Software Requirements Specification (SRS) Document

Project Name:	Project management of Sanchay.
Team Number:	15
Team Name	Lite Team
Team members	Akshat Tandon,Anuj Rathore,Hemant Kasat,Sanket Markan

Brief problem statement

Sanchay is a tool written primarily in Java that is used for annotation and other language processing related tasks.What is required is to make it into a server-client web application on which a user can log in and access and edit files according to his privileges.The client can also access previous versions of the same file.Apart from this various NLP tools like chunking/tagging need to be developed in the application.

System requirements

* The system must be implemented using Ruby on Rails + Angular.JS

* The system must be able to run on a server and a client can access the services of Sanchay without installing Sanchay tool.

Users profile

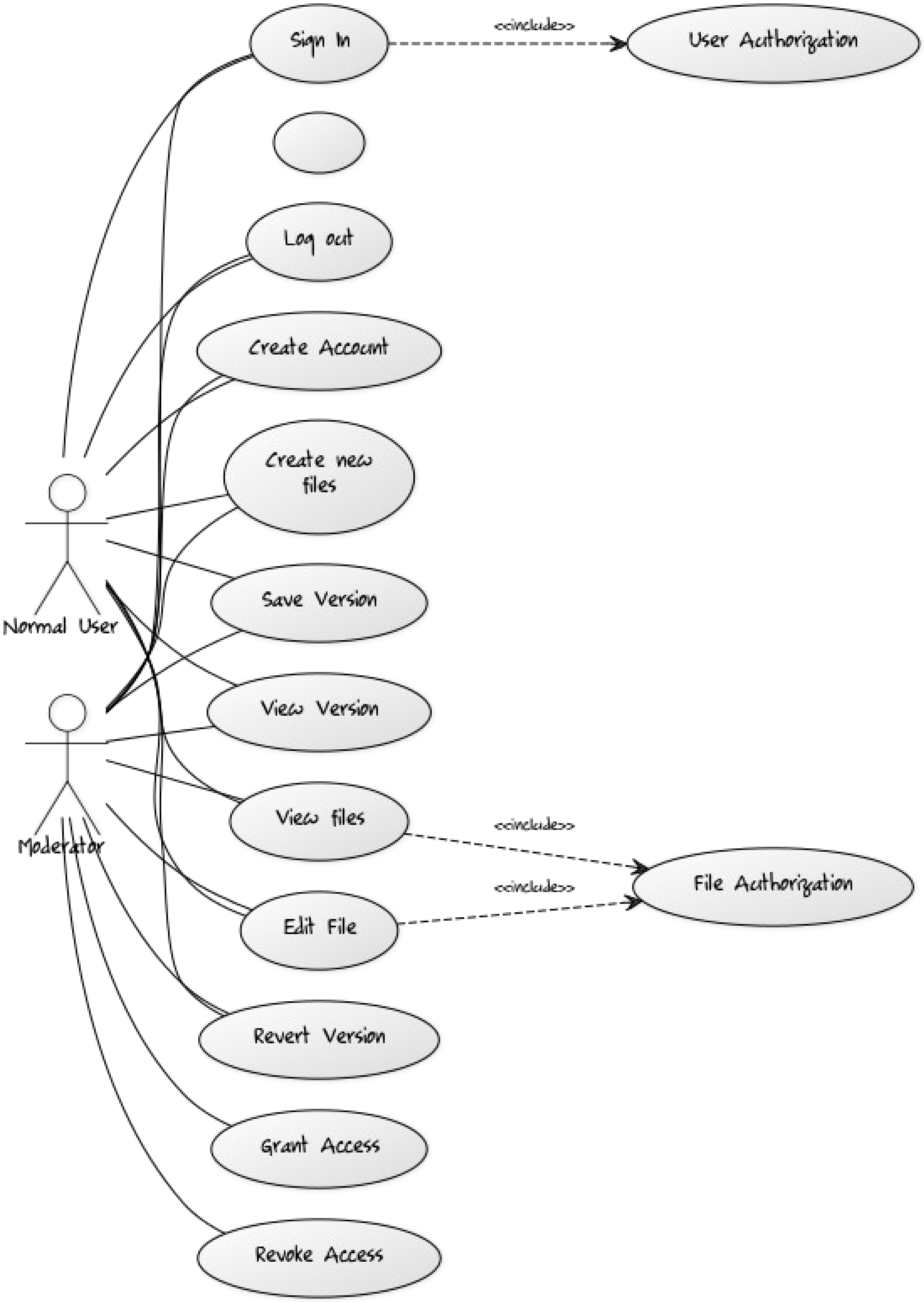
* Users who are involved in Natural Language Processing tasks.

* These users are supposed to be fairly comfortable operating basic online apps.

Feature requirements (described using use cases)

	No.	User Case Name	Description	Release
1. **11**	**1**	Create account	A new user can create his account on the site to manage and edit files.	R1
2. 1.	2	View files	A users gets a listing of all the files whom he has access to.	R1
3. 33	3	Create new file	A user creates a new file.	R1
4. 44	4	Log in	A user logs in to the system.	R1
5. 55	5	Grant access	A moderator can grant read/write access of his created files to different users.	R1
	6	Edit file	A user can edit files whom he has access to.	R1
6. 77	7	Save version	A user can save the edited file,this gets stored as a new version of the same file. Each version has a time-stamp corresponding to its date/time created.	R1
7. 88	8	View version	A user can access the past versions of the same file.	R1
8. 99	9	Revert version	A user can revert back to a past version of the same file.	R1
9. 11	10	Revoke access	A moderator can revoke the read/write writes of a user concerning a specific file.	R1
10. 11	11	Delete file.	A user (if he has the permission) can delete a file.	R1
11. 121	12	Log out	User logs out of the system.	R1

Use case diagram



Use case description

Use Case Number:	UC- 1
Use Case Name:	Create normal account
Overview:	A new user can create his account on the site to manage and edit files.
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be on the home page of Sanchay portal.
Flow:	Main (success) Flow: 1. User clicks on the create account link. 2. User enters his required credentials. 3. User clicks the Create button. 4. The credentials are verified by the system. 5. If the credentials are correct a new account is created.
	Alternate Flows: after step 3: 3.1 Credentials entered are not right. 3.2 User is notified of his anomaly. 3.3 User redirected on the crate account page.
Post Condition:	A new user account is created.

Use Case Number:	UC- 2
Use Case Name:	Log in
Overview:	A user can log into the system.
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be on the home page of Sanchay portal.
Flow:	Main (success) Flow: 1. User clicks on the sign in link. 2. User enters his required credentials. 3. User clicks the login button. 4. The credentials are verified by the system. 5. If the credentials are correct user is logged in.
	Alternate Flows: after step 3: 3.1 Credentials entered are not right. 3.2 User is notified of his anomaly. 3.3 User redirected on the log in page.
Post Condition:	The user is logged into the system and can now manipulate files.

Use Case Number:	UC- 3
Use Case Name:	Log out.
Overview:	A user logs out of the system.
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be already logged into the system.
Flow:	Main (success) Flow: 1. User clicks on log off link. 2. User is logged of the system. 3. Redirected to home page.
	Alternate Flows: after step 1: 1.1 There is a network glitch due to which user is not logged off.
Post Condition:	User is logged off the system.

Use Case Number:	UC- 4
Use Case Name:	Create a new file
Overview:	A new creates a new file.
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be already logged in.
Flow:	Main (success) Flow: 1. User clicks on the create file link. 2. User enters name , description. 3. User clicks the Create button. 4. The credentials are verified by the system. 5. If the credentials are correct a new file is created.
	Alternate Flows: after step 3: 3.1 Credentials entered are not right. 3.2 User is notified of his anomaly. 3.3 User redirected on the crate account page.
Post Condition:	A new file is created.

Use Case Number:	UC - 5
Use Case Name:	View Version.
Overview:	User views all the versions of the file
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be already logged in and have selected the file.
Flow:	Main (success) Flow: 1. User clicks on the file 2. In the menu user clicks view version 3. Page is opened having all the version of that file. 4. Version are shown along with their creation date and time.
	Alternate Flows: after step 1: 1.1 Network Error.
Post Condition:	Listing of all the versions of the given file.

Use Case Number:	UC - 6
Use Case Name:	Save Version
Overview:	User views all the versions of the file
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User is editting or opened that file.
Flow:	Main (success) Flow: 1. User clicks on save version Button on the top-right. 2. A confirmation message is shown. 3. User clicks on the OK. 4. New version is saved.
	Alternate Flows: after step 1: 1.1 Network Error.
Post Condition:	New Version of the file is saved.

Use Case Number:	UC- 7
Use Case Name:	View files
Overview:	Shows a listing of all the files which the user has access to.
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be already logged in.
Flow:	Main (success) Flow: 1. User clicks on the view files link. 2. System fetches all the files which the user has access to. 3. User redirected to a page showing the links of all the files.
	Alternate Flows: after step 1: 1.1 System not able to fetch the files due to some server/database anomaly. 1.2 User shown an error page.
Post Condition:	A user is shown a listing of all the files which he can access, and then can edit and save.

Use Case Number:	UC- 8
Use Case Name:	Edit file
Overview:	A user can edit the specified file.
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be already logged in. User must be on view files page.
Flow:	Main (success) Flow: 1. User clicks on the specified file. 2. User redirected to a page which show the file opened in a WYSIWYG editor. 3. If the user has edit permission for the specific file then the user can edit the file using WYSIWYG editor. 4. User clicks on the save button to save his edit.
	Alternate Flows: after step 1: 1.1 Due to network/database error file was not opened. After step 3 3.1 If the user does not have edit permissions he can only view the file.
Post Condition:	The changes made have been saved to the file. The previous edit(version) of the file can now be accessed using the versions view panel.

Use Case Number:	UC- 9
Use Case Name:	Revert version
Overview:	A user can revert back his main file to a specific version
Actors:	Normal User<<Primary>> , Moderator <<Primary>>
Pre condition:	User must be already logged in. User must be on versions page of the specific file.
Flow:	Main (success) Flow: 1. User clicks on the revert button besides the specific version of the file on the versions page of the file. 2. System asks for confirmation form the user. 3. If the user confirms then the main file is now reverted back to the specified version I.e it now contains edits till the date and time of the specified version.
	Alternate Flows: after step 1: 1.1 Due to network/database error file was not opened.
Post Condition:	The file is now reverted back to its past version.

Use Case Number:	UC- 10
Use Case Name:	Grant access
Overview:	Moderator grants some read/write privileges to a specific user for a specific file.
Actors:	Moderator <<Primary>>
Pre condition:	User must be already logged in. User must be on the view files page which shows all the files of the user.
Flow:	Main (success) Flow: 1. User clicks on the grant access button besides a file name. 2. User enters the username of the person whom he wishes to authorize the file. 3. Users enters specific read/write permission which he wants to provide to another user. 4. User confirms his selection. 5.File authorized to the specified user.
	Alternate Flows: after step 1: 1.1 Due to network/database error file was not opened. After step 4: 4.1 If the user already has some authorization of the file specified the user is notified of this.
Post Condition:	The file is now accessible by the new user.

Use Case Number:	UC- 11
Use Case Name:	Revoke Access
Overview:	Moderator revokes some read/write privileges from a specific user for a specific file.
Actors:	Moderator <<Primary>>
Pre condition:	User must be already logged in. User must be on the view files page which shows all the files of the user.
Flow:	Main (success) Flow: 1. User clicks on the members button besides the file. 2. User redirected to a page which shows all the users which have permission to read/write the file. 3. User de selects some permission from some specific user. 4. User confirms his selection. 5.File de authorized from a specified user.
	Alternate Flows: after step 1: 1.1 Due to network/database error file was not opened.
Post Condition:	The file is now reverted back to its past version.

Use Case Number:	UC- 12
Use Case Name:	Delete file
Overview:	User deletes file.
Actors:	User <<Primary>>,Moderator <<Primary>>
Pre condition:	User must be already logged in. User must be on the view files page which shows all the files of the user.
Flow:	Main (success) Flow: 1. User clicks on the delete button besides the file. 2. If the user has permission for editing the file then he can delete the file.After pressing the button the user is asked for reconfirmation. 3. If the user reconfirms that he wants to delete the file , the file and all its associated versions are deleted from the system.
	Alternate Flows: after step 1: 1.1 Due to network/database error file was not opened. After step 2: 2.1 User does not have edit permissions for the file and hence can't delete it. 2.2 The user is notified of this.
Post Condition:	The file and all its versions are deleted from the system.

Use Case Number:	UC - 13
Use Case Name:	User Authorization
Overview:	Checks the login credentials.
Actors:	System
Pre condition:	User is on the homepage of the website.
Flow:	Main (success) Flow: 1. User enters the user name and password. 2. User Clicks on the Login button. 3. Verification of user name and password. 4. Login to the user's homepage if its verified.
	Alternate Flows: after step 3: 3.1 A message is shown login credential is wrong. 3.2 User clicks OK 3.3 Again the homepage is opened
Post Condition:	User get its homepage opened.

Use Case Number:	UC - 14
Use Case Name:	File Authorization
Overview:	Checks the given file is valid for given operation or not.
Actors:	System
Pre condition:	User tries an operation on the file
Flow:	Main (success) Flow: 1. User clicks on file 2. A menu of operations is shown 3. User selects one. 4. It checks for the validation 5. Return True or False according to saved info.
	Alternate Flows: after step 4: 4.1. A message is shown about the error 4.2. User clicks OK 4.3. Page is rloaded
Post Condition:	Required action is done.