

I7BAC – Acceptance test specification

Project Name:	Orbit-Share	
Project counselor	Brian Vestergaard Danielsen (BVD)	
	Student Number:	Name:
<i>Team Member 1</i>	201510924	Andreas Blaabjerg
<i>Team Member 2</i>	201611828	Christopher Thomas William Bernold

Date of conduction: 07-12-2020

Introduction

This document is in place to describe the acceptance test specification for the bachelor project. This acceptance specification is split up into epics to test each coherent user story, which is then used to evaluate the systems compliance with the defined requirements and check if the product is acceptable for delivery.¹

To complete the acceptance test black box testing has been used to manually execute the tests found in this document.²

Each user story has a main scenario, and, in some cases, exceptions are tested to ensure that all aspects of the story works. These exceptions are not defined directly in the requirements section found in the documentation, but rather can be found in their respective sequence diagrams.

Functional requirements:

1. Epic 1

Acceptance test for:		User Story 1.1 - Login			
Scenario:		Main Scenario			
Precondition:		User has opened a browser, routed to the orbit-share webpage, and has valid login credentials.			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses any login button on welcome page	Routes to login page	Routes to login page	Accepted	
2	User enters credentials and presses Login	User is routed to the home page and receive a JWT.	Valid token is returned from the server and user is routed to home	Accepted	

Acceptance test for:		User Story 1.1a - Login exception			
Scenario:		Exception: User inserts incorrect information			
Precondition:		User is at login page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User inserts incorrect information	User is prompted with an error message and the server	User is prompted with an error message and the server	Accepted	

¹ <https://softwaretestingfundamentals.com/acceptance-testing/>

² <https://softwaretestingfundamentals.com/black-box-testing/>

		returns a 401 HTTP status.	responds with a 401.		
--	--	----------------------------	----------------------	--	--

Acceptance test for:		User Story 1.2 - Logout			
Scenario:		Main Scenario			
Precondition:		User is logged into webpage			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses logout button on toolbar	User is logged out of the webpage and valid JWT is removed from the local storage	User is logged out of the webpage, routed to welcome page and credentials are removed from local storage	Accepted	

Acceptance test for:		User Story 1.3 - Returning user			
Scenario:		Main Scenario			
Precondition:		User has logged in and checked the “stay logged in” check box			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User navigates to webpage	User is automatically routed to home page	Check box can be checked scenario is same as normal login	Not Accepted	Not Implemented

Acceptance test for:		User Story 1.4 - Administrative authentication			
Scenario:		Main Scenario			
Precondition:		Administrator has a login and is already at the login page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	Administrator uses their login on the login form	Admin is logged in with administrative rights.		Not Accepted	Not Implemented

Acceptance test for:		User Story 1.5 - Register new user			
Scenario:		Main Scenario			
Precondition:		User has opened browser and routed to orbit-share webpage			

Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses login	Routes to login page	User is routed to Login page	Accepted	
2	User presses register button	User is routed to registration page	User is routed to registration page	Accepted	
3	User fills out form information and creates an account	User is routed back to the login page and 201 is returned from the server.	Server responds with a 201 indicating a created user, and the page routes the user to the login page.	Accepted	

Acceptance test for:		User Story 1.5a Register new user exception			
Scenario:		Exception: user enters an existing user's username			
Precondition:		User is on the registration page.			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User fills out registration form with a username that already exists	User receives an error message telling that the username already exists. server returns a 400 status.	Server responds with a 400 HTTP status and an error message is rendered	Accepted	

Acceptance test for:		User Story 1.6 - Login - OAuth, UNI-Login			
Scenario:		Main Scenario			
Precondition:		User has opened browser and routed to orbit-share webpage			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses login with UNI-Login	Routes to UNI-Login broker		Not Accepted	Not implemented
2	User uses UNI-Login credentials	User is routed to home page		Not Accepted	Not implemented

2. Epic 2

Acceptance test for:	User Story 2.1 - View/Edit				
Scenario:	Main Scenario				
Precondition:	User is logged into webpage				
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses profile button	User is routed to their profile, where either three 200 or 304 status codes are returned from the server.	New users are routed to their profile where 200 status codes are received for fetching user info, content and collections.	Accepted	
2	User presses edit button on credentials and makes changes	User edits are saved		Not Accepted	Not implemented

Acceptance test for:	User Story 2.2 - Delete profile				
Scenario:	Main Scenario				
Precondition:	User is on their profile page.				
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses delete profile	User is prompted with a dialog box confirming deletion	Dialog box pops up	Accepted	Dialog box does not show username.
2	User confirms deletions	User is deleted from system		Not Accepted	Missing implementation

Acceptance test for:	User Story 2.3 - Profile picture				
Scenario:	Main Scenario				
Precondition:	User is on their profile page.				
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User uploads a profile picture	Uploaded profile picture is shown on profile		Not Accepted	Not implemented

3. Epic 3

Acceptance test for:		User Story 3.1 - Upload content			
Scenario:		Main Scenario			
Precondition:		User is logged in			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses plus button on toolbar	User is routed to content page	User is routed to content page	Accepted	
2	User fills out form and presses "create content"	User is routed to a new page containing a view of the newly created content. Server returns a 201 status code.		Accepted	Content viewer does not work when creating new content

Acceptance test for:		User Story 3.2 - Private or Public check			
Scenario:		Main Scenario			
Precondition:		User is logged in and on the content creation page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User flags content as private and submits form	Content is not shown on the "home" page and server returns a 200 status	Content is not shown on the "home" page and server returns a 200 status	Accepted	

Acceptance test for:		User Story 3.3 - Connecting Tags to work			
Scenario:		Main Scenario			
Precondition:		User is logged in and on the content creation page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User adds tags to a content	Content has user's entered tags connected to it	Content has user's entered tags connected to it	Accepted	Tags are part of content when creating a content.

Acceptance test for:		User Story 3.4 - View own published content			
Scenario:		Main Scenario			
Precondition:		User is logged in			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User navigates to profile and expands a content they have created	User can view their own created content	Server returns 200 status codes when receiving content and content comments along with google api successful google get request http codes.	Accepted	

Acceptance test for:		User Story 3.5 - Edit own published content			
Scenario:		Main Scenario			
Precondition:		User is logged in and is viewing one of their content			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses edit on a field and changes values.	Field values update		Not Accepted	Not implemented
2	User presses save changes	Changes are saved.		Not Accepted	Not implemented

Acceptance test for:		User Story 3.6 - Delete own published content			
Scenario:		Main Scenario			
Precondition:		User is logged in and is inside of their profile page.			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses delete content (x button) on one of their content	Server returns 204 and content is deleted	Server returns 204 and content is deleted	Accepted	

Acceptance test for:	User Story 3.7 - View other users' content				
Scenario:	Main Scenario				
Precondition:	User is logged in and on home page				
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses expand button on any content	Content is opened into a full screen view, server sends successful http status codes when getting content and comment info, along with successful status codes from google api for the viewer.	Server returns 200 status codes when receiving content and content comments along with google api successful google get request http codes.	Accepted	

Acceptance test for:	User Story 3.8 - Export content				
Scenario:	Main Scenario				
Precondition:	User is currently viewing any content found on the page				
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses download button on content	Content is downloaded to the device. Server returns a 200 status code.	Server returns 200 get request and browser triggers a download	Accepted	

Acceptance test for:	User Story 3.9 - Converting when exporting uploaded content				
Scenario:	Main Scenario				
Precondition:	User is currently viewing any content found on the page				
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses "export as" button	User is prompted with options		Not Accepted	Not implemented
2	User clicks export choice	Content is converted to user's choice and triggers a download		Not Accepted	Not implemented

Acceptance test for:		User Story 3.10 - Export to ORBIT-Box			
Scenario:		Main Scenario			
Precondition:		User is currently viewing any content found on the page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses export to ORBIT-Box	Content is exported to ORBIT-Box		Not Accepted	Format has to be ORBIT-boxes format (<i>which is undefined currently</i>). Not implemented

Acceptance test for:		User Story 3.11 Subscribing to Tags			
Scenario:		Main Scenario			
Precondition:		User is logged in			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User updates "interests" field	Content is filtered by user choices		Not Accepted	Not implemented

Acceptance test for:		User Story 3.12 - First relevant tags			
Scenario:		Main Scenario			
Precondition:		User logs in for the first time			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User is prompted with tags of interest	Content is filtered by user choices		Not Accepted	Not implemented

4. Epic 4

Acceptance test for:		User Story 4.1 - Home Page			
Scenario:		Main Scenario			
Precondition:		User is on login page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User enters login credentials to access home page by clicking login	User is routed to 'home' page	User is routed to 'home' page	Accepted	

Acceptance test for:		User Story 4.2 - Searching for content			
Scenario:		Main Scenario			
Precondition:		User is logged in and at home page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User enters something to search for in search bar	Content is filtered by users input	Content is filtered by users input	Accepted	Errors causing home page not to go to initial state

5. Epic 5

Acceptance test for:		User Story 5.1 - Rating			
Scenario:		Main Scenario			
Precondition:		User is logged in			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses stars on any content	Rating is updated	Rating is not updated	Not Accepted	Functionality not implemented , UI in place.

Acceptance test for:		User Story 5.2 - Comment content			
Scenario:		Main Scenario			
Precondition:		User is logged in			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User expands a content	Routes to specified content.	Routes to specified content	Accepted	
2	User enters a comment and presses comment button	Users comment is shown on the content, and two HTTP requests return two successful statuses for posting and getting the comment.	Server returns 200 when posting the comment and 200 when getting the comment to display it on the page.	Accepted	HTTP status for posting comment is 200 but should be 201

Acceptance test for:		User Story 5.3 - User subscription			
Scenario:		Main Scenario			
Precondition:		User is logged in			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User subscribes to another user	User gets a notification whenever user subscribed to uploads new content		Not Accepted	Not implemented

6. Epic 6

Acceptance test for:		User Story 6.1 - Upload content to collection			
Scenario:		Main Scenario			
Precondition:		User is logged in and on upload content page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses create collection button	User is routed to create collection page	User is routed to create collection page	Accepted	
2	User enters form data and presses plus button	Another content form is shown	Another content form is shown and nothing is returned from the server.	Accepted	
3	User fills out content form once again and presses create collection	Four server calls are made, two posts returning 200 and 201 when creating the collection and 200 twice from the home component to fetch content and collections, routing the user to the home page..	Successful http statuses are returned from the server and the user is routed to the home page.	Accepted	Renders new instance of content before rerouting.

Acceptance test for:		User Story 6.2 - Attaching tags			
Scenario:		Main Scenario			
Precondition:		User is logged in and on upload collection page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User attaches tags to collection	Collection has tags connected to them	Tags are passed in the request body when creating a collection.	Accepted	Tags are not shown anywhere, but is attached to each content

Acceptance test for:		User Story 6.3 - Adding content to a collection			
Scenario:		Main Scenario			
Precondition:		User is logged in and on profile page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses "+" on a content and adds it to one of their collections	Collection is updated.		Not Accepted	Not implemented

Acceptance test for:		User Story 6.4 - Viewing collection			
Scenario:		Main Scenario			
Precondition:		User is logged in and on profile page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User hovers mouse over collection and can view content inside of each collection	User can view content inside of collection	User can view content inside of collection	Accepted	Missing items in view

Acceptance test for:		User Story 6.5 - Edit collection			
Scenario:		Main Scenario			
Precondition:		User is logged in and on profile page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses "edit" on collection and updates fields	Collection is updated, but content is <u>not</u> updated.		Not Accepted	Content has to be updated inside of

					viewer. Not implemented
--	--	--	--	--	-------------------------

Acceptance test for:		User Story 6.6 - Delete collection			
Scenario:		Main Scenario - Publish Collection			
Precondition:		User is logged in and on profile page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses "x" on collection	Collection is deleted.		Not Accepted	Not implemented

Acceptance test for:		User Story 6.7 - Publish collection			
Scenario:		Main Scenario			
Precondition:		User is logged in and on profile page			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	User presses publish on collection	Collection is viewable for all users.		Not Accepted	Collections are always public.

7. Epic 7

Acceptance test for:		User Story 7.1			
Scenario:		Main Scenario			
Precondition:		Admin is logged in with administrative account			
Step	Test description	Expected result	Actual result	Accepted/Not Accepted	Comments
1	Admin presses "x" on any content, collection or comment	Item is removed.		Not Accepted	Not implemented

Non-functional requirements

Non-functional requirements					
Scenario:		Main Scenario			
Precondition:		Admin is logged in with administrative account			
"FURPS+" id	Test description	Expected result	Reasoning	Accepted/Not Accepted	Comments
Functionality					
1.1	User data is stored securely, meaning it is inaccessible for others than the user or official administrators	Backend calls can not be made without proper authentication. Storing of passwords directly is not permitted	To help keep key user data stored securely we use Bcrypt for encrypting passwords which is based on an encryption called Blowfish. This is done to avoid storing the users passwords directly. Also when calling endpoints a middleware will check if the user calling it is authorized to perform	Accepted	

			the requested action..		
1.2	User upload files are stored as content or as content part of a collection in database	Files uploaded by users is visible in database upon upload finish	Files are accessible to download from content link right after upload	Accepted	
1.3	Content files of types pdf, jpeg/jpg, png, docs and Orbit-Box works as intended while other file formats are not uploadable	filetypes expected working: pdf, jpeg/jpg, png, docs. Orbit-Box type does not exist to test.	Files of all types are uploadable, however .orbit can not be tested, as it does not exist	Not accepted	Type .Orbit-Box does not exist yet, as such it can not be tested. Regardless types .pdf, .docx, excel, mp4, .jpg, .png works. Any type is uploadable and downloadable in single content.
1.4	File size limitation is x>5MB	Files of 5MB and above size are not uploadable	4MB file was uploadable. 5MB file was not uploadable with error message payload to large.	Accepted	4MB file was uploadable. 5MB file was not uploadable. Above
Usability					
2.1	System interface is expected to be intuitive.	Test results from a set of scenarios tested on people should result in being more positive than negative.	No test subjects available due to school closing down because of Covid-19	Not accepted	A test was planned out on what points a user had to complete to conclude if the interface is intuitive.
2.2	System design layout matches	Majority of SPA layout	Majority of SPA layout	Accepted	

	ORBIT-Labs official styling profile	styling matches ORBIT-Lab s official styling profile colours	styling matches ORBIT-Labs official styling profile colours		
Reliability					
3.1	System is expected to be online at almost all times except for during maintenance	System is expected to be online as much as deployment platform allows	Not immediately testable, however deployed on heroku, our guarantee is what Heroku will guarantee https://status.heroku.com/ over the last 60 days the uptime is 99.99998%	Accepted	Deployment sites often have an SLA agreement in place in relation to what they are willing to guarantee.
Performance					
4.1	system needs to score a rating of 90 or higher in Google Pagespeed	Google Pagespeed rating is 90 or higher	Lighthouse test run on all pages and the average value of this gives: $100+82+96+96+96+95+95+95=755$ $755/8 = 94,375$. Above 90.	Accepted	Google page speed is based on the extension for google Chrome called Lighthouse.
4.2	System's scaling of files are according to filesize	Smaller files are faster than bigger files in xx	One big file was uploaded and one small file was uploaded. Small files uploads near instantly	Accepted	
4.3	System's exporting of files are according to filesize	Smaller files exports faster than larger files in xx	One big file was downloaded and one small file	Accepted	

			was downloaded. big files take longer, can take seconds		
Supportability					
5.1	System is supported on Google Chrome v. 83 and newer and on Firefox v. 77 and newer. No guarantees are made for older versions.	Deployed application works as intended on Google Chrome v. 83 and newer and also Firefox v. 77 and newer.	Application was tested in both firefox and chrome to a satisfactory degree. And intended functionality was in order. with few known bugs (on known bugs and errors list)	Accepted	
5.2	The system supports English language	The SPA layout is available in english.	English is available	Accepted	
5.3	System guarantees support for file sizes of up to 5MB	Files up to 5MB work in the SPA, as intended	System does in fact support files up to, but not on 5MB.	Accepted	
5.4	System contains alternative login options for official administrators	System allows for administrators to login using normal credentials as opposed to UNI-login	System does not allow for administrators to login using normal credentials as opposed to UNI-login	Not Accepted	System does not contain such a feature. Not implemented
Implementation					
6.1	Client is produced in Angular	Client is produced in Angular	Client is produced in Angular	Accepted	
6.2	Server is coded in Node.js using Express	Server is coded in Node.js using Express	Server is coded in Node.js using Express	Accepted	
6.3	File structure follows a known folder "treelike"	Collections can store one or	Collections can store one or more Contents	Accepted	Collections being somewhat useless at the

	structure where collections contain content(s)	more Contents			moment, can store several contents
--	--	------------------	--	--	--