# **Cloud Storage Domain - TryCloud**





#### **DEEP INTEGRATION**

Access data from FTP, Windows Network
Drive, SharePoint, NFS, Object storage and
more.

Simple deployment, configuration and integration lower costs and decrease risk while leveraging existing IT investments.



#### **ALL FILES**

All file types, including doc/ppt/xls/csv/psd/ai, can be backed up without changing the file path so there is no need to rearrange the file structure after recovery.



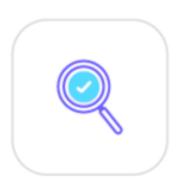
#### ULTIMATE SECURITY

An open source, self-hosted product.

Powerful on-server and End-to-end

Encryption.

Files are stored inside your company infrastructure, without any third party ever gaining access.



#### **KEYWORD SEARCH AND RECOVERY**

For large amounts of data, simply key in the keywords during emergency recovery to search for related headings and content in order to quickly locate the file you want.



Contact TRIAL

Login O

#### **EASY ACCESS ANYWHERE**

A modern and easy-to-use web interface, desktop clients and mobile apps.

Real-time collaboration and instant access to all data from any device, anywhere!



#### SUPPORTS MULTIPLE TYPES OF INTERNET DEVICES

Automatically uploads PC, notebook computer, NAS, and cell phone photos and other data to the cloud so you don't have to worry about sudden hard drive failure or running out of space.



Framework	Cucumber BDD framework	
Design Pattern	Singleton, Page Object model (POM), POJO	
Languages	Java OOP language, Gherkin, HTML, SQL	
Testing Tools	Selenium WebDriver, Junit, Cucumber, TestNG, Apache POI, Selenium Grid, JDBC, RestAssured	
Build Management Tool	Maven	
CI/CD tool	Jenkins	
Project Methodology	Waterfall, Agile, Agile-Scrum, Agile-Kanban	
Bug Tracking / Project management tool	Jira-Xray, (Jira)	
IDEs	IntelliJ, (Eclipse), SQL Developer, (DBeaver), Postman	
Version Control System	Git, GitHub, (Gitlab)	
Browsers	Google Chrome, Firefox, Internet Explore, Edge, Safari	
Operating System	MacOS, Microsoft Windows	
Testing types	Dynamic, System, Functional, UI/FrontEnd, DataBase, API (backEnd), End-To-End, Smoke, Regression, Black box, gray box, cross-browser, parallel testing	



# **Company names: TryCloud**

# What does this company do?

This is a business consulting and software development company

This company' projects such as:

- Produce several business theme softwares
- Consulting business ventures
- Business training courses



# Describe the TryCloud project?

# This project is cloud storage web-based app

#### **Features are:**



User contact management

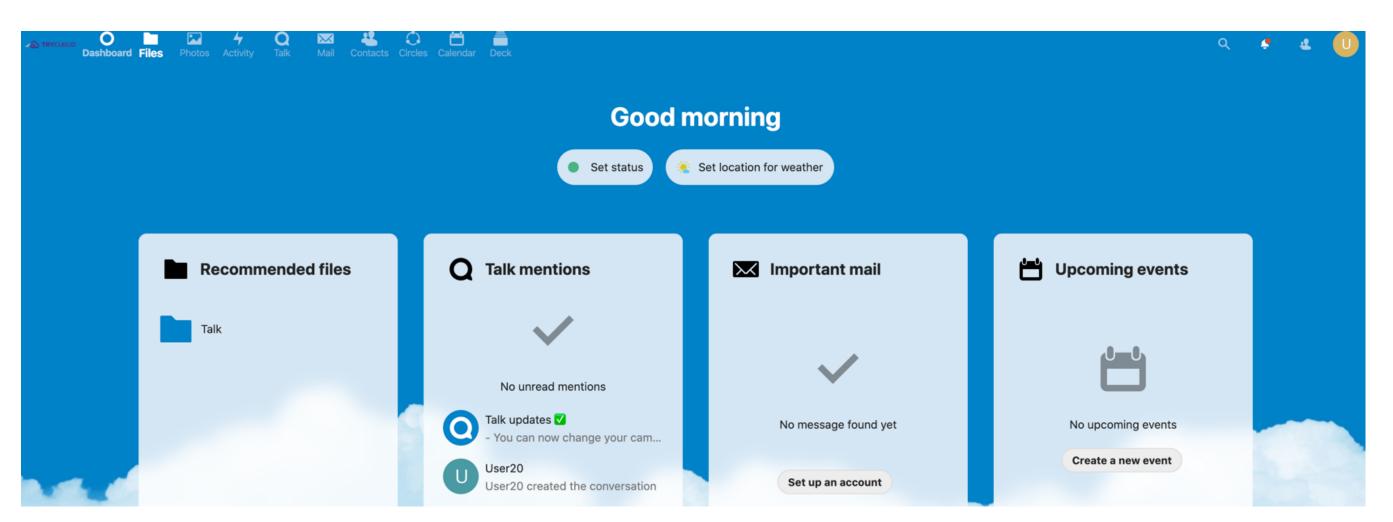
Communication - text / audio / video chat

Keyword search file & All types of files like: doc/ppt/xls/csv/psd/ai

Customizable dashboards and wallboards

Calendar & event creation

Customizable workflows





# **Assignment Description**

In this assignment each group will create a BDD Cucumber framework to test the TryCloud company's cloud storage webBased application. Each student at least automate one user story.

The framework will have: Java

Maven

Selenium WebDriver, Junit, Cucumber

Design patterns

Git/Github

Conduct "Code Review" whenever is needed to explain any re-usable method to the team, or go over **PR**.

#### **Application information:**

Credentials link: click here



# **Assignment to do list:**

## 1. Each group creates a Cucumber-BDD Framework

- Conduct a meeting with the team to SET UP the project [look at the next slide for how to set up a project]
- One person responsible to set up the project & push to the GitHub AND add all the other team members as collaborators
- All the team member accept the invitation & clone the project

#### 2. Perform UI/FrontEnd automation

- Create one features file per story
- Create separate branch for the class, and send PR to merge with Master branch

Note: You will practice performing **Smoke & Regression testings** from this framework after learning Jenkins & Selenium Grid topics.



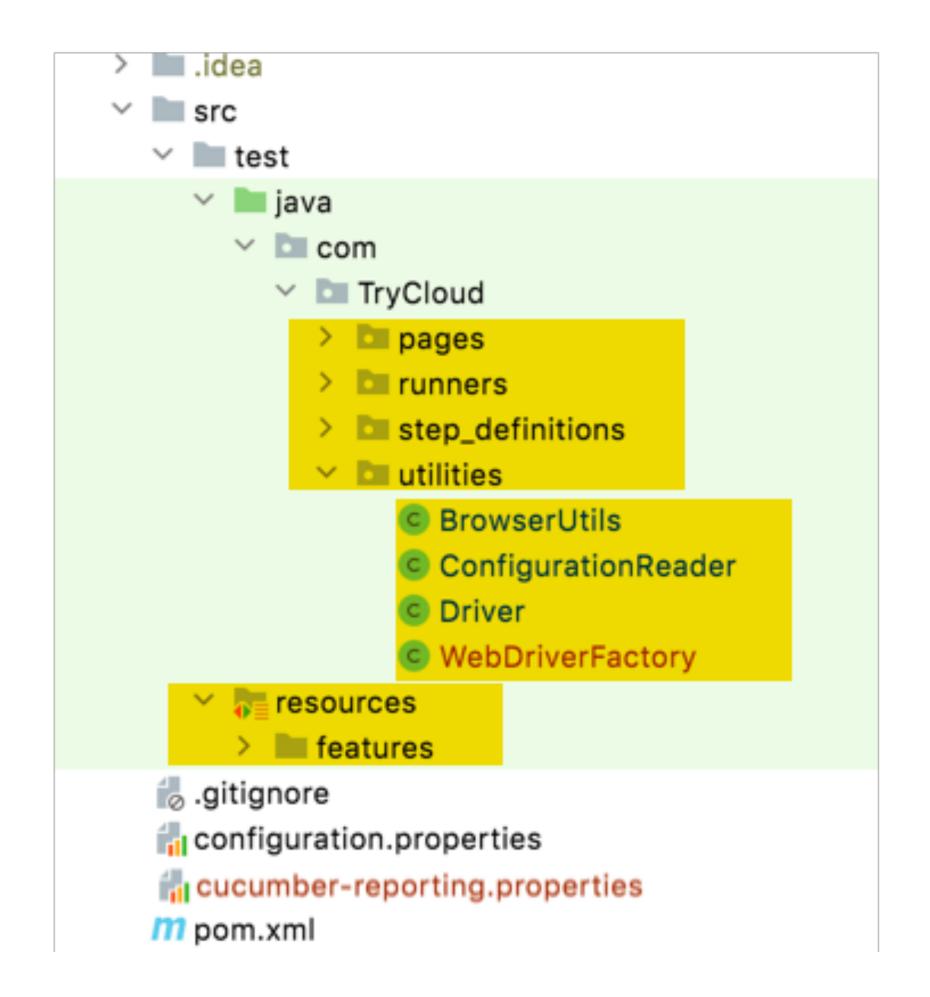
## Project set up

1. Create a maven project.

Group-id: com.trycloud

Artifact-id: trycloudAutomation

- 2. Add all the required dependencies for automation to the pom file.
- 3. Under src/test/java/ create following packages:
  - a. com.trycloud.pages
  - b. com.trycloud.runner
  - c. com.trycloud.step\_definitions
  - d. com.trycloud.utilities
- 4. Create "resources" directory and "features" package
- 5. Under the project folder add file configuration.properties add file cucumber-reporting.properties add file .gitignore
- 6. Create feature file based on the given user stories & Acs
- 7. Create classes under the setp-definitions to automate the scenario or scenario outline





Write feature files based on the requirements. User story & AC provided. AC has the steps already so you just need to add gherkin language. You can write Scenario or scenario outline. You can adjust the steps if needed.

#### How I write feature file?

# For example:

1. Story: As a user, I should be able to login to TryCloud app

#### AC #1 - user login successfully with valid credentials

- 1. Navigate to login page
- 2. Enter valid username and password
- 3. Click login button
- 4. Verify user on the dashboard

#### AC #2 verify user login fails with invalid credentials

- 1. Navigate to login page
- 2. Enter valid username invalid password
- 3. Click login button
- 4. Message "Wrong username or password." should be displayed



1. Feature: As a user, I should be able to login.

## **Background:**

Given user on the login page <a href="http://qa3.trycloud.net/index.php/login?clear=1">http://qa3.trycloud.net/index.php/login?clear=1</a>

Scenario - verify user login successfully

When user enter valid username and password

And user click login button

Then Verify user launched to the dashboard

Scenario- verify user login fail with invalid credentials

When user enter invalid credentials

And click login button

Then "Wrong username or password." message should be displayed



1. Feature: As a user, I should be able to login.

Scenario Outline: Login with valid credentials

Given user on the login page <a href="http://qa3.trycloud.net/index.php/login?clear=1">http://qa3.trycloud.net/index.php/login?clear=1</a>

When user use username "<username>" and passcode "<password>"

And user click login button

Then user should be at dashboard page

#### **Examples**:

lusername	password
l user7	l Userpass123 l
l user34	l Userpass123 l
l user99	l Userpass123 l



2. Story: As a user, I should be access all the main modules of the app.

## AC#1 - verify users access to the main modules

- 1. Login as a user
- 2. Verify the user see the following modules:

**Dashboard** 

**Files** 

**Photos** 

Activity

Talk

Contacts

Circles

Calendar

Desk





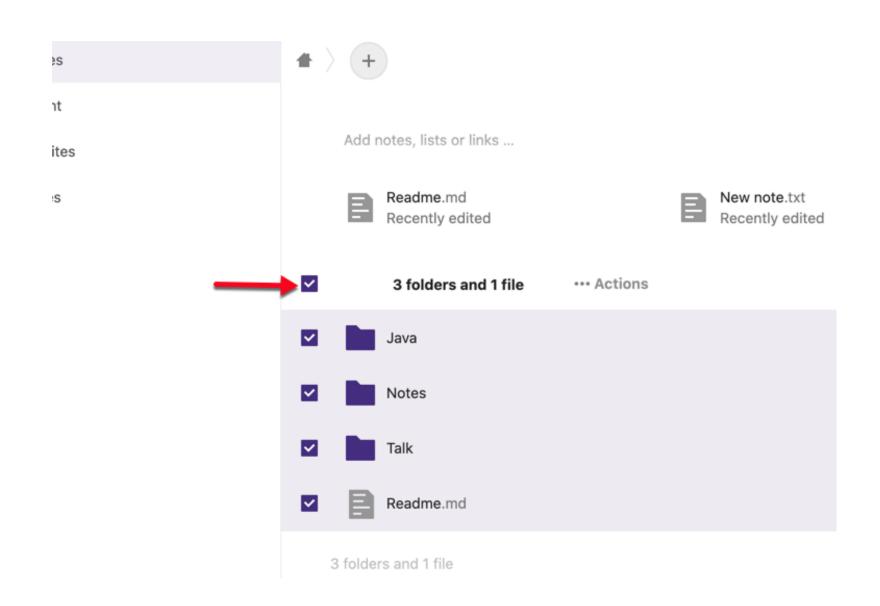
### AC#1 - verify users can access to Files module

- 1. Login as a user
- 2. Verify the page tile is "Files Trycloud"

### AC#2 - verify users can select all the uploaded files from the page

- 1. Login as a user
- 2. Click the top left checkbox of the table
- 3. Assert all the files are selected

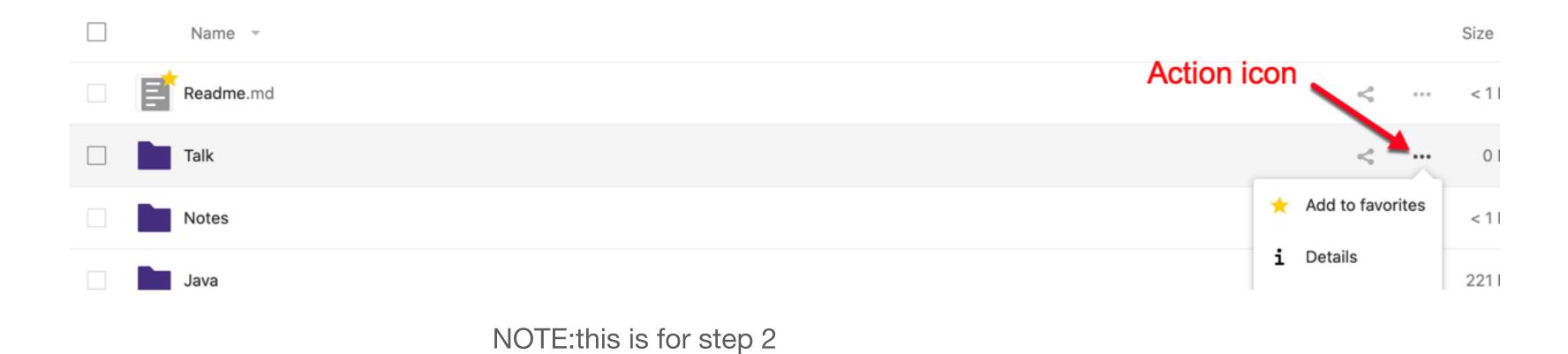
(Pre-condition: there should be at least 2 files are uploaded the page)

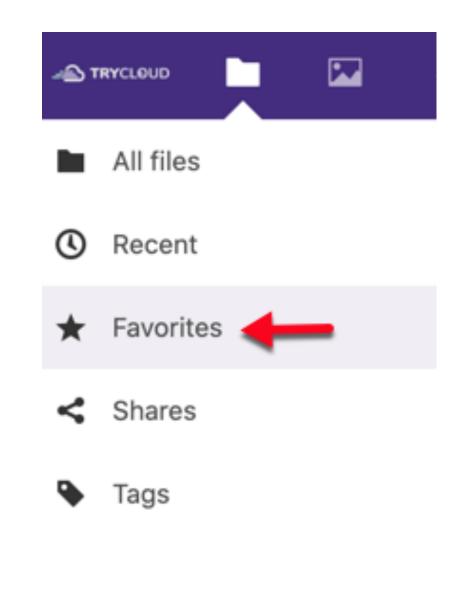




#### AC #1 - users can add a file to favorites.

- 1. Login as a user
- 2. Click action-icon from any file on the page
- 3. Choose "Add to Favorites" option
- 4. Click "Favorites" sub-module on the lest side
- 5. Verify the chosen file is listed on the table (Pre-condition: there should be at least 1 file is uploaded)





NOTE: this is for step 4



## AC #1 - verify users can remove a file from favorites.

- 1. Login as a user
- 2. Click action-icon from any file on the page
- 3. Click "Remove from Favorites" option
- 4. Verify that the file is removed from Favorites sub-module's table.

(Pre-condition: there should be at least 1 file is added to favorites table)

## AC #2 - verify users can upload a file directly to the homepage (File-Uploading)

- 1. Login as a user
- 2. Click the "+" icon on top
- 3. Click "upload file"
- 4. Upload a file
- 5. Verify the file is displayed on the page



## AC #1- verify users can create a folder

- 1. Login as a user
- 2. Click the "+" icon on top
- 3. Click "New Folder"
- 4. Write a folder name
- 5. Click submit icon
- 6. Verify the folder is displayed on the page

## AC #2 - verify users can upload a file inside a folder

- 1. Login as a user
- 2. Choose a folder from the page
- 3. Click the "+" icon on top
- 4. Click "upload file"
- 5. Upload a file
- 6. Verify the file is displayed on the page

(Pre-condition: there should be at least 1 folder is created on the filers page)



### AC #1 - verify users can delete a file/folder.

- 1. Login as a user
- 2. Click action-icon from any file on the page
- 3. Choose "delete files" option
- 4. Click deleted files on the left bottom corner
- 5. Verify the deleted file is displayed no the page.

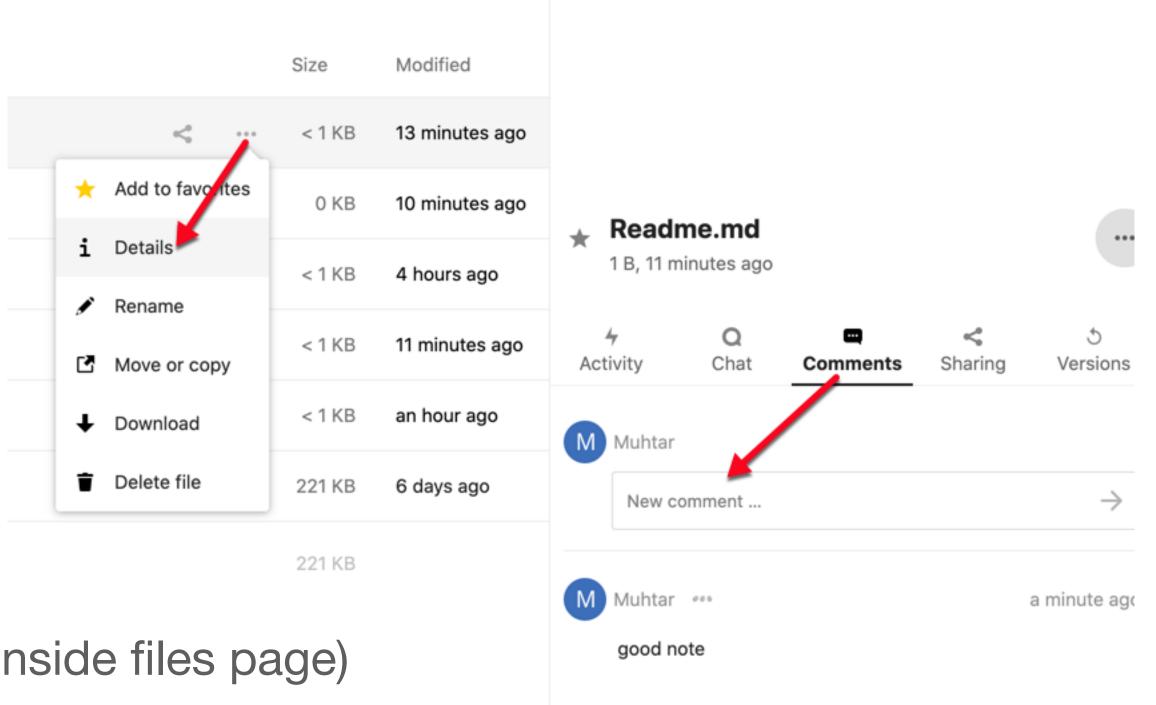
(Pre-condition: there should be at least 1 file is uploaded inside files page)



## AC #1 - verify users can write comments to files/folders.

- 1. Login as a user
- 2. Click action-icon from any file on the page
- 3. Click details
- 4. Write a comment inside to the input box
- 5. Click the submit button to post it
- 6. Verify the comment is displayed in the comment section.

(Pre-condition: there should be at least 1 file/foler is uploaded inside files page)



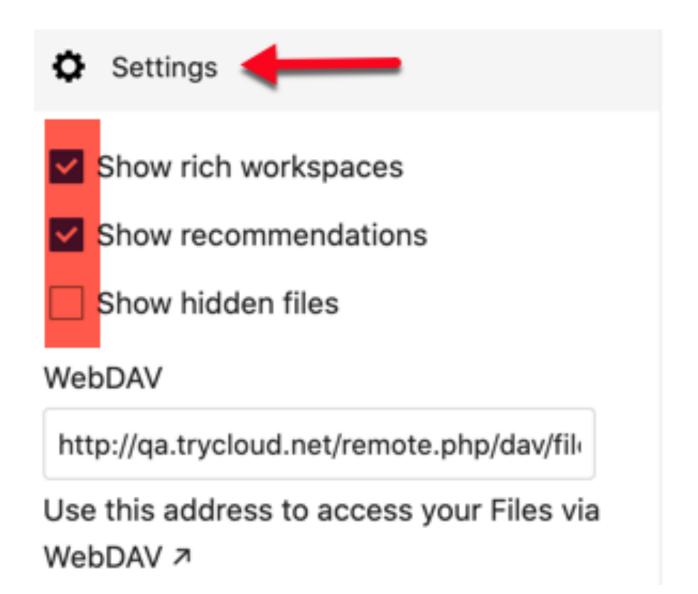


## AC #1 - verify users can change the app Settings.

- 1. Login as a user
- 2. Click Settings on the left bottom corner
- 3. Verify user can click any buttons.

#### AC #2- verify users see the app storage usage

- 1. Login as a user
- 2. Check the current storage usage
- 3. Upload a file
- 4. Refresh the page
- 5. Verify the storage usage is increased





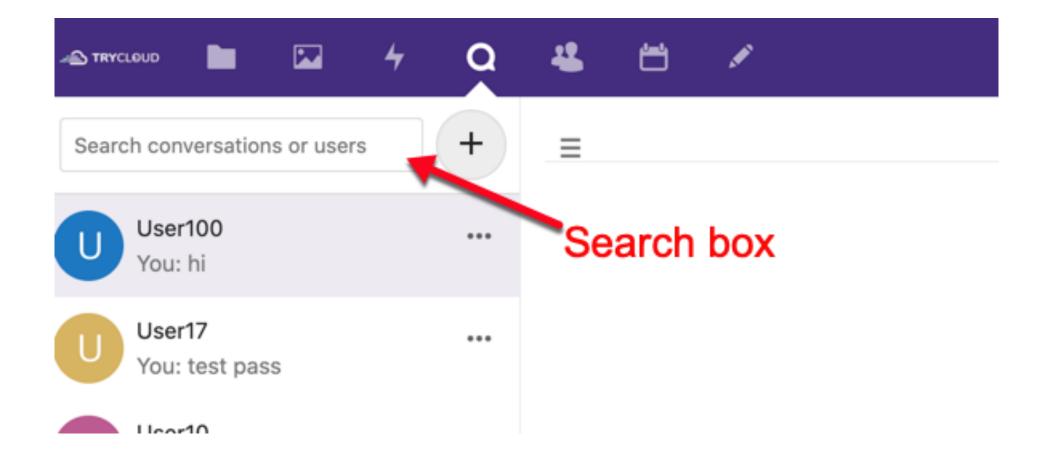


## AC #1 - verify users can access to Talks module

- 1. Login as a user
- 2. Click Talks module
- 3. Verify the page tile is Talks module's tile

## AC #2 - verify users can send message

- 1. Login as a user
- 2. Click Talks module
- 3. Search a user from search box on the left
- 4. Write a message
- 5. Click submit button
- 6. Verify the message is displayed on the conversation log



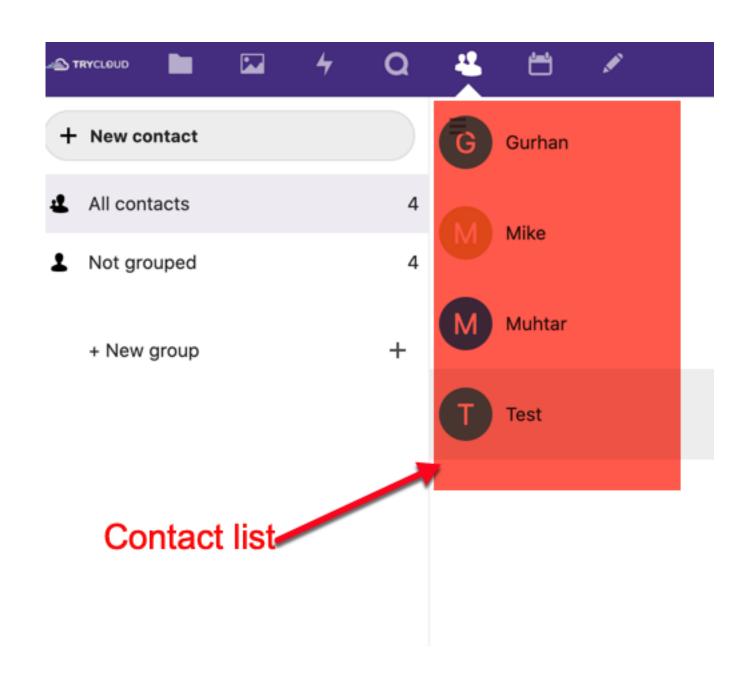


## AC #1 - verify users can access to Talks module

- 1. Login as a user
- 2. Click "Contacts" module
- 3. Verify the page tile is Contents module's tile

## AC #2 - verify users can add contacts

- 1. Login as a user
- 2. Click contacts module
- 3. Click "New Contact" button
- 4. Fill out the contact info like: Title, Phone, email, address, etc
- 5. Verify the contact name is added to the contact list

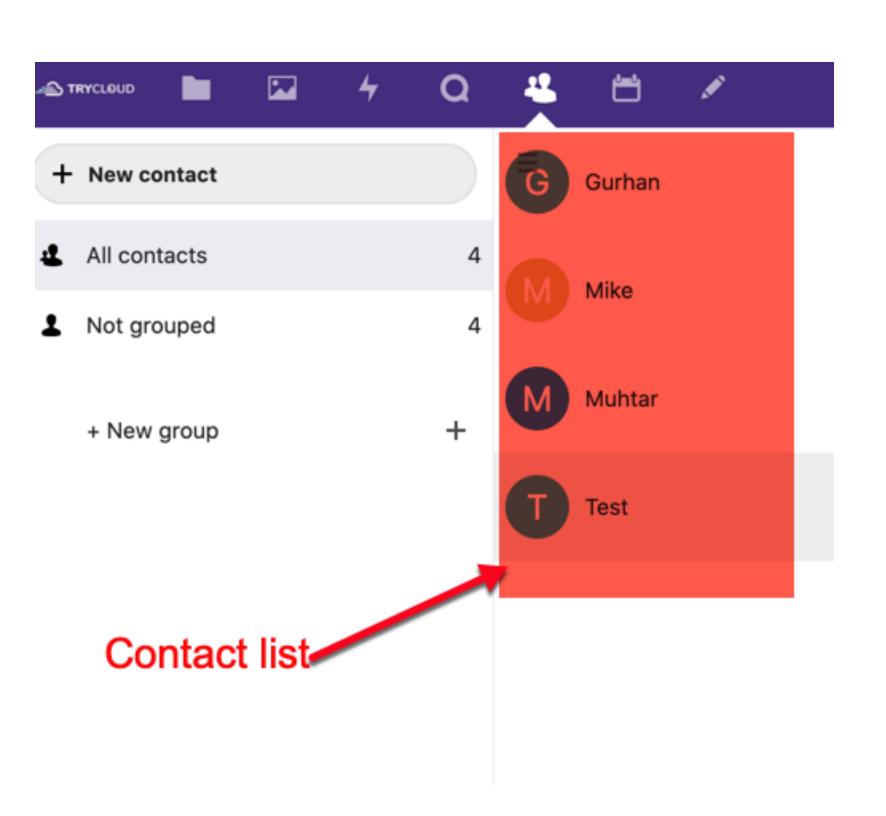




## AC #1 - verify users can see all the contact names on the contact list

- 1. Login as a user
- 2. Click contacts module
- 3. Verify the contact names are in the list

(Pre-condition: there should be at least 2 contact names are displayed On the contact list)





14. Story: As a user, I should be able to access search any item/ users from the homepage.

## Test case #1 - Verify users can search any files/folder/users from the search box.

- 1. Login as a user
- 2. Click magnifier icon Q on the right top
- 3. Search any existing file/folder/user name
- 4. Verify the app displays the expected result option

