



Post-Doctoral Application Management System

Functional testing document

August 22, 2014

Version 0.9

Iteration 3

Prepared for Ms. Cathy Sandis (UP Research Office)
by SoftServe Group

Group members

Kgothatso Phatedi Alfred Ngako (12236731)

Tokololo “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

1	Document description:	4
1.1	Document purpose:	4
1.2	Documentation methodology	4
1.3	Document conventions:	5
1.4	Unit Testing	5
1.4.1	ApplicationProgressViewer	13
1.4.2	Application Services Util	13
1.4.3	Audit Trail	15
1.4.4	CV Management	16
1.4.5	DRIS Approval	16
1.4.6	Deans Endorsement	17
1.4.7	Grant Holder Finalisation	18
1.4.8	HOD Recommendation	19
1.5	References:	19
2	Glossary:	20

List of Figures

1 Document description:

1.1 Document purpose:

This vision and scope document serves the purpose of providing a detailed overview of the project's scope and its vision as well the goals that SoftServe's Post-Doctoral application management system wishes to satisfy. Further it defines the abstract interaction of stakeholders with 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 scope.

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. 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

1.3 Document conventions:

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

1.4 Unit Testing

Unit Tests	
Name:	testAddMeetingComments
Level:	Unit
TestCode:	v MM1
Module:	Meeting Man- age- ment
Function	MM1
Description:	To test out the ability to add meet- ing com- ments

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

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

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

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

Unit Tests	
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

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

Unit Tests	
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

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

Unit Tests	
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

Unit Tests	
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.

Unit Tests	
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

Unit Tests	
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

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

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

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

Unit Tests	
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

Unit Tests	
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

Unit Tests	
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

Unit Tests	
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

Unit Tests	
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

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

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

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

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

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

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

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

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

Unit Tests	
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

Unit Tests	
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

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

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

Unit Tests	
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

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

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

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

Unit Tests	
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

1.4.1 ApplicationProgressViewer

Unit Tests	
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.

Unit Tests	
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.

1.4.2 Application Services Util

Unit Tests	
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.

Unit Tests	
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.

Unit Tests	
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.

Unit Tests	
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.

Unit Tests	
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.

Unit Tests	
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.

Unit Tests	
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.

Unit Tests	
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.

1.4.3 Audit Trail

Unit Tests	
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.

1.4.4 CV Management

Unit Tests	
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.

Unit Tests	
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.

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

1.4.5 DRIS Approval

Unit Tests	
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.

Unit Tests	
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.

Unit Tests	
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.

1.4.6 Deans Endorsement

Unit Tests	
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.

Unit Tests	
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.

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

1.4.7 Grant Holder Finalisation

Unit Tests	
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.

Unit Tests	
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.

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

Unit Tests	
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.

1.4.8 HOD Recommendation

Unit Tests	
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.

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

Unit Tests	
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.

1.5 References:

- Dr.Phool, K., 2010, *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer, Heidelberg.

2 Glossary:

- **API** - Application Programming Interface
- **Application** -Both renewal applications or new fellowship applications are seen as applications by this project.
- **CV** - Curriculum Vita
- **HTML** - Hyper Text Mark-up Language
- **Java EE** - Java Enterprise Edition
- **NRF** - National Research Foundation
- **PhD** - A doctoral degree in a particular field of study.
- **PDF** - Portable Document Format file
- **Spreadsheet** - A special type of digital document that is used to represent data in rows and columns
- **Use case** - A visual depiction of a service or group of services.
- **UP** - University of Pretoria