This document contains test cases that covers and tests the UI/UX part of the application. The following test cases are written according to the Acceptance Criteria derived from the corresponding User Stories.
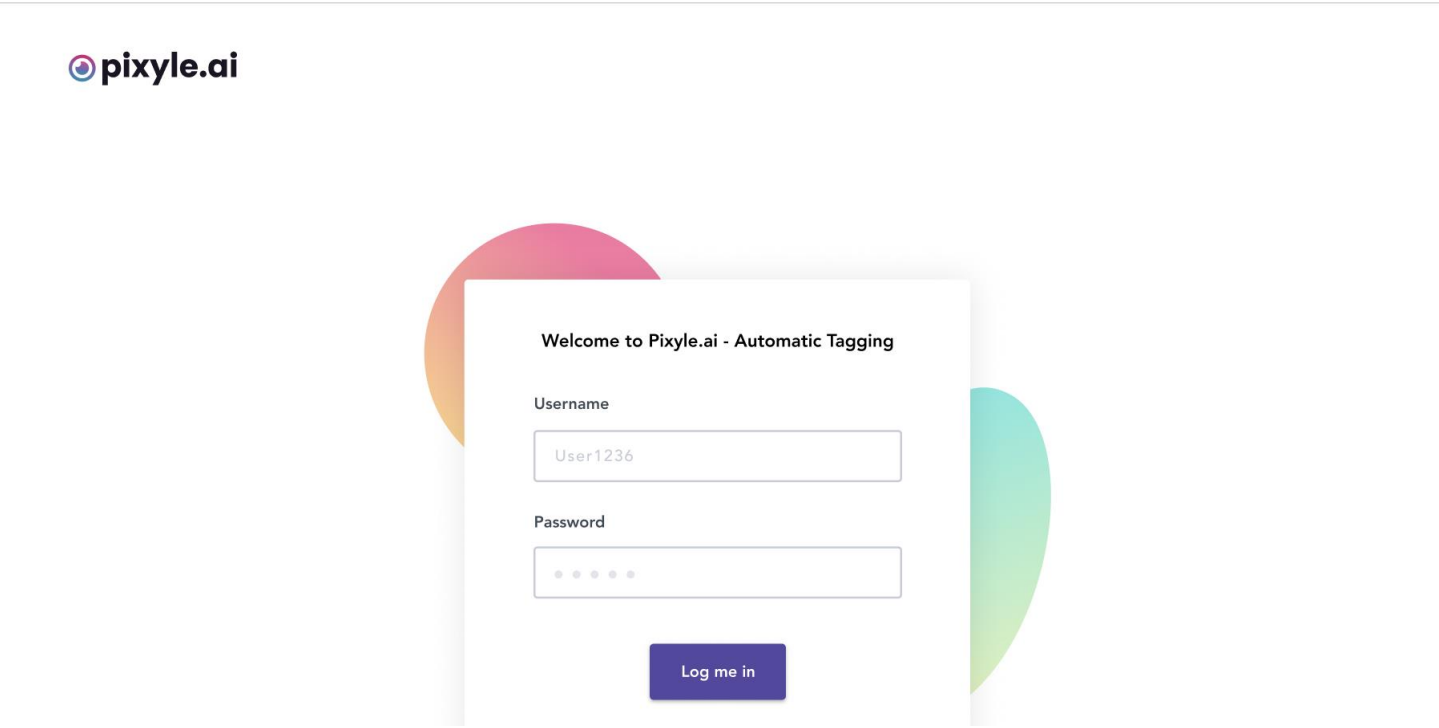
User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically follow a simple template:

As a < type of user >, I want < some goal > so that < some reason >

It's the product owner's responsibility to make sure a product backlog of agile user stories exists, but that doesn't mean that the product owner is the one who writes them. Over the course of a good agile project, you should expect to have user story examples written by each team member.

# US_001: Pixyle Login Page User Story:

| Description | **As a customer, I want to login with my credentials so I can securely access my online account** |
|---|---|
| Assigned To | PO (Product Owner) |
| State | Ready / In Design / Done / In Review |
| Priority | High |
| Risk | High |



## Acceptance Criteria:

| **Username** | Input type field with 256 max chars |
|---|---|
| **Password** | At least 7 chars, 1 number, 1 uppercase letter, 1 lowercase letter, 1 special char |
| **Log Me In** | Type Button |

All fields are required data fields - error alert should be shown if user clicks on "Log Me In" button without typing anything in the required fields.

Password entries will show the keystroke typed, and then after certain time (1s) the entered character is masked.

Green check mark next to 'Password' data field should be displayed that indicates to the user after typing password that the entry meets criteria

Red mark next to 'Password' data field should be displayed that indicates to the user after typing password that the entry does not meet criteria

After successful validation of all entered fields and on clicking "Log Me In" button, a confirmation message for successful login appears.

## US_002: Pixyle Reset Login Credentials:

| Description | **As a customer, I want to login with my credentials so I can securely edit/modify my online account** |
|---|---|
| Assigned To | PO (Product Owner) |
| State | Ready / In Design / Done / In Review |
| Priority | High |
| Risk | High |



## Acceptance Criteria:

| | |
|---|---|
| **Old Password** | At least 7 chars, 1 number, 1 uppercase letter, 1 lowercase letter, 1 special char |
| **New Password** | At least 7 chars, 1 number, 1 uppercase letter, 1 lowercase letter, 1 special char |
| **Repeat New Password** | At least 7 chars, 1 number, 1 uppercase letter, 1 lowercase letter, 1 special char |
| **Change Password** | Type Button |
| **Name** | Label (text) – non editable |
| **Last Name** | Label (text) – non editable |
| **Company** | Label (text) – non editable |
| **Business Model** | Label (text) – non editable |

'Old Password', 'New Password' and 'Repeat New Password'' entries should be masked
Password entries will show the keystroke typed, and then after certain time (1s) the entered character is masked.

If 'Old Password' entry does not match criteria specified and user hits Change Password, error alert "Password entry does not meet criteria" will be shown.

If entries in 'New Password' and Repeat New Password' do not match and user hits Change Password, error alert 'Password entries do not match' is shown.

After successful validation of all entered fields and on clicking Change Password button, message is shown indicating successful account modification.

# Test Suite 1: Pixyle Authentication

## Pixyle 1.1: Login with invalid credentials

| Description | This test script covers login with invalid credentials |
|---|---|
| Link to User Story | US_001 |
| Assigned To | QA Manual Tester |
| State | Ready / In Progress / Done / In Review |
| Priority | High |
| Risk | High |
| Status | PASS / FAIL |
| Product | Pixyle |
| Type | Regression |
| Automation | Automated / Can Not Be Automated / In Progress |
| Configuration | Windows 10 / Windows 8.1 / Windows 7 (XP not supported) MAC |
| Prerequisites | Chrome version x.x.x or above / Android Version x.x.x or above / IOS |

| Action | Expected Result |
|---|---|
| Open Chrome and enter **https://www.pixyle.ai/Login** | User successfully navigates and lands on page where a content area with title "Welcome to Pixyle.ai – Automatic Tagging", Username, Password and Log Me In button are shown. At the bottom of the form there is a text "Problem logging in? Contact Us so we can solve it right away" where the User is able to click on the link Contact Us. |
| Use {EMPTY} Username and Password and click on the Log Me In button | "Alert" appears notifying the User with the message "Incorrect Login." |
| Type any nonexistent user (ex. SomeUser) in the Username input field and click on the Log Me In button | "Alert" appears notifying the User with the message "Incorrect Login." |
| Use {EMPTY} Username with existing correct Password and click on the Log Me In button | "Alert" appears notifying the User with the message "Incorrect Login." |
| Type correct Username in the Username input field and wrong password in the Password field | "Alert" appears notifying the User with the message "Incorrect Login." |
| Type correct Username in the Username input field and add space on the end. Use correct password corresponding the User from above in the Password field | "Alert" pop up window appears notifying the User with the message "Incorrect Login." |
| *try any other combination of invalid login | / |

# Pixyle 1.2: Login with valid credentials

| Description | This test script covers login with valid credentials |
|---|---|
| Link to User Story | US_001 |
| Assigned To | QA Manual Tester |
| State | Ready / In Progress / Done / In Review |
| Priority | High |
| Risk | High |
| Status | PASS |
| Product | Pixyle |
| Type | Regression |
| Automation | Automated / Can Not Be Automated / In Progress |
| Configuration | Windows 10 / Windows 8.1 / Windows 7 (XP not supported) MAC |
| Prerequisites | Chrome version x.x.x or above / Android Version x.x.x or above / IOS |

| Action | Expected Result |
|---|---|
| Open Chrome and enter **https://www.pixyle.ai/Login** | User successfully navigates and lands on page where a content area with title "Welcome to Pixyle.ai – Automatic Tagging", Username, Password and Log Me In button are shown.<br>At the bottom of the form there is a text "Problem logging in? Contact Us so we can solve it right away" where the User is able to click on the link Contact Us. |
| Log In with Valid Credentials | User1236 is logged in successfully in the application and lands on the "My Collections" page.<br>Within the main panel there is a table of content with following columns that the user can/cannot sort/filter/search by:<br>    -    Folder Name<br>    -    Number of images<br>    -    Verified Images<br>    -    Competition<br>The User is able to delete an item. |
| At the left side off the page | My Collections, Account Settings tabs are present and Upload Images Button is enabled. |
| At the top of the page | Search by Product ID input filed is shown and a drop down list with the username of the current logged in user (in this case User1236) |

# Pixyle 1.3: Log Out

| Description | This test script covers Logging out current user and verification that the same has landed on the Login Page |
|---|---|
| Link to User Story | US_003 (tbd) |
| Assigned To | QA Manual Tester |
| State | Ready / In Progress / Done / In Review |
| Priority | High |
| Risk | High |
| Status | PASS |
| Product | Pixyle |
| Type | Regression |
| Automation | Automated / Can Not Be Automated / In Progress |
| Configuration | Windows 10 / Windows 8.1 / Windows 7 (XP not supported) MAC |
| Prerequisites | Chrome version x.x.x or above / Android Version x.x.x or above / IOS |

| Action | Expected Result |
|---|---|
| Open Chrome and enter https://www.pixyle.ai/Login | User successfully navigates and lands on page where a content area with title "Welcome to Pixyle.ai – Automatic Tagging", Username, Password and Log Me In button are shown. At the bottom of the form there is a text "Problem logging in? Contact Us so we can solve it right away" where the User is able to click on the link Contact Us. |
| Log In with Valid Credentials | User1236 is logged in successfully in the application and lands on the "My Collections" page. Within the main panel there is a table of content with following columns that the user can/cannot sort/filter/search by: <br> - Folder Name <br> - Number of images <br> - Verified Images <br> - Competition <br> The User is able to delete an item. |
| Click on the **User1236** (ddl item) on the top right corner of the application and click on Log Out | The User lands on the Login form and is logged out of the application successfully |
| Click on the back button of the browser | Ensure that the User is not able to navigate back to the last state/session of the application. |

# TABLE OF CONTENTS

The purpose of this test plan is to:

- Analyze the product

  - Who will use the website?
  - What is it used for?
  - How will it work?
  - What are software/ hardware the product uses?

- Design the Test Strategy
  - Define scope of testing
  - Identify Testing Type (Unit, API, Integration, System, Install/Uninstall, Agile Testing)
  - Document Risk and Issues (Risk is future's uncertain event with a probability of occurrence and a potential for loss. When the risk actually happens, it becomes the 'issue'.)
  - Create Test Logistics (Who and when will the test occur?)
- Define the Test Objectives
  - List all the software features (functionality, performance, GUI…) which may need to test.
- Define Test Criteria
  - *Suspension Criteria* (If your team members report that there are 40% of test cases failed, you should suspend testing until the development team fixes all the failed cases.)
  - Exit Criteria (The exit criteria are the targeted results of the test and are necessary before proceeding to the next phase of development. Example: 95% of all critical test cases must pass.)
- Resource Planning
  - Resource plan is a **detailed summary** of all types of resources required to complete project task. Resource could be human, equipment and materials needed to complete a project
  - System Resource (Server, Test Tool, Network, Computer, Mobile devices etc.)
- Plan Test Environment
  - A testing environment is a setup of software and hardware on which

Risk)
- Determine Test Deliverables
  - **Test Deliverables** is a list of all the documents, tools and other components that has to be developed and maintained in support of the testing effort.
  - **Before Testing** (Test plans document, Test cases documents and Test Design specifications.)
  - **During Testing** (Test Scripts Simulators, Test Data, Test Traceability Matrix Error logs and execution logs.)
  - **After Testing** (Test Results/reports, Defect Report, Installation/ Test procedures guidelines and Release notes)

# 2 TEST ITEM

## 2.1 PROJECT DESCRIPTION

We've built image recognition software that enables fashion retailers to **deliver great online customer experiences.** Our AI technology helps shoppers navigate e-commerce sites in a much more intuitive and convenient way, using images instead of text. We're offering AI solutions ranging from visual search, similar recommendations, automatic tagging to a fully customizable business intelligence and data visualization platform. Pixyle's technology is fit for the future: it's accurate, powerful and integrates seamlessly with any tech stack. Interface for automatic tagging used by fashion e-commerce platforms to ease the tagging process.

## 2.2 ITEMS TO BE TESTED / NOT TO BE TESTED

| Item to Test | Test Description | Test Date | Responsibility |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 2.3 ITEMS TO BE TESTED

| Item Not to Test | Comment |
|---|---|
|  |  |

- Proactive: This approach defines and prepares the approach or way to carry out the testing procedures, as soon as possible, even before the development of the initial software build. This technique may be seen as a preventive approach, in order to find and fix defects.

- **Reactive**: Contrary to proactive approach, reactive approach involves the defining and creation of test procedures, only after the completion of the development phase, i.e. approach could only be initiated, when the development of a software product has been finished. (Pixyle.ai case)

The scope of this document is to test the UI/UX part of the application as well as boundary values, equivalence partitioning, smoke, positive/negative testing, functional and regression testing. Manual test will be reviewed and planned for automation once approved (by Manager/Test Lead) and ready to be automated. Automated test scripts will run on schedule, on build or on demand. The following test cases are written according to the Acceptance Criteria derived from the corresponding User Stories (see User Stories Acceptance Criteria folder).

## 2.5 TEST PASS / FAIL CRITERIA

The test has passed only if the expected results from the test case are correct and shown in reasonable time (defined by PO or stake holder), while failed test will be the one that do not fulfill or partially verify the expected results from the tests case. That concludes that the application does not meet the acceptance criteria and the requirements from the client.

## 2.6 TEST ENTRY / EXIT CRITERIA

Generally, the entry and exit criteria are determined for the four test levels of Software Testing:

- Unit Testing
- Integration testing
- System Testing
- Acceptance Testing

**Screenshot 1 - Pixle_Navigation.robot**

Tabs: Settings.robot | Validations.robot | Generic_Commands.robot | Navigation.robot | Pixle_Navigation.robot

```robot
*** Settings ***

Resource    ../Keywords/Settings.robot
Resource    ../Keywords/Navigation.robot
Resource    ../Keywords/Generic_Commands.robot
Resource    ../Keywords/Validations.robot

*** Test Cases ***

Pixyle Landing Page
    [Tags]    Navigation
    Run Pixyle in web browser

    #Pixyle logo i shown on landing page
    Ensure that element is visible    //a[@class='l-brand w-nav-brand w--current']

    #h1 is shown with text Visual AI for Fashion Retail
    Ensure that element is visible    //h1[@class='c-hero-heading'][.='Visual AI for Fashion Retail']

    #Your bussiness email form and show me more button
    Ensure that element is visible    //*[@id='email'][@placeholder='Your business e-mail']
    Ensure that element is visible    //input[@value='Show me more']

    #Solutions tab item (hover)
    Ensure that element is visible    //nav[@role='navigation'][@class='nav-menu w-nav-menu']//div[@class='text-block-10'][.='Solutions']


    [Teardown]    Close all open browsers and clear cookies
```

Console:
```
<terminated> Pixyle [Program] C:\Program Files (x86)\Python37-32\Scripts\robot.exe
1 test total, 1 passed, 0 failed
==============================================================================
Output:  C:\workspace\Pixyle\output.xml
Log:     C:\workspace\Pixyle\log.html
Report:  C:\workspace\Pixyle\report.html
```



**Screenshot 2 - Navigation.robot**

Tabs: Settings.robot | Validations.robot | Generic_Commands.robot | Navigation.robot | Pixle_Navigation.robot

```robot
*** Settings ***

Resource    ../Keywords/Settings.robot
Resource    ../Keywords/Generic_Commands.robot
Resource    ../Keywords/Validations.robot

*** Keywords ***

Accept Automation Alert
    SeleniumLibrary.Handle Alert
    #${message} =    Handle Alert

Navigate to [${LEFTMENUITEM}]
    SeleniumLibrary.Click Element    //div[@class='x-text-el'][.='${LEFTMENUITEM}']/../../..
    #SeleniumLibrary.Go To  https://dvmax-cloud-qa.appspot.com/#${LEFTMENUITEM}
    #SeleniumLibrary.Reload Page
    Wait for Ajax

Navigate to the sub-item [${LEFTMENUSUBITEM}]
    SeleniumLibrary.Click Element    //div[@class='x-treelist-item-text x-treelist-item-text-noicon'][.='${LEFTMENUSUBITEM}']

Navigate to DVMAX News page
    Ensure that element is visible    //a[@data-componentid='tab-1072']
    SeleniumLibrary.Click Element    //a[@data-componentid='tab-1072']
    Wait for Ajax
    Ensure that the element is not visible    //div[@class='loader']
    SeleniumLibrary.Select Frame    //iframe[@class='x-component x-fit-item x-component-default']
    Ensure that element is visible    //html[contains(@class, 'js flexbox canvas')]
    ${frames} =    Current Frame Contains    ${DVMAXTITLE}
    ${frames} =    Current Frame Contains    ${DVMAXDESC}
    ${frames} =    Current Frame Contains    ${DVMAXBTN}
```

Console:
```
<terminated> Pixyle [Program] C:\Program Files (x86)\Python37-32\Scripts\robot.exe
1 test total, 1 passed, 0 failed
==============================================================================
Output:  C:\workspace\Pixyle\output.xml
Log:     C:\workspace\Pixyle\log.html
Report:  C:\workspace\Pixyle\report.html
```

Top editor — Generic_Commands.robot:

```robotframework
*** Settings ***
Resource    ../Keywords/Settings.robot
Resource    ../Keywords/Navigation.robot
Resource    ../Keywords/Validations.robot
Resource    ../Keywords/Context.robot

*** Variables ***
${IsAjaxComplete}    ${EMPTY}

*** Keywords ***

Open Browser And Go To Login Page
    SeleniumLibrary.Open Browser    ${ServerName}    ${Browser}
    SeleniumLibrary.Maximize Browser Window
    SeleniumLibrary.Set Selenium Speed    0

Run Pixyle in web browser
    Open Browser And Go To Login Page
    Generic_Commands.Wait for Ajax

Login as Administrator
    #aSeleniumLibrary.Scroll Element Into View    //span[@data-ref='btnInnerEl'][contains(.,'Login')]
    SeleniumLibrary.Input Text    //span[contains(.,'Password')]/../../div/input    foobar123
    Ensure that element is visible    //div[@class='x-text-el'][.='Login']
    SeleniumLibrary.Click Element    //div[@class='x-text-el'][.='Login']/../../..

Login as [${USERN}] with password [${PSWD}]
    #SeleniumLibrary.Scroll Element Into View    //div[@class='x-text-el'][.='Login']/../../..
    SeleniumLibrary.Input Text    //span[contains(.,'Username')]/../../div/input    ${USERN}
    SeleniumLibrary.Input Text    //span[contains(.,'Password')]/../../div/input    ${PSWD}
```

Console:

```
<terminated> Pixyle [Program] C:\Program Files (x86)\Python37-32\Scripts\robot.exe
1 test total, 1 passed, 0 failed
==============================================================================
Output:  C:\workspace\Pixyle\output.xml
Log:     C:\workspace\Pixyle\log.html
Report:  C:\workspace\Pixyle\report.html
```

Bottom editor — *Settings.robot:

```robotframework
*** Settings ***
Library     SeleniumLibrary    timeout=60 s
#Library     Collections
#Library     RequestsLibrary

*** Variables ***
${ENVIRONMENT}        Pixyle Cloud

${BROWSER}            chrome
${ServerName}         https://www.pixyle.ai/
${DELAY}              0
${SETPIXYLEFRAME}     //iframe[@class='x-component x-fit-item x-component-default']
${PIXYLETITLE}        Visual AI for Fashion Retail
${APPTDATE}
```

Console:

```
<terminated> Pixyle [Program] C:\Program Files (x86)\Python37-32\Scripts\robot.exe
1 test total, 1 passed, 0 failed
==============================================================================
Output:  C:\workspace\Pixyle\output.xml
Log:     C:\workspace\Pixyle\log.html
Report:  C:\workspace\Pixyle\report.html
```

Top window - Eclipse IDE with Details.robot open:

Tabs: *Settings.robot | Validations.robot | Generic_Commands.robot | Navigation.robot | Pixyle_Navigation.robot | Details.robot

Package Explorer:
- DVMAX_Cloud
  - Autobots
    - 1_Home
    - 2_Clients
    - 3_Patients
      - Overview
      - Details.robot
      - Docs.robot
      - History.robot
      - Labs.robot
      - Requisitions.robot
      - ToDos.robot
      - WP.robot
    - 4_Scheduling
    - 5_Messaging
    - 6_ToDos
    - 7_Whiteboard
    - 8_Settings
    - Authentication
    - Medical Record
      - AcceptAlert.robot
      - DVMAX_Navigation.robot
      - HTTP_demo.robot
    - Keywords
    - RemoteSystemsTempFiles
    - Tests
  - debug.log
  - log.html
  - output.xml
  - report.html
  - runDVMAX.txt
- Pixyle
  - Keywords
    - Context.robot
    - Generic_Commands.robot
    - Navigation.robot
    - Settings.robot
    - Validations.robot

```
*** Settings ***

Resource    ../../Keywords/Settings.robot
Resource    ../../Keywords/Navigation.robot
Resource    ../../Keywords/Generic_Commands.robot
Resource    ../../Keywords/Validations.robot

*** Test Cases ***

Patients
    [Tags]    Patients
    Run DVMAX in web browser
    Ensure User is on DVMAX Login Page
    Login as Administrator
    Left Menu items are available
    Verify that the User has landed on the Home page
    Navigate to [Patients]
    Ensure Title on page is [Patients]
    Open the record and validate correct parameters are present on the screen    388083
    Within Clients panel I navigate to [Details]

Details-labels
    [Tags]    Patients
    Ensure label with text [Name] is present
    Ensure label with text [Species] is present
    Ensure label with text [Breed] is present
    Ensure that element is visible    //label[.='M']
    Ensure that element is visible    //label[.='F']
    Ensure that element is visible    //label[.='Spayed']
```

Console:
```
<terminated> Pixyle [Program] C:\Program Files (x86)\Python37-32\Scripts\robot.exe
Pixyle Cloud                                                              | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
==============================================================================
Output:   C:\workspace\Pixyle\output.xml
Log:      C:\workspace\Pixyle\log.html
Report:   C:\workspace\Pixyle\report.html
```

Bottom window - Pixyle Cloud Report (report.html):

file:///C:/workspace/Pixyle/report.html

# Pixyle Cloud Report

Generated
20200111 22:48:35 UTC+01:00
16 hours 38 minutes ago

## Summary Information

| | |
|---|---|
| Status: | All tests passed |
| Start Time: | 20200111 22:48:06.622 |
| End Time: | 20200111 22:48:35.002 |
| Elapsed Time: | 00:00:28.380 |
| Log File: | log.html |

## Test Statistics

| Total Statistics | Total | Pass | Fail | Elapsed | Pass / Fail |
|---|---|---|---|---|---|
| Critical Tests | 1 | 1 | 0 | 00:00:27 | |
| All Tests | 1 | 1 | 0 | 00:00:27 | |

| Statistics by Tag | Total | Pass | Fail | Elapsed | Pass / Fail |
|---|---|---|---|---|---|
| Navigation | 1 | 1 | 0 | 00:00:27 | |

| Statistics by Suite | Total | Pass | Fail | Elapsed | Pass / Fail |
|---|---|---|---|---|---|
| Pixyle Cloud | 1 | 1 | 0 | 00:00:28 | |

## Test Details

Totals | Tags | Suites | Search

Type:
○ Critical Tests
○ All Tests

Bottom Console:
```
<terminated> Pixyle [Program] C:\Program Files (x86)\Python37-32\Scripts\robot.exe
Pixyle Cloud                                                              | PASS |
1 critical test, 1 passed, 0 failed
```

Package Explorer (bottom window):
- DVMAX_Cloud
- Pixyle
  - Keywords
    - Context.robot
    - Generic_Commands.robot
    - Navigation.robot
    - Settings.robot
    - Validations.robot
  - Tests
    - Authentication.robot
    - Pixyle_Navigation.robot
  - log.html
  - output.xml
  - report.html
  - runPixyle.txt