Paylocity Benefits Dashboard Test Report

# manual testing Bug report

## Project Information

|  |  |  |  |
| --- | --- | --- | --- |
| UI Website/URL | https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/Benefits | Browsers | Google Chrome, Microsoft Edge |
| Test Engineer | Jeffrey Cerminaro | Test Targets | API, UI |
| Report start date | 3/2/2022 | Report end date | 3/4/2022 |

## Test # 1

Objective: ensure that the input format has proper validations while performing data entry.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on add employee button | Add User form is displayed | Pass |
| Enter numbers and symbols for First Name field | Numbers and symbols not allowed to type | Fail |
| Enter numbers and symbols for Last Name field | Numbers and symbols not allowed to type | Fail |
| Save added employee | User should not be allowed to save with symbols or numbers | Fail |
| Verify if employee was added using a get request in postman | User should not be added | Fail |
| Bug Summary: No validation performed on format for first name and last name fields on adding employee in the UI. Employee should not be able to allowed to be saved without validating a correct format. See screenshot: |  |  |

## Test # 2

Objective: ensure that employee full names are unique or have a way to difference between them

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Follow steps to add a user once successfully | User saved | Pass |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on add user button | Add employee form displayed | Pass |
| Enter the first name identical to the previously saved employee | Name entered in input | Pass |
| Enter the last name identical to the previously saved employee | Name entered in input | Pass |
| Save added employee | User should not be saved if first and last name are identical to existing records or a new field allowing to identify as unique should be present | Fail |
| Verify if employee was added using a get request in postman | Employee should not be added | Fail |
| **Bug summary:** employees should have at least one way to be identified as unique within the list of all employees, duplicate names and last names are being allowed to be saved. |  |  |

## Test # 3

Objective: ensure all employee fields in the table displayed in the UI are in the right column and the value matches the title.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on add user button | Add employee form displayed | Pass |
| Enter the first | Name entered in input | Pass |
| Enter the last name | Name entered in input | Pass |
| Save added employee | Employee should be registered | Pass |
| Record should be displayed in the table | New employee should be present in the table after refresh | Pass |
| Verify if employee was added using a get request in postman | Employee should be added | Pass |
| Verify if fields of employee record are matching the actual data presented in postman via get request | Al fields should be in the right column | Fail |
| **Bug summary**: first name and last name columns seem to be switched and “Last Name” value is actually the first name in the actual recorded data. The same way the “First Name” value is actually the last name in the actual recorded data. |  |  |

## Load Employees Table Definition

## Test # 4

Objective: on benefit cost calculation verify the result values are correct up to the entirety of the decimal precision places.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on add user button | Add employee form displayed | Pass |
| Enter Name Data | Name accepted in the input | Pass |
| Enter the dependent quantity | Numbers accepted in the input | Pass |
| Save added employee | Employee should be registered | Pass |
| Verify manually the operations performed by using the following formula:  Pay Period Individual Benefit Cost = PPIBC = 1000  Pay Period Dependent Benefit Cost = PPDBC = (#Dependents\*500)  **Gross Pay – (PPIBC + PPDBC/26)** | Manual result and system result should match | Fail |
| **Bug Summary:** the benefit cost while doing manually does not match after a certain decimal place with what the system is returning using the formula.  PPIBC = 1000  PPDBC = 3 \* 500  Calculator => 2000 – (1000+1500/26) = 96.1538**461**  System => 96.1538**544** |  |  |

## Test # 4

Objective: on net pay calculation verify the result values are correct up to the entirety of the decimal precision places.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on add user button | Add employee form displayed | Pass |
| Enter Name Data | Name accepted in the input | Pass |
| Enter the dependent quantity | Numbers accepted in the input | Pass |
| Save added employee | Employee should be registered | Pass |
| Verify manually the operations performed by using the following formula:  Gross Pay – Benefits cost | Manual result and system result should match | Fail |
| **Bug Summary:** the net pay value while doing manually does not match after a certain decimal place with what the system is returning using the formula.  Calculator => 2000 – 96.1538544 = 1,903.846145 🡪 if we round up 1,903.8461**5**  System => 1903.8461**9** |  |  |

## Test # 5

Objective: verify the input elements display when the input validations are violated as to know why it failed

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on add user button | Add employee form displayed | Pass |
| Enter Name Data | Name accepted in the input | Pass |
| Enter letters in the Dependent field | Input not accepted | Pass |
| Verify a validation notification when violated | Violation reason displayed | Fail |
| **Bug Summary**: user’s input is invalid and the system correctly prevents the input to be populated, although it does not display any reason or violation notification so the user can fix the input. |  |  |

## Test # 6

Objective: ensure the content is aligned correctly and adjusted according to the size of the inputs in data table when editing content.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on edit employee button on the far right | Edit employee form displayed | Pass |
| Edit with a long string as the name field 100 characters or more | Input should be accepted | Pass |
| Edit with a long string as the last name field 100 characters or more | Input should be accepted | Pass |
| Verify a content adjusts accordingly with the data table | Table adjusts size to content | Fail |
| **Bug Summary**: user input with a large number of characters disrupts the data table and it brings it out of proportions. The data table overlaps and becomes distorted. Ideally all components that are part of the table should adjust – grow or shrink with the content maintaining the same proportions.  Possible problem: one of the containers has a side that is static or has reached the max length set and will not adjust to width.  Possible solution: make the container width adjust to fit. |  |  |

## Test # 7

Objective: ensure the employee data is accessible when the data is extensive.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Search for a specific employee | Employee should be accessible | Need Improvement |
| **Issue/Enhancement Summary**: employee list can get long and to scroll down to a specific entry it will be tedious and time consuming. You can apply the browser string find, but this is not specific. Possible enhancement might include a search option by any method. Filters and search options are an easy way to improve a simple time consuming task. | |  |

## Test # 8

Objective: ensure the employee data interaction is accessible when the data is extensive.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Add employee | Add entry button should be displayed | Need Improvement |
| **Issue/Enhancement Summary**: employee list can get long and you need to scroll down to the bottom to find the add entry button which will be tedious and time consuming. Possible enhancement might include pagination and setting the button to add an entry on the top of the table or static on the bottom. | |  |

## Test # 9

Objective: ensure the employee data interaction is accessible when the data is extensive.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Search last entry of the table | Find last entry in table | Need Improvement |
| **Issue/Enhancement Summary**: employee list can get long and you need to scroll down to the bottom to find the last entry in the table which will be tedious and time consuming. Possible enhancement might include sorting options. | |  |

## Test # 9

Objective: ensure a proper user session management when session expires.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Wait for session to expire | Notification should display and user should be logged out | **Fail** |
| Verify user cannot interact with web application until logged in again | Ideally web application should redirect to login page | **Fail** |
| **Bug Summary**: there is no way the user will know that the session will expire as the web application does not notify or display a label with any warning. The session expires, but the user is not logged out, instead the table is still visible although if you refresh the page the data table is empty. The user definitely needs to be logged out and redirected to the login with previous notification or optional extension of the session if required. Without this option there is no way the user can login again as the page does not have a login option. Refreshing the page only returns to the expired session obviously. | |  |

## Test # 10

Objective: ensure the content is aligned correctly and adjusted according to the size of the inputs in data table when interacting with the browser window

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Reduce window to a smaller size | Window reduced in size | Pass |
| Verify a content adjusts accordingly with the data table | Table adjusts size to content | Fail |
| **Bug Summary**: The data table overlaps and becomes distorted. Ideally all components that are part of the table should adjust – grow or shrink with the content maintaining the same proportions.  Possible problem: one of the containers has a side that is static or has reached the max length set and will not adjust to width.  Possible solution: make the container (div) width adjust to fit. |  |  |

## Test # 10

Objective: ensure a proper user session management with multiple sessions from same user.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in chrome browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Open new incognito session and open website in chrome browser | Navigate to login page | Pass |
| Log in with provided user | Web application will only allow one session at a time | Fail |
| Open new Microsoft Edge session and open website in edge browser | Navigate to login page | Pass |
| Log in with provided user | Web application will only allow one session at a time | Fail |
| **Bug Summary**: multiple sessions are being allowed for the same user when using different browser users or different browsers at the same time. For best practices only one session should be allowed at a time for the same user. |  |  |

## Test # 10

Objective: verifying behavior when adding employees with a large number of dependents. Exploring the validations set for number of dependents and the benefits cost limits.

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in chrome browser | Navigate to login page | Pass |
| Log in with provided user | User logs into the data entry form | Pass |
| Click on add employee button | Add employee form should be displayed | Pass |
| Enter name information | Name and last name should be accepted in input | Pass |
| Access the debug tools for chrome (Press F12) | Debug tools mode should be displayed | Pass |
| Enter a number of 6 digits in dependents | Accept number in input | Pass |
| Submit the new employee request | Employee should not be added | Pass |
| Verify reason code for employee not being added | Reason code or notification should be displayed | Fail |
| Verify reason code in the console of the debug tools | Reason code and error description displayed | Needs Improvement |
| **Bug/Enhancement Summary**: there is no error notification or reason code for the employee not being added within the UI. There is a reason code of 401 in the debug tool console but there is no error description of what actually happened.  After some experimentation the following results were obtained:  When adding a new employee with any number of dependents greater than the specified number it will generate 🡪 Error 401  When updating an existing employee with a number between 3 and 5 digits in the dependents field it produces 🡪 Error 400  When updating an existing employee with a number greater than 5 it produces 🡪 Error 500  Although no actual reason code is displayed in the UI that will indicate the user how he can enter a valid number. |  |  |

## Test # 11

Objective: ensure a proper user session management when the wrong user tries to log in

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in chrome browser | Navigate to login page | Pass |
| Log in with wrong user | Error notifies that wrong user and password has been entered | Fail |
| **Bug/Enhancement Summary**: when trying to log in as a wrong user there is no notification that indicates that its an incorrect user it only redirects the user to an error page with an internal error notification. The error layer should start with reporting what the exact mistake the user did. |  |  |

## Test # 12

Objective: ensure a proper user session management and login

|  |  |  |
| --- | --- | --- |
| Step | Expected | Result |
| Open website in chrome browser | Navigate to login page | Pass |
| Log in with wrong user | User is not logged in | Pass |
| Verify user can reset or remember the login information | User can reset password | Needs Improvement |
| **Issue/Enhancement Summary**: when user forgets password there is no way the system allows the user to log in. An improvement could be adding the possibility to log in with an alternative method where the user can reset or recuperate its password. |  |  |