BDD – Behaviour Driven Development

* Understanding requirement

Structured Scenario:-

Scenario: Login to the System

Given – pre-requisite for that scenario

When – actions/operation

Then – verification

Benefits

* Living Documentation
* Enables good communication between non-techical and techincal team
* Documentation will guide the product owners, end users to understand the application.
* Focus on end-users experience

Where to apply BDD?

1. Everyone knows about the requirement.
2. Some of the team members knows about the requirement.
3. Only top level management knows about the requirement.
4. Somewhere some other organization developed it and it is new to your organization
5. Research and then need to develop it.

ATDD – Acceptance Test Driven Development (Modified form of BDD)

* For acceptance testing

BDD concepts

Steps to create BDD project using Specflow

1. Extension -> specflow
2. Create Specflow project
3. Create or use Feature folder for the feature file.
4. Right click on Feature folder 🡪 New item 🡪 specflow 🡪 provide file name Login.feature
5. Architecture of Specflow (to achieve BDD concept in C#)

Feature (.feature) 🡪 Step Definition (.cs)

1. Create feature file
2. Add feature header
3. Add feature description
   1. Unformatted description

I want to login to connect with people

* 1. Formatted description

In order to [Business goal]

As a [Role]

I want to [visible change in the application]

1. Create scenario and provided scenario title
   1. Provide minimum one given,when, then.
2. Step parameterization – passing data in single/double quotes
3. Scenario Outline – reduce duplicates in feature file. **One scenario – run with multiple set of test data**
4. Background – repeated given
5. Datatable – send tabular data in one step.
6. Hooks
   1. [BeforeScenario] -- runs before each scenario
   2. [AfterScenario] -- runs after each scenario wether scenario passed or fail.
7. Tags
   1. Can be added at feature or scenario level.
   2. If it mentioned in feature level then all scenario will be included for that tag.
8. To ignore the scenario or feature – use @ignore
9. livingdoc
   1. Install

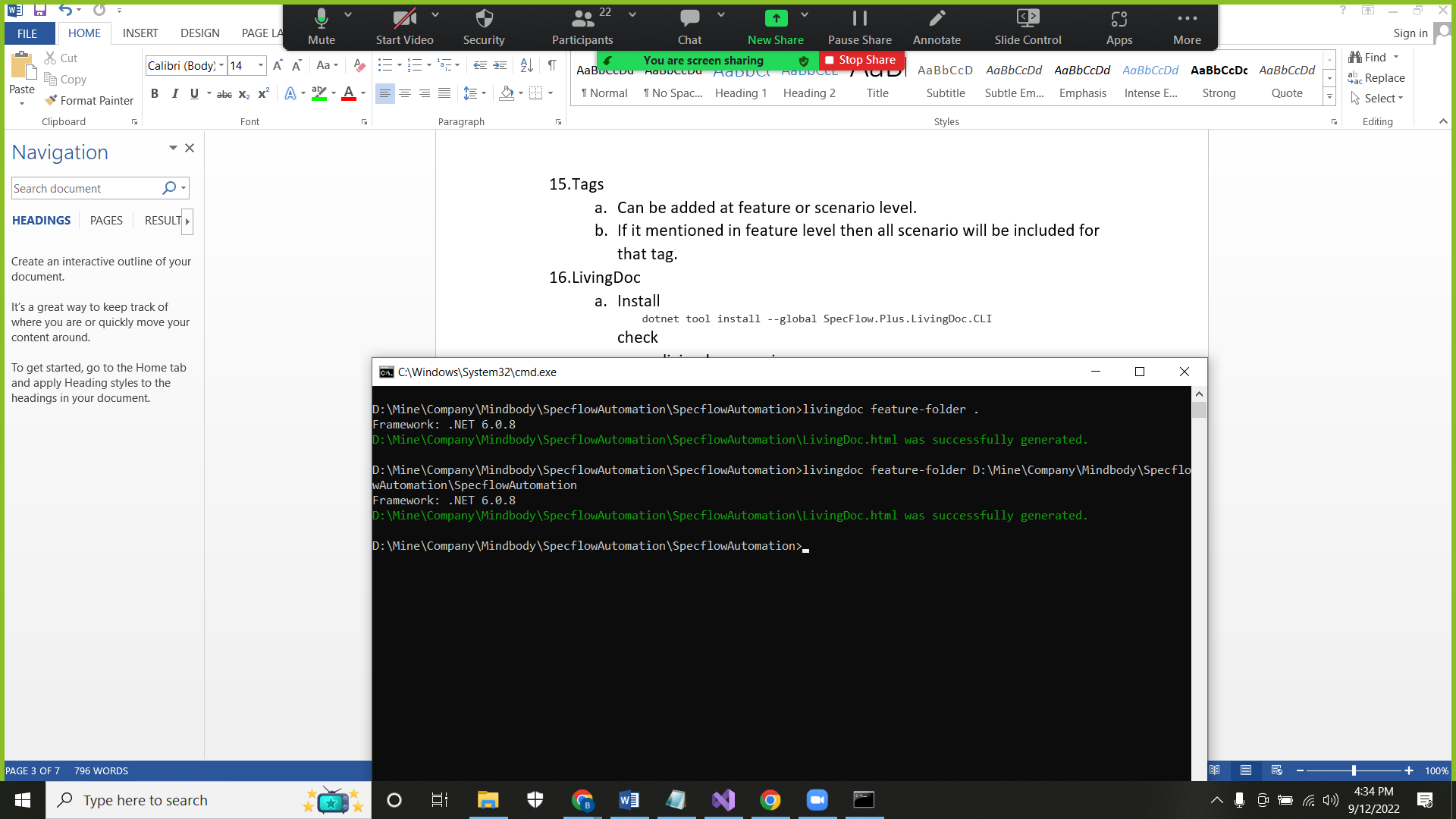
dotnet tool install --global SpecFlow.Plus.LivingDoc.CLI

check

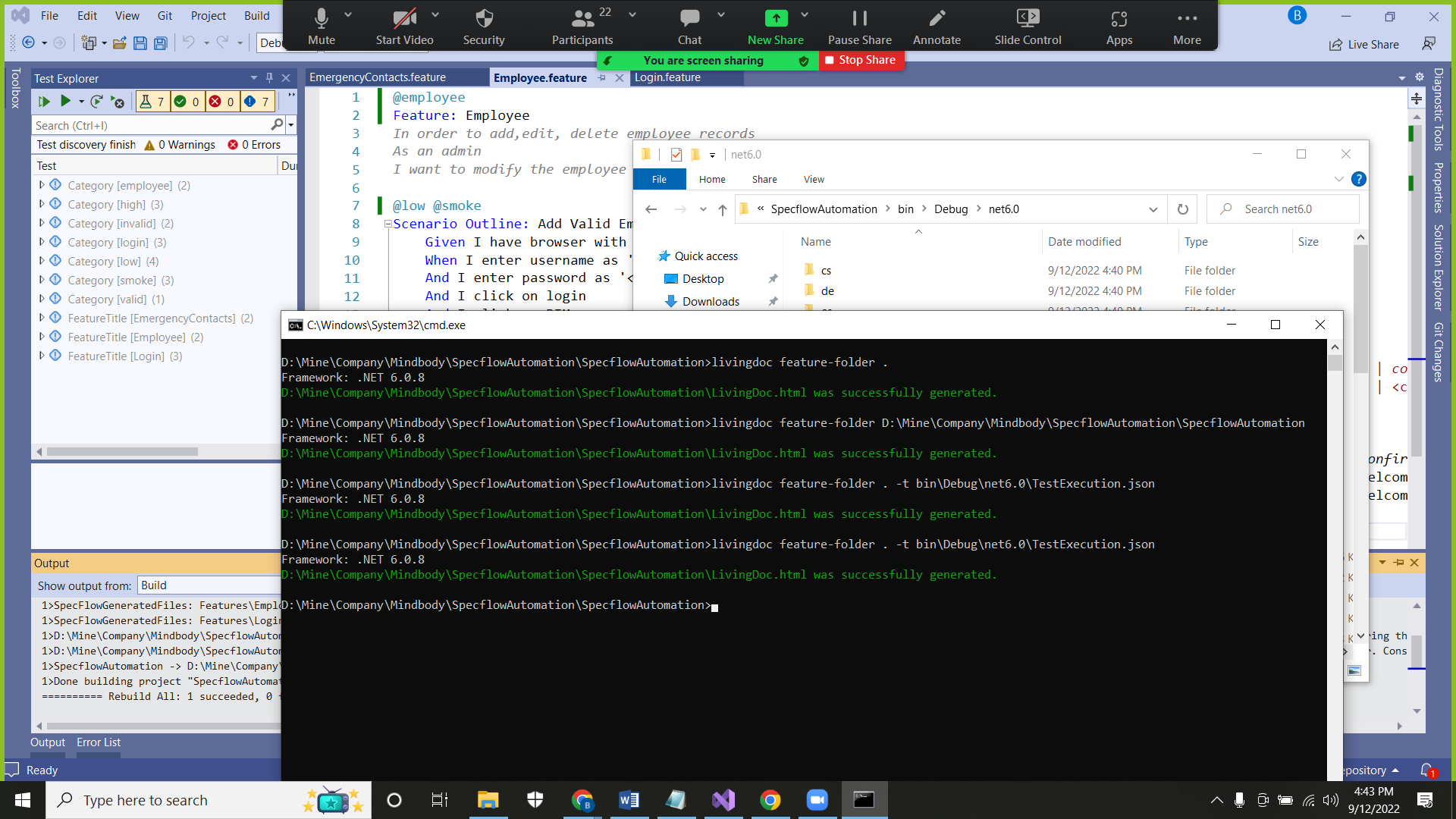
livingdoc --version

* 1. Generate reports for **feature folder.**

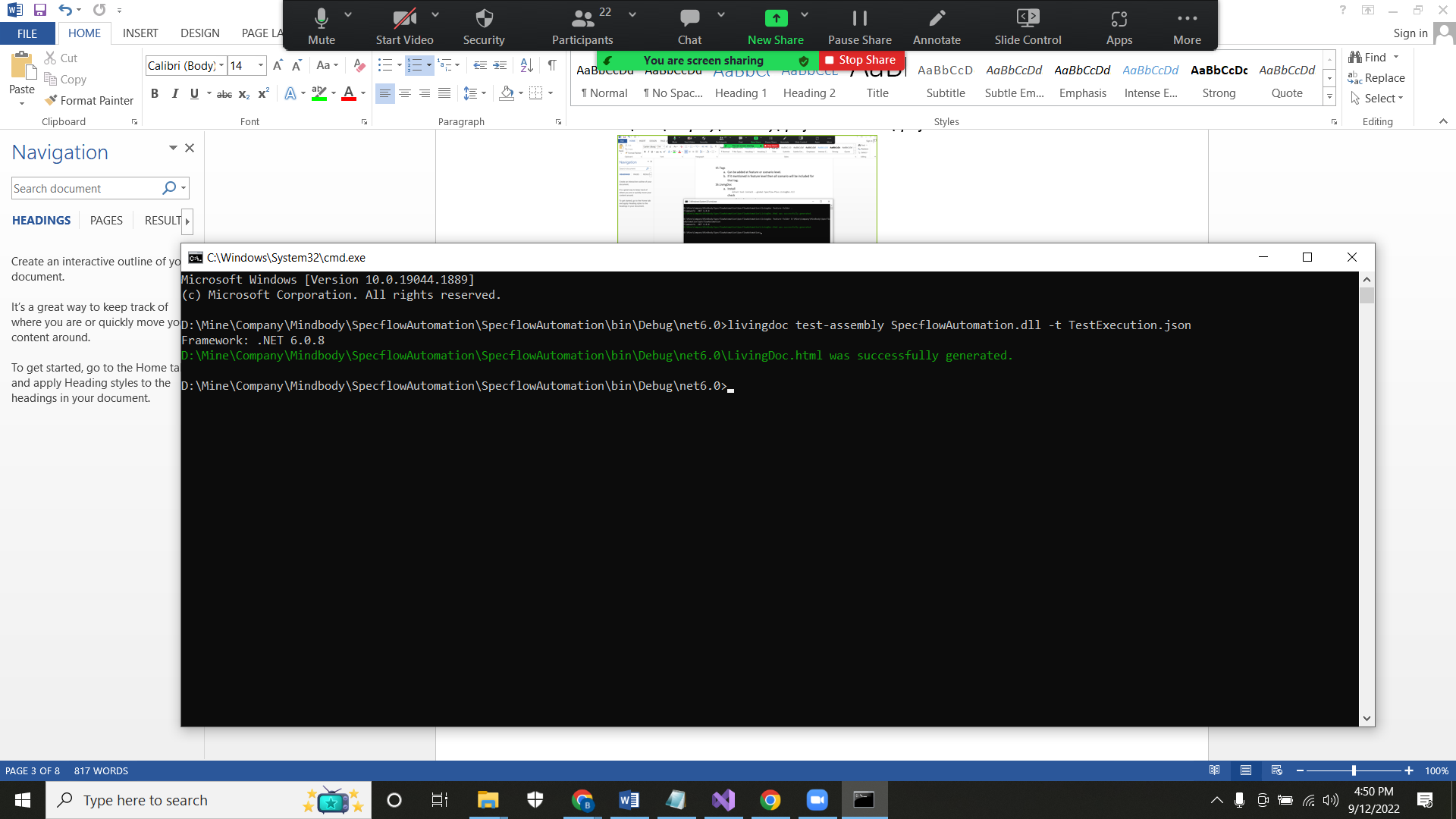
***livingdoc feature-folder D:\Mine\Company\Mindbody\SpecflowAutomation\SpecflowAutomation***



* 1. Generate reports for **feature folder with TestExecution.json report**



* 1. Generate reports for test-assembly **.dll with TestExecution.json report**



BDD – Important concepts – Improve the readability:-

1. Step parameterization – parametering the data so same step defn can be used
2. Scenario Outline – one scenario – run with multiple set of test data
3. Background – repeated given
4. Datatable – tabular data

Automation – Web application – selenium

1. Add .dll file for Selenium WebDriver and Selenium Support from nuget package manager
2. To launch chrome browser – add the chromedriver.
3. Inspect – tagname, attribute, text or not
4. Locators
   1. Id
   2. Name
   3. Class name
   4. Tagname
   5. Link text
   6. Partial text
   7. Xpath
   8. CSS

When there are duplicate locator, then findElement method will pick the first one.

1. driver.FindElement(By.Name("username")) 🡪 it will take only 0.5s to check the presence of element
2. Synchronization
   1. Unconditional wait (Thread.Sleep(5000)
   2. Conditional wait
      1. Implicit wait
         1. Default implicit wait = 0s
         2. Applicable for all findElement methods
         3. Example: Implicit wait =20s
            1. If element not present, it will check for 20s and then throw exception
            2. If element present, immediately it will do the operation.
            3. Polling time – 0.5s

Git – SCM

Git is a [free and open source](https://git-scm.com/about/free-and-open-source) distributed version control system

Architecture:-

Project (local machine) 🡪 local repository (local machine) 🡪 remote repository (github, gitlab, bitbucket, aws code commit)

#1. Navigate to the url

#2. Enter Admin username

#3. Enter password

#4. Click Login

#5. click on PIM

#6. Click on Add Employee

#7. Enter firstname

#8. Enter middlename

#9. Enter lastname

#10. Enter Employee Id

#11. Check - create credential

#12.

#13. Enter Username

#14. Enter password

#15. Enter Confirm password

#16. Select status as Disabled

#17. Click on save

Scenario: Add Valid Employee

Given I have browser with orangehrm page

When I enter username as 'Admin'

And I enter password as 'admin123'

And I click on login

And I click on PIM

And I click on Add Employee

And I fill the add employee section

| *firstname* | *middlename* | *lastname* | *employee\_id* | *toggle\_login\_detail* | *username* | *password* | *confirm\_password* | *status* |

| Saul | J | Goodman | 1001 | on | jimmy | Welcome@123 | Welcome@123 | disabled |

And I click on save employee

Then I should be navigated to personal details section with added employee records

Scenario: Add Valid Employee2

Given I have browser with orangehrm page

When I enter username as 'Admin'

And I enter password as 'admin123'

And I click on login

And I click on PIM

And I click on Add Employee

And I fill the add employee section

| *firstname* | *middlename* | *lastname* | *employee\_id* | *toggle\_login\_detail* | *username* | *password* | *confirm\_password* | *status* |

| Peter | J | Goodman | 1002 | on | jimmy | Welcome@123 | Welcome@123 | disabled |

And I click on save employee

Then I should be navigated to personal details section with added employee records

Scenario: Add Emergency Contact

Given I have browser with orangehrm page

When I enter username as '<username>'

And I enter password as '<password>'

And I click on login

And I click on My Info

And I click on Emergency Contacts

And I click on add Assigned Emergency Contacts

And I fill the emergency contact details

| *name* | *relationship* | *home\_telephone* | *mobile* | *work\_telephone* |

| Jack | brother | 788 | 88 | 778 |

And I click on save emergency contact

Then I should see the added records in the table