work from Google Scholar.

5 Usability Tests

For the Usability Tests Softserve will be using the DECIDE Framework:

5.1 DECIDE Framework

5.1.1 Determine the goals of evaluation

A Post-Doctoral fellow is a person who conducts research after they have completed their PhD, with the aim of deepening their knowledge in a specified. The University of Pretoria supports such research opportunities in order to the increase research output of the University. Post-Doctoral fellows who conduct their research at the University of Pretoria do so under the supervision of a staff member of the University and their research may be privately or internally funded. This is a growing field in Universities around South Africa. A lack in the software solutions for the application management of Post-Doctoral fellows has been identified by the SoftServe group.

The purpose of the evaluation is to check areas of usability that are successfully comprehended by the user, and at the same time also discover those areas that are not consistent or intuitive for the user, all in efforts to better the system. The specific goal of the evaluation is to see if the users can intuitively navigate through the application to perform common tasks by understanding their role in the system. The evaluation will also consider the time it took to perform the tasks and will help discover better solutions for parts of the evaluation where the user struggled to use the application. The evaluation is important to the success of the system, as it is imperative that users be able to understand the options available to them and be able to efficiently navigate the system for their benefit.

5.1.2 Explore the specific questions to be answered

Are users intuitively able to navigate through the application? Are users able to perform tasks faster after using the application again? Is the process of creating an application intuitive enough? How do the users feel about the forms filling process? General opinions on User Interface Elements?

5.1.3 Choose the evaluation paradigm and techniques to answer questions

The primary evaluation technique that was used for the evaluation was Usability Testing. The reason for choosing this technique is because the evaluations needed to take place in a controlled setting where the evaluators are able keep track of various behaviours of the users after performing predetermined tasks on the application. The data gathering techniques used was taking notes, video recording and monitoring key strokes. These techniques were used to produce both quantitative and qualitative data but primarily quantitative data that can easily be quantified and summarized into meaningful averages that can be further analysed at a later stage.

5.1.4 Identify the practical issues that must be addressed

1. Users will have to include current Post Doctoral students, DRIS members, the client and other computer literate members.
2. Users will have to concurrently be able to complete their tasks on the application with other users in the laboratory.
3. The users may be disturbed by external noise from nearby construction, students passing by etc., but should not affect the outcomes of the evaluation because the application was designed with simplicity and does not require a lot of focus.
4. The user only has an hour to complete all their tasks otherwise their results will have to be nullified.
5. Different users will have different tasks and as a result there could be distractions in the lab
6. The users may be disturbed by external noise from nearby construction, students passing by etc... This may affect the outcomes of the evaluation because although the application was designed with simplicity and it does require a lot of focus.
7. Making the application accessible over the UP network.

5.1.5 Decide how to deal with the ethical issues

Ethical issues, which we may encounter, would be dealing with confidentiality of information we receive from the users. We will address this issue by presenting a consent form for the use to sign – that guards their rights and our special privilege to their information. Our duty after that is to make sure their personal information is not compromised or released. Another issue, which we may encounter, would be dealing with the actual testing. The participants will be told in advance that if any part of the testing is not comfortable, they may choose to quit the testing.

5.1.6 Evaluate, analyse, interpret and present the data

We will evaluate the data looking for patterns in the user's behaviour for certain tasks and take all users into consideration that gave us consent and completed all the tasks in time. We will analyse the data by using both quantitative and qualitative frameworks and techniques to get meaningful data that can help us in improving the application's user experience and expectations. We will interpret the data to a human understandable format that allows an easier understanding evaluation taking place. We will then present the data in order to help detect where to improve or tweak the application to improve a user's experience of using the application.

5.2 Questionnaire

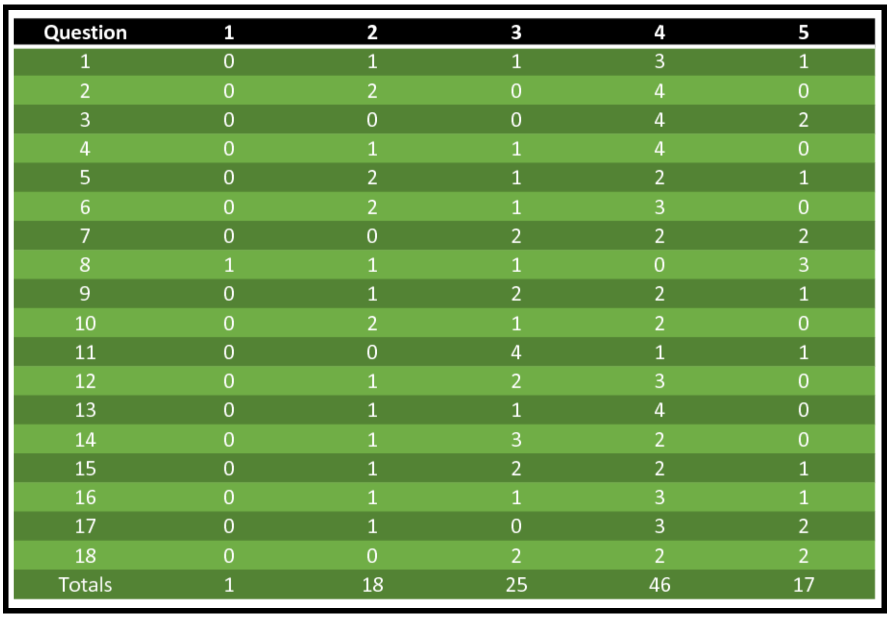
The following is the questionnaire users will be filling out. Using a scale of 1 to 5, where 1 is Strongly Disagree and 5 is Strongly Agree

Question:	Rating
Overall, I am satisfied with how easy it is to use the system.	
It was simple to use the system.	
I would effectively complete the tasks using this system.	
I was able to complete the tasks quickly using this system.	
I was able to efficiently complete the tasks using the system.	
I feel comfortable using the system.	
It was easy to learn to use the system.	
I believe I could become productive quickly using the system.	
The system gave error messages that clearly told me how to fix the problem.	
Whenever I made a mistake using the system, I could recover easily and quickly.	
It was easy to find the information I needed.	
The information provided for the system was easy to understand.	
The information was effective in helping me complete the tasks.	
The organization of information on the system screens was clear.	
The interface of the system pleasant.	
I liked using the interface of the system.	
The system has all the functions and capabilities I expect it to have.	
Overall, I am satisfied with the system.	

- Please list three things you liked most about this system software.

- Please list three things you liked least about this system software.

5.3 Results



Question	1	2	3	4	5
1	0	1	1	3	1
2	0	2	0	4	0
3	0	0	0	4	2
4	0	1	1	4	0
5	0	2	1	2	1
6	0	2	1	3	0
7	0	0	2	2	2
8	1	1	1	0	3
9	0	1	2	2	1
10	0	2	1	2	0
11	0	0	4	1	1
12	0	1	2	3	0
13	0	1	1	4	0
14	0	1	3	2	0
15	0	1	2	2	1
16	0	1	1	3	1
17	0	1	0	3	2
18	0	0	2	2	2
Totals	1	18	25	46	17

Figure 1: Table displaying the results from the tests

In essence, the users found the application easy to use as they could each complete the tasks they were required to complete successfully and with ease. The users further indicated that they were generally pleased and happy with the application.

However users felt that the application could have been designed to provide helpful error messages and a better way to navigate through the system.

With the results from our usability test we can say that our product gave the users a satisfactory user experience. Our product was able to achieve its intended use and would make the processes involved in the Post Doctoral Management easier to manage. Also, the product was designed well enough as users indicated in the satisfactory questionnaire and we were able to meet the usability goals. So our product would be good enough to undergo a field test/study (with some minor editions to it as pointed out by our users)

6 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 Scrpping** - 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