LinkedTek

T10-SWA - Advanced Cloud

« Advanced Cloud »

Software Qualification Specifications Acceptance

Table of contents

Introduction

The aim of this software qualification specification (SQS) is to present test procedures involved in order to validate the **LikedTek** project.

1. Acronyms table

NA	Not applicable
NR	Not run
POK	Partially OK
TC	Test case

2. Project context

The aim of this project is to provide a fully workable professional social network system, the main features will be the abilities to find and add new relations (thanks to some markers like schools, works or even relation of relation "friend of friend"), the abilities to publish some posts and comment them, to see on a single page all the relation's posts (like a feed on Facebook, Twitter or LinkedIn) and to send messages to the other platform's users.

3. Global results

	OK	KO	POK	NR	NA
Manual tests	100%	0%	0%	0%	0%
	57/57	0/57	0/57	0/57	0/57

4. Manual tests

MAN_010	Login and register tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Try to connect with valid account and password. **Expected:** *Connection succeed, user is logged, and feed activities page is displayed.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Try to connect with invalid account and password. **Expected:** *Connection failed popup is displayed to warn user that connection failed.*	OK	
TC_03	**Test:** 1. Open linked test application. 2. Click on register button. **Expected:** *Register page is displayed.*	OK	
TC_04	**Test:** 1. Open linked test application. 2. Click on register button. 3. Try to create an account with all fields correctly filled. **Expected:** *Account creation succeed, succeed popup must be displayed. When user validate this popup, the login page must be displayed.*	OK	
TC_05	**Test:** 1. Open linked test application. 2. Click on register button. 3. Try to create an account with any field not filled. **Expected:** *Account creation failed; popup is displayed to warn user that the field must be filled.*	OK	
TC_06	**Test:** 1. Open linked test application. 2. Click on register button. 3. Try to create an account with already register email address.	OK	

	Expected:		
	Account creation failed; popup is displayed to warn user that the current email address is already in use.		
Final result	Execution date	02/06/20109	**OK**
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_020	Schools tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. **Expected:** *School and companies page must be displayed. School list is displayed on the school list module.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On school and company creation module, select school as input type. 5. On school and company creation module, enter a school name. 6. On school and company creation module, enter a school description. 7. On school and company creation module, select the school country. 8. Click on validate button. **Expected:** *School creation succeed, popup is displayed to warn user that school creation succeeds.*	OK	
TC_03	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On school and company creation module, let any field blank. 5. Click on validate button. **Expected:** *School creation failed; popup is displayed to warn user that school creation failed.*	OK	
TC_04	**Test:** 1. Open linked test application.	OK	

	2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On the school list module, click on any school name. **Expected:** *Edit school popup must be displayed with school name and description used as placeholders into their own fields.*	
TC_05	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On the school list module, click on any school name. 5. On the edition popup, enter new school name or description. **Expected:** *School edition succeeds, a popup is displayed to warn user that the school edition succeeds.*	OK

Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_030	Companies tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. **Expected:** *School and companies page must be displayed. School list is displayed on the school list module.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On school and company creation module, select company as input type. 5. On school and company creation module, enter a company name. 6. On school and company creation module, enter a company description.	OK	

	7. On school and company creation module, select the company country. 8. Click on validate button. **Expected:** *Company creation succeed, popup is displayed to warn user that company creation succeeds.*	
TC_03	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On school and company creation module, let any field blank. 5. Click on validate button. **Expected:** *Company creation failed; popup is displayed to warn user that company creation failed.*	OK
TC_04	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On the company list module, click on any company name. **Expected:** *Edit company popup must be displayed with company name and description used as placeholders into their own fields.*	OK
TC_05	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On the company list module, click on any company name. 5. On the edition popup, enter new company name or description. **Expected:** *Company edition succeeds, a popup is displayed to warn user that the company edition succeeds.*	OK

Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_040	List user tests	Result	Comment
TC_01	**Test:** 1. Go to the neo4j back-office and connect with valid login and password. 2. Click on validate button. **Expected:** *Connection succeeds and database management page is displayed.*	OK	
TC_02	**Test:** 1. Go to the neo4j back-office and connect with invalid login and password. 2. Click on validate button. **Expected:** *Connection failed message is displayed.*	OK	
TC_03	**Test:** 1. Go to the neo4j back-office and connect with valid login and password. 2. Click on validate button. 3. On the node label field, click on User button. **Expected:** *User list is displayed with graph visualisation.*	OK	
TC_04	**Test:** 1. Go to the neo4j back-office and connect with valid login and password. 2. Click on validate button. 3. On the node label field, click on User button. 4. On the left menu, click on Text button **Expected:** *User list is displayed with list visualisation.*	OK	

Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_050	Handle user test	Result	Comment
TC_01	**Test:** 1. Go to the neo4j back-office and connect with valid login and password. 2. Click on validate button. 3. Create user **Expected:** *User has been created.*	OK	
TC_02	**Test:** 4. Go to the neo4j back-office and connect with valid login and password.	OK	

	5. Click on validate button. 6. Delete user **Expected:** *User has been deleted.*		
TC_03	**Test:** 7. Go to the neo4j back-office and connect with valid login and password. 8. Click on validate button. 9. Edit user **Expected:** *User has been edited.*	OK	

Final result	Execution date	02/06/2019	OK
	Tested version	Version 1.0	
	Tester name	Clément BELVISEE	

MAN_060	Handle schools and companies' tests	Result	Comment
TC_01	**Test:** 10. Go to the neo4j back-office and connect with valid login and password. 11. Click on validate button. 12. Delete school and school relations. **Expected:** *School has been deleted.*	OK	
TC_02	**Test:** 1. Go to the neo4j back-office and connect with valid login and password. 2. Click on validate button. 3. Delete company and company relations. **Expected:** Company *has been deleted.*	OK	

Final result	Execution date	02/06/2019	OK
	Tested version	Version 1.0	
	Tester name	Clément BELVISEE	

MAN_070	Handle posts and comments tests	Result	Comment
TC_01	**Test:** 1. Go to the neo4j back-office and connect with valid login and password. 2. Click on validate button. 3. Create post **Expected:** post *has been created.*	OK	
TC_02	**Test:**	OK	

		Result	Comment
	4. Go to the neo4j back-office and connect with valid login and password. 5. Click on validate button. 6. Delete company **Expected:** post *has been deleted.*		
TC_03	**Test:** 7. Go to the neo4j back-office and connect with valid login and password. 8. Click on validate button. 9. Edit company **Expected:** post *has been edited.*	OK	
TC_04	**Test:** 1. Go to the neo4j back-office and connect with valid login and password. 2. Click on validate button. 3. Create comment **Expected:** comment *has been created.*	OK	
TC_05	**Test:** 4. Go to the neo4j back-office and connect with valid login and password. 5. Click on validate button. 6. Delete comment **Expected:** comment *has been deleted.*	OK	
TC_06	**Test:** 7. Go to the neo4j back-office and connect with valid login and password. 8. Click on validate button. 9. Edit comment **Expected:** comment *has been edited.*	OK	
Final result	Execution date — 02/06/2019 Tested version — Version 1.0 Tester name — Clément BELVISÉE	OK	

MAN_080	User remove school or companies safety tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu.	OK	

	4. Try to delete school through a school list module. **Expected:** *School and companies page must be displayed but there is no way for user to delete school.*		
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. Try to delete company through a company list module. **Expected:** *School and companies page must be displayed but there is no way for user to delete company.*	OK	
Final result	Execution date 02/06/2019 Tested version Version 1.0 Tester name Romain PAWLACZYK	**OK**	

MAN_090	User school and company subscription tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On school list, click on any school where user is not yet subscribed. **Expected:** *School subscription succeeds; a popup is displayed to warn user that the school subscription succeeds.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On company list, click on any company where user is not yet subscribed. **Expected:** *Company subscription succeeds; a popup is displayed to warn user that the company subscription succeeds.*	OK	
TC_03	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On school list, click on any school where user is subscribed.	OK	

	Expected:		
	School subscription is removed; a popup is displayed to warn user that the school subscription is now removed.		
TC_04	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on schools and companies button on left menu. 4. On company list, click on any company where user is subscribed. **Expected:** *Company subscription is removed; a popup is displayed to warn user that the company subscription is now removed.*	OK	

Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_100	User profile tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on relations button on left menu. **Expected:** *Relation page is displayed. On your relationships module, user relation, school relation and companies relations are listed.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on relations button on left menu. 4. Click on your relationships module, click on any user. 5. On the user information popup, click on view user profile button. **Expected:** *User profile page must be displayed.*	OK	
TC_03	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on relations button on left menu. 4. Click on your relationships module, click on any school. 5. On the school information popup, click on view school profile button. **Expected:**	OK	

	School profile page must be displayed.		
Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_110	User relations tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on relations button on left menu. 4. On add new relation module, enter username. 5. Click on search button. 6. On the user information popup, click on add as relation button. **Expected:** *Add user as relation succeeds; A popup is displayed to warn user that user is now add as relation.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on relations button on left menu. 4. On your relationships module, click on any user relation. 5. Click on search button. 6. On the user information popup, click on remove relation button. **Expected:** *User relation is removed; A popup is displayed to warn user that user relation is now removed.*	OK	
Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_120	User feed tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. **Expected:** *User is now connected, and actuality feed is displayed.*	OK	
Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_130	User posts tests	Result	Comment

TC_01	**Test:**
	   1. Open linked test application.
	   2. Connect with a valid account and password.
	   3. Click on posts management button on left menu.
	
	Expected:
	Post management page must be displayed.
		OK	

TC_02	**Test:**
	   1. Open linked test application.
	   2. Connect with a valid account and password.
	   3. Click on posts management button on left menu.
	   4. On add new post module, enter a post title.
	   5. On add new post module, enter a post content.
	   6. Click on post button.
	
	Expected:
	User post succeeds; a popup is displayed to warn user that post succeeds.
		OK	

TC_03	**Test:**
	   1. Open linked test application.
	   2. Connect with a valid account and password.
	   3. Click on posts management button on left menu.
	   4. On add new post module, let post title field or post description field blank.
	   5. Click on post button.
	
	Expected:
	User post failed; a popup is displayed to warn user that post failed.
		OK	

TC_04	**Test:**
	   1. Open linked test application.
	   2. Connect with a valid account and password.
	   3. Click on posts management button on left menu.
	   4. On my posts module, click on any post.
	   5. On the post information module, click on remove post button.
	
	Expected:
	User post is removed with success; a popup is displayed to warn user that post is now removed.
		OK	

TC_05	**Test:**
	   1. Open linked test application.
	   2. Connect with a valid account and password.
	   3. Click on posts management button on left menu.
		OK	

4. On my posts module, click on any post.		
5. On the post information module, enter new title or new content on their own fields.		
6. Click on edit post button.		
Expected:		
User post is edited with success; a popup is displayed to warn user that post is now edited.		

Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_140	User comments tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on any post on actuality feed. 4. On write new comment module of post page, enter a new comment into the comment field. 5. Click on add comment button. **Expected:** *Comment is now added to the post; a popup is displayed to warn user that the comment had been posted.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on any post on actuality feed. 4. On write new comment module of post page, let the comment field blank. 5. Click on add comment button. **Expected:** *Comment post failed; a popup is displayed to warn user that the comment had not been posted.*	OK	
TC_03	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on posts management button on left menu. 4. On my comment module, click on any comment. 5. On comment information module, click on remove comment button. **Expected:** *Comment removed with success; a popup is displayed to warn user that the comment is now removed.*	OK	
TC_04	**Test:** 1. Open linked test application.	OK	

	2. Connect with a valid account and password. 3. Click on posts management button on left menu. 4. On my comment module, click on any comment. 5. Enter new data as comment into the comment field. 6. On comment information module, click on edit comment button. **Expected:** *Comment edited with success; a popup is displayed to warn user that the comment is now edited.*		
Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_150	Relations suggestions tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on relations button on left menu. **Expected:** *On relation suggestion module, user is suggested based on common friends or schools and companies subscriptions.*	OK	
Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

MAN_160	User messages tests	Result	Comment
TC_01	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on messages button on left menu. **Expected:** *Message page is displayed. On inbox module, the message received by user are displayed.*	OK	
TC_02	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on messages button on left menu. 4. On the inbox list, click on delete button for any received message.	OK	

	Expected: *A confirmation popup is displayed. In case of user validation, the message is removed.*		
TC_03	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on messages button on left menu. 4. On the inbox list, click on reply button for any received message. 5. Enter data into the content field. 6. Click on validate button. **Expected:** *Sending reply success; a popup is displayed to warn user that the reply message had been send.*	OK	
TC_04	**Test:** 1. Open linked test application. 2. Connect with a valid account and password. 3. Click on messages button on left menu. 4. On the inbox list, click on reply button for any received message. 5. Let the reply content field empty. 6. Click on validate button. **Expected:** *Sending reply failed; a popup is displayed to warn user that the reply message had not been send.*	OK	

Final result	Execution date	02/06/20109	OK
	Tested version	Version 1.0	
	Tester name	Romain PAWLACZYK	

5. Traceability matrix

This matrix makes the correspondence between components, classes, functions and requirements developed in the request for proposal.

Id test	Test title	Id requirement	Requirement description
MAN_010	Login and register tests	REQ_FUNC_010	You must be able to register / log-in on the software
MAN_020	Schools tests	REQ_FUNC_020	You must be able to add / edit / list the schools
MAN_030	Companies tests	REQ_FUNC_030	You must be able to add / edit / list the companies
MAN_040	List user tests	REQ_FUNC_040	As administrator, you must be able to list all the users
MAN_050	Handle user test	REQ_FUNC_050	As administrator, you must be able to edit / remove /add/ ban the users
MAN_060	Handle schools and companies tests	REQ_FUNC_060	As administrator, you must be able to delete schools or companies
MAN_070	Handle posts and comments tests	REQ_FUNC_070	As administrator, you must be able to edit / remove / add all the posts and user's comments.
MAN_080	User remove school and companies safety tests	REQ_FUNC_080	A user (non-administrator) is not able to remove school or companies
MAN_090	User school and company subscription tests	REQ_FUNC_090	A user can subscribe or unsubscribe in several companies or in several schools
MAN_100	User profile tests	REQ_FUNC_100	A user can see other users' profiles (schools, companies, …)
MAN_110	User relations tests	REQ_FUNC_110	A user can add or remove another user to is relations
MAN_120	User feed tests	REQ_FUNC_120	A user has access to a feed with all is relations' activities (new post, comment, …)
MAN_130	User posts tests	REQ_FUNC_130	A user can add a new post, edit or remove it
MAN_140	User comments tests	REQ_FUNC_140	A user can comment a post (and edit / remove the comment as well)
MAN_150	Relation suggestions tests	REQ_FUNC_150	The platform needs to suggest to the user new relation (thanks to common markers like schools, companies or relations)
MAN_160	User messages tests	REQ_FUNC_160	A user can send, see and respond to messages from another user.