

## Use Case 1

<b>Use case ID:</b>	UC 07
<b>Use Case Name:</b>	Create directory
<b>Description:</b>	Unprivileged user must be able to create a directory.

Actors:	User																	
Pre-conditions:	User is logged in, has active plan, open file navigator, current working directory is the user’s home directory																	
Post-conditions:	User is presented with Create Directory, Delete, Move, Rename buttons																	
Frequency of use:	Daily																	
Flow of Events:	<table><tr><th>#</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Navigate to target directory from home directory via clicks</td><td>Display directory structure as tree, present contents upon selection</td></tr><tr><td>2</td><td>Select “Create Directory”</td><td>Display text field for user input</td></tr><tr><td>3</td><td>Type directory name into text field</td><td></td></tr><tr><td>4</td><td>Click “Save” button</td><td>If the directory does not exist in the destination, commit change via Java FS API</td></tr></table>			#	Actor Action	System Response	1	Navigate to target directory from home directory via clicks	Display directory structure as tree, present contents upon selection	2	Select “Create Directory”	Display text field for user input	3	Type directory name into text field		4	Click “Save” button	If the directory does not exist in the destination, commit change via Java FS API
#	Actor Action	System Response																
1	Navigate to target directory from home directory via clicks	Display directory structure as tree, present contents upon selection																
2	Select “Create Directory”	Display text field for user input																
3	Type directory name into text field																	
4	Click “Save” button	If the directory does not exist in the destination, commit change via Java FS API																
Variations:																		
Exceptions:	Directory name already exists																	
Developer Notes:	Integrate with Linux commands																	

## Use Case 2

<b>Use case ID:</b>	UC 08
---------------------	-------

<b>Use Case Name:</b>	Delete directory
<b>Description:</b>	Unprivileged user must be able to delete a directory.

<b>Actors:</b>	User													
<b>Pre-conditions:</b>	User is logged in, has active plan, open file navigator, current working directory is the user's home directory													
<b>Post-conditions:</b>	User is presented with Create Directory, Delete, Move, Rename buttons													
<b>Frequency of use:</b>	Daily													
<b>Flow of Events:</b>	<table> <tr> <th>#</th><th>Actor Action</th><th>System Response</th></tr> <tr> <td>1</td><td>Navigate to and select target directory from home directory via clicks</td><td>Display directory structure as tree, present contents upon selection</td></tr> <tr> <td>2</td><td>Select "Delete"</td><td></td></tr> <tr> <td>3</td><td>Click "Save" button</td><td>Commit change via Java FS API</td></tr> </table>		#	Actor Action	System Response	1	Navigate to and select target directory from home directory via clicks	Display directory structure as tree, present contents upon selection	2	Select "Delete"		3	Click "Save" button	Commit change via Java FS API
#	Actor Action	System Response												
1	Navigate to and select target directory from home directory via clicks	Display directory structure as tree, present contents upon selection												
2	Select "Delete"													
3	Click "Save" button	Commit change via Java FS API												
<b>Variations:</b>	Recursively delete contents													
<b>Exceptions:</b>														
<b>Developer Notes:</b>	Integrate with Linux commands													

### Use Case 3

<b>Use case ID:</b>	UC 09
<b>Use Case Name:</b>	Move directory
<b>Description:</b>	Unprivileged user must be able to move a directory.

<b>Actors:</b>	User
----------------	------

<b>Pre-conditions:</b>	User is logged in, has active plan, open file navigator, current working directory is the user’s home directory																	
<b>Post-conditions:</b>	User is presented with Create Directory, Delete, Move, Rename buttons																	
<b>Frequency of use:</b>	Daily																	
<b>Flow of Events:</b>	<table><tr><th>#</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Navigate to target directory from home directory via clicks</td><td>Display directory structure as tree, present contents upon selection</td></tr><tr><td>2</td><td>Select “Move”</td><td>Display text field for user input</td></tr><tr><td>3</td><td>Type new directory path into text field</td><td></td></tr><tr><td>4</td><td>Click “Save” button</td><td>If the directory does not exist in the destination, commit change via Java FS API</td></tr></table>			#	Actor Action	System Response	1	Navigate to target directory from home directory via clicks	Display directory structure as tree, present contents upon selection	2	Select “Move”	Display text field for user input	3	Type new directory path into text field		4	Click “Save” button	If the directory does not exist in the destination, commit change via Java FS API
#	Actor Action	System Response																
1	Navigate to target directory from home directory via clicks	Display directory structure as tree, present contents upon selection																
2	Select “Move”	Display text field for user input																
3	Type new directory path into text field																	
4	Click “Save” button	If the directory does not exist in the destination, commit change via Java FS API																
<b>Variations:</b>	Recursively move contents																	
<b>Exceptions:</b>	Directory name already exists at target																	
<b>Developer Notes:</b>	Integrate with Linux commands																	

#### Use Case 4

<b>Use case ID:</b>	UC 10
<b>Use Case Name:</b>	Rename directory
<b>Description:</b>	Unprivileged user must be able to rename a directory.

<b>Actors:</b>	User
<b>Pre-conditions:</b>	User is logged in, has active plan, open file navigator, current working directory is the user's home directory
<b>Post-conditions:</b>	User is presented with Create Directory, Delete, Move, Rename buttons

Frequency of use:	Daily		
Flow of Events:			
	#	Actor Action	System Response
	1	Navigate to target directory from home directory via clicks	Display directory structure as tree, present contents upon selection
	2	Select “Rename”	Display text field for user input
	3	Type new directory name into text field	
	4	Click “Save” button	If the directory does not exist in the destination, commit change via Java FS API
Variations:			
Exceptions:	Directory name already exists		
Developer Notes:	Integrate with Linux commands		

## Use Case 5

<b>Use case ID:</b>	UC 14
<b>Use Case Name:</b>	View photo
<b>Description:</b>	Unprivileged user must be able to view a photo.

<b>Actors:</b>	User				
<b>Pre-conditions:</b>	User is logged in, has active plan, open file navigator, current working directory is the user's home directory				
<b>Post-conditions:</b>	User can view the photo in a view window				
<b>Frequency of use:</b>	Daily				
<b>Flow of Events:</b>	<table> <tr> <th>#</th><th>Actor Action</th><th>System Response</th></tr> </table>		#	Actor Action	System Response
#	Actor Action	System Response			

	1	Navigate to and select target directory from home directory via clicks	Display directory structure as tree, present contents upon selection
	2	Select file	
	3	Click "View" button	Check if compatible photo, if so, open in photo view window
<b>Variations:</b>			
<b>Exceptions:</b>	Incompatible file type		
<b>Developer Notes:</b>			

## Use Case 6

<b>Use case ID:</b>	UC 17
<b>Use Case Name:</b>	Change plan
<b>Description:</b>	Unprivileged user must be able to change plan.

Actors:	User		
Pre-conditions:	User is logged in, has active plan, open file navigator, current working directory is the user’s home directory		
Post-conditions:	User can upload more files		
Frequency of use:	Monthly		
Flow of Events:			
	#	Actor Action	System Response
	1	Click on “Change Plan” button	Navigate to new view
	2	Select plan	
	3	Click “Confirm Selection” button	Commit new plan to DB

<b>Variations:</b>	
<b>Exceptions:</b>	
<b>Developer Notes:</b>	

## Use Case 7

<b>Use case ID:</b>	UC 20
<b>Use Case Name:</b>	Create user
<b>Description:</b>	Admin user must be able to create user.

Actors:	Admin														
Pre-conditions:	Admin is logged in, open navigator														
Post-conditions:	New user exists														
Frequency of use:	Weekly														
Flow of Events:	<table><tr><th>#</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Click on “Create User” button</td><td>Navigate to new view</td></tr><tr><td>2</td><td>Input username</td><td></td></tr><tr><td>3</td><td>Click “Confirm” button</td><td>Check if username exists, if not, create</td></tr></table>			#	Actor Action	System Response	1	Click on “Create User” button	Navigate to new view	2	Input username		3	Click “Confirm” button	Check if username exists, if not, create
#	Actor Action	System Response													
1	Click on “Create User” button	Navigate to new view													
2	Input username														
3	Click “Confirm” button	Check if username exists, if not, create													
Variations:															
Exceptions:	User already exists														
Developer Notes:															

## Use Case 7

<b>Use case ID:</b>	UC 20
<b>Use Case Name:</b>	Create user
<b>Description:</b>	Admin user must be able to create user.

Actors:	Admin														
Pre-conditions:	Admin is logged in, open navigator														
Post-conditions:	New user exists														
Frequency of use:	Weekly														
Flow of Events:	<table><tr><th>#</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Click on “Create User” button</td><td>Navigate to new view</td></tr><tr><td>2</td><td>Input username</td><td></td></tr><tr><td>3</td><td>Click “Confirm” button</td><td>Check if username exists, if not, create</td></tr></table>			#	Actor Action	System Response	1	Click on “Create User” button	Navigate to new view	2	Input username		3	Click “Confirm” button	Check if username exists, if not, create
#	Actor Action	System Response													
1	Click on “Create User” button	Navigate to new view													
2	Input username														
3	Click “Confirm” button	Check if username exists, if not, create													
Variations:															
Exceptions:	User already exists														
Developer Notes:															