

Use Case 1

Use case ID:	UC 07
Use Case Name:	Create directory
Description:	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 target directory		
Post-conditions:	User is presented with Create Directory, Delete, Move, Rename buttons		
Frequency of use:	Daily		
Flow of Events:			
	#	Actor Action	System Response
	1	Select “Create Directory”	Display text field for user input
	2	Type directory name into text field	
	3	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

Use case ID:	UC 08
Use Case Name:	Delete directory
Description:	Unprivileged user must be able to delete a directory.

Actors:	User											
Pre-conditions:	User is logged in, has active plan, open file navigator, current working directory is the target 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>Select “Delete”</td><td></td></tr><tr><td>2</td><td>Click “Save” button</td><td>Commit change via Java FS API</td></tr></table>			#	Actor Action	System Response	1	Select “Delete”		2	Click “Save” button	Commit change via Java FS API
#	Actor Action	System Response										
1	Select “Delete”											
2	Click “Save” button	Commit change via Java FS API										
Variations:	Recursively delete contents											
Exceptions:												
Developer Notes:	Integrate with Linux commands											

Use Case 3

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

Actors:	User
Pre-conditions:	User is logged in, has active plan, open file navigator, current working directory is the target directory
Post-conditions:	User is presented with Create Directory, Delete, Move, Rename buttons
Frequency of use:	Daily
Flow of Events:	

	#	Actor Action	System Response
	1	Select “Move”	Display text field for user input
	2	Type new directory path into text field	
	3	Click “Save” button	If the directory does not exist in the destination, commit change via Java FS API
Variations:	Recursively move contents		
Exceptions:	Directory name already exists at target		
Developer Notes:	Integrate with Linux commands		

Use Case 4

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

Actors:	User											
Pre-conditions:	User is logged in, has active plan, open file navigator, current working directory is the target 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>Select “Rename”</td><td>Display text field for user input</td></tr><tr><td>2</td><td>Type new directory name into text field</td><td></td></tr></table>			#	Actor Action	System Response	1	Select “Rename”	Display text field for user input	2	Type new directory name into text field	
#	Actor Action	System Response										
1	Select “Rename”	Display text field for user input										
2	Type new directory name into text field											

	<table><tr><td>3</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>	3	Click “Save” button	If the directory does not exist in the destination, commit change via Java FS API
3	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

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

Actors:	User											
Pre-conditions:	User is logged in, has active plan, open file navigator, current working directory is the target directory											
Post-conditions:	User can view the photo in a view window											
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>Select file</td><td></td></tr><tr><td>2</td><td>Click “View” button</td><td>Check if compatible photo, if so, open in photo view window</td></tr></table>			#	Actor Action	System Response	1	Select file		2	Click “View” button	Check if compatible photo, if so, open in photo view window
#	Actor Action	System Response										
1	Select file											
2	Click “View” button	Check if compatible photo, if so, open in photo view window										
Variations:												
Exceptions:	Incompatible file type											
Developer Notes:												

Use Case 6

Use case ID:	UC 17
Use Case Name:	Change plan
Description:	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:	<table><tr><th>#</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Click on “Change Plan” button</td><td>Navigate to new view</td></tr><tr><td>2</td><td>Select plan</td><td></td></tr><tr><td>3</td><td>Click “Confirm Selection” button</td><td>Commit new plan to DB</td></tr></table>			#	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
#	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													
Variations:															
Exceptions:															
Developer Notes:															

Use Case 7

Use case ID:	UC 20
Use Case Name:	Create user
Description:	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 border="1"> <thead> <tr> <th>#</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <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> </tbody> </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

Use case ID:	UC 20
Use Case Name:	Create user
Description:	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:	

	#	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:			