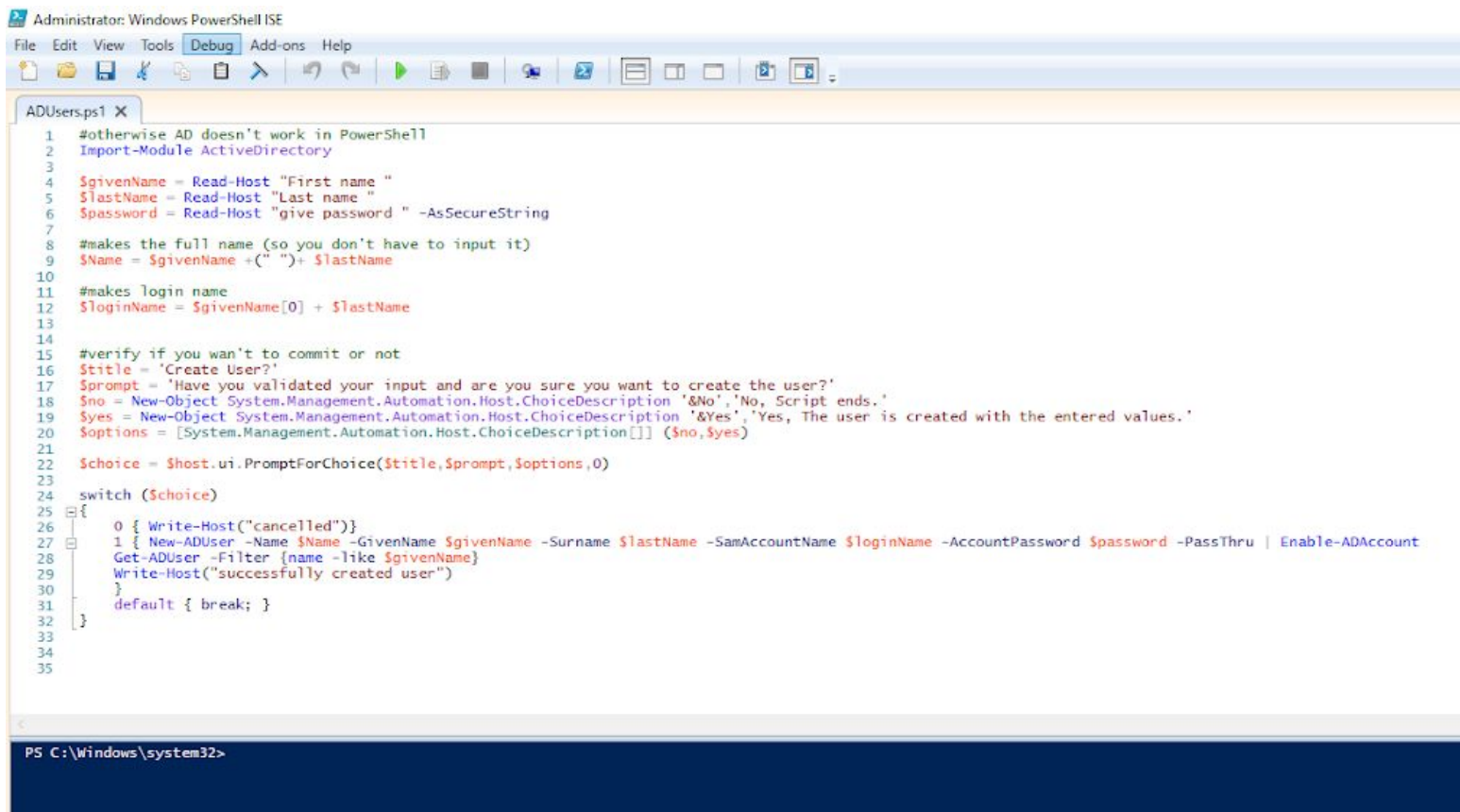


# Case Study

Zoals u misschien vernomen had in mijn dagverslag gaat mijn Case Study over een PowerShell Script.

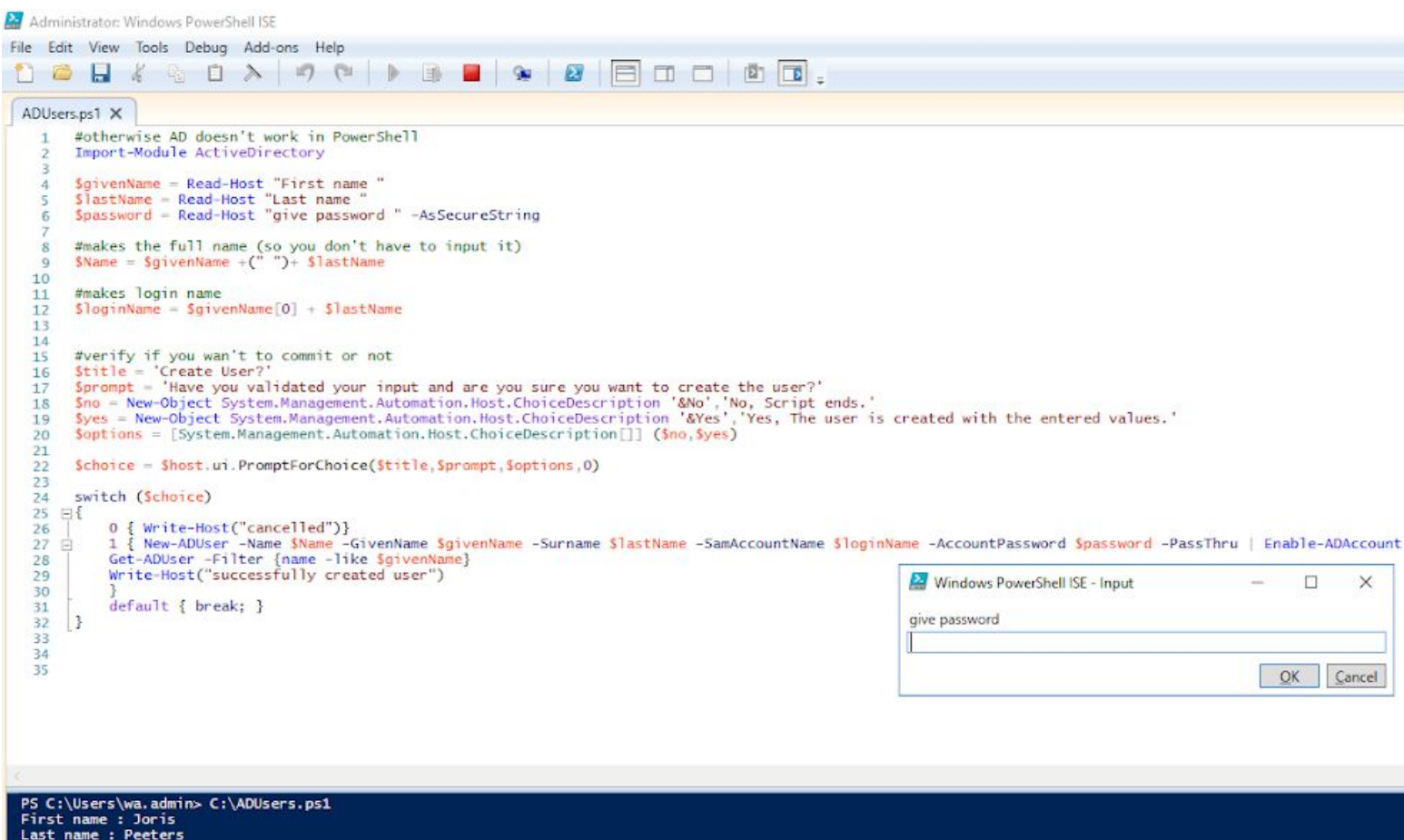
Mijn stagebegeleider Filip De Loore had me een collega aangewezen die mij zou helpen met mijn Case Study, die collega was Kristof Aerts. Hij heeft me een Case Study gegeven over PowerShell Scripts. Hij heeft eerst uitgelegd waarom hij dat gebruikt en hoe krachtig het is. Hij gebruikt PowerShell Scripts bijna voor alles wat hij meer dan 1 keer moet doen zoals bv. het toevoegen van AD Users door middel van een lijst. Dat is ook wat ik als Case Study heb gekregen, namelijk het toevoegen van AD Users op een Domain Controller. Kristof heeft mij eerst ook nog een paar tips gegeven en laten zien wat ik allemaal kon gebruiken. hij heeft mij ook heel veel voorbeelden laten zien. Ik kreeg van Kristof namelijk ook een remote desktop die ik mocht gebruiken om te testen zodat als ik iets fout had, dit geen effect had op belangrijke servers.

The image shows a screenshot of the Windows PowerShell ISE (Integrated Scripting Environment) running as an administrator. The window title is "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar contains various icons for file operations and execution. The script file is named "ADUsers.ps1". The script code is as follows:

```
1 #otherwise AD doesn't work in PowerShell
2 Import-Module ActiveDirectory
3
4 $givenName = Read-Host "First name "
5 $lastName = Read-Host "Last name "
6 $password = Read-Host "give password " -AsSecureString
7
8 #makes the full name (so you don't have to input it)
9 $Name = $givenName + (" ") + $lastName
10
11 #makes login name
12 $loginName = $givenName[0] + $lastName
13
14
15 #verify if you want to commit or not
16 $title = 'Create User?'
17 $prompt = 'Have you validated your input and are you sure you want to create the user?'
18 $no = New-Object System.Management.Automation.Host.ChoiceDescription '&No','No, Script ends.'
19 $yes = New-Object System.Management.Automation.Host.ChoiceDescription '&Yes','Yes, The user is created with the entered values.'
20 $options = [System.Management.Automation.Host.ChoiceDescription[]] ($no,$yes)
21
22 $choice = $Host.ui.PromptForChoice($title,$prompt,$options,0)
23
24 switch ($choice)
25 {
26 0 { Write-Host("cancelled")}
27 1 { New-ADUser -Name $Name -GivenName $givenName -Surname $lastName -SamAccountName $loginName -AccountPassword $password -PassThru | Enable-ADAccount
28     Get-ADUser -Filter {name -like $givenName}
29     Write-Host("successfully created user")
30 }
31 default { break; }
32 }
33
34
35
```

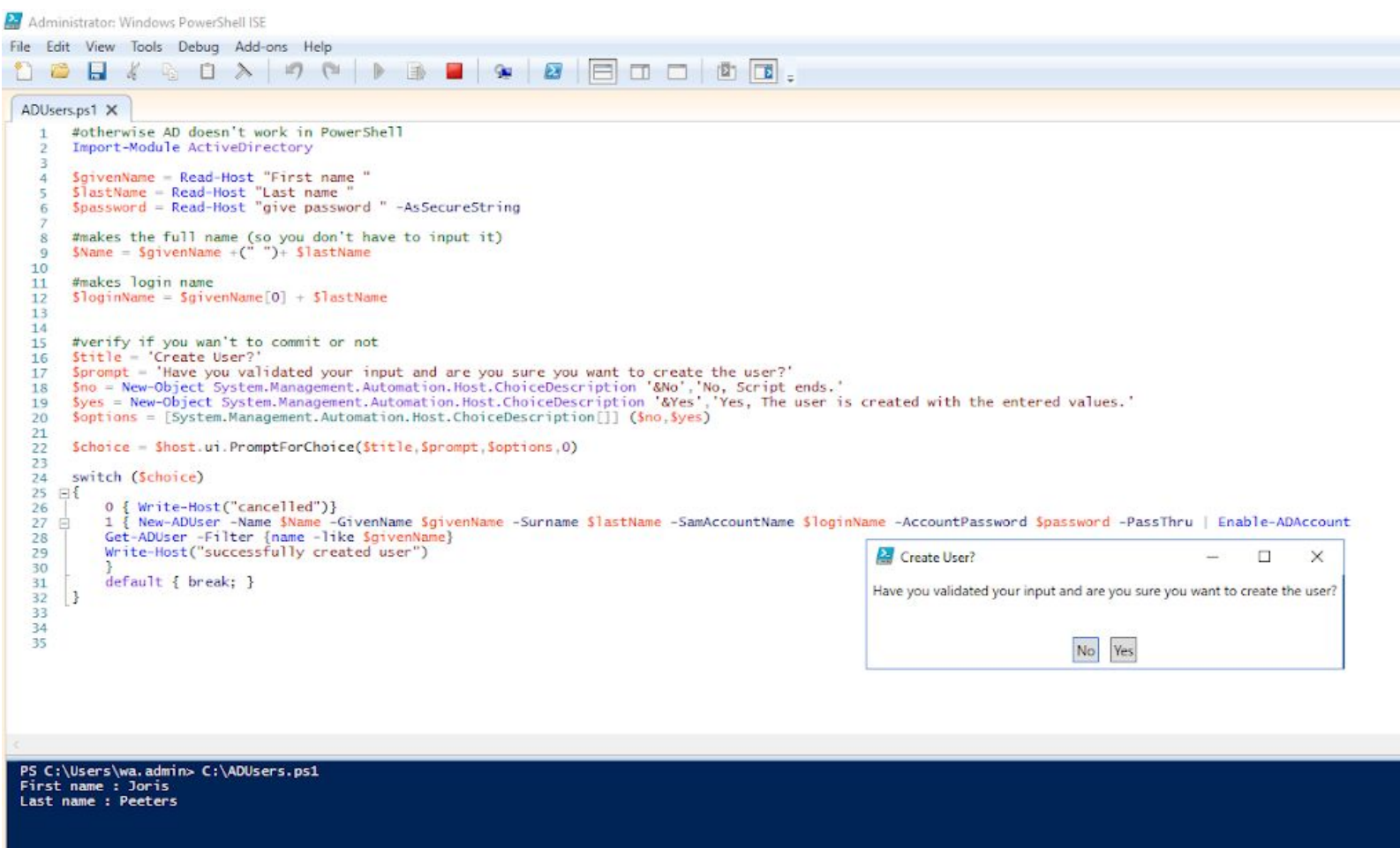
The bottom of the window shows a command prompt with the path "PS C:\Windows\system32>".

Hier ziet u het scriptje dat ik heb geschreven. U kan het ook vinden op mijn GitHub: <https://github.com/MatsB-immalle/ADUser> . Het doel van dit script is om automatisch een AD User aan te maken waarbij je enkel de voornaam, achternaam en wachtwoord moet invoeren en de rest wordt automatisch voor u ingestelds.



```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
ADUsers.ps1 X
1 #otherwise AD doesn't work in PowerShell
2 Import-Module ActiveDirectory
3
4 $givenName = Read-Host "First name "
5 $lastName = Read-Host "Last name "
6 $password = Read-Host "give password " -AsSecureString
7
8 #makes the full name (so you don't have to input it)
9 $Name = $givenName + (" ") + $lastName
10
11 #makes login name
12 $loginName = $givenName[0] + $lastName
13
14
15 #verify if you want to commit or not
16 $title = 'Create User?'
17 $prompt = 'Have you validated your input and are you sure you want to create the user?'
18 $no = New-Object System.Management.Automation.Host.ChoiceDescription '&No;', 'No, Script ends.'
19 $yes = New-Object System.Management.Automation.Host.ChoiceDescription '&Yes;', 'Yes, The user is created with the entered values.'
20 $options = [System.Management.Automation.Host.ChoiceDescription[]] ($no, $yes)
21
22 $choice = $host.ui.PromptForChoice($title, $prompt, $options, 0)
23
24 switch ($choice)
25 {
26 0 { Write-Host("cancelled")}
27 1 { New-ADUser -Name $Name -GivenName $givenName -Surname $lastName -SamAccountName $loginName -AccountPassword $password -PassThru | Enable-ADAccount
28     Get-ADUser -Filter {name -like $givenName}
29     Write-Host("successfully created user")
30 }
31 default { break; }
32 }
33
34
35
PS C:\Users\wa.admin> C:\ADUsers.ps1
First name : Joris
Last name : Peeters
```

Hier zit u het script in werking, er wordt een voornaam en achternaam ingevoerd en voor het wachtwoord komt er een pop-up die het wachtwoord opgevraagd en als je dit invult komen er sterretjes zodat het wachtwoord niet meer herkenbaar is. Dit is makkelijk te doen door de -AsSecureString command.



The screenshot shows the Windows PowerShell ISE interface. The script file is named 'ADUsers.ps1'. The script code is as follows:

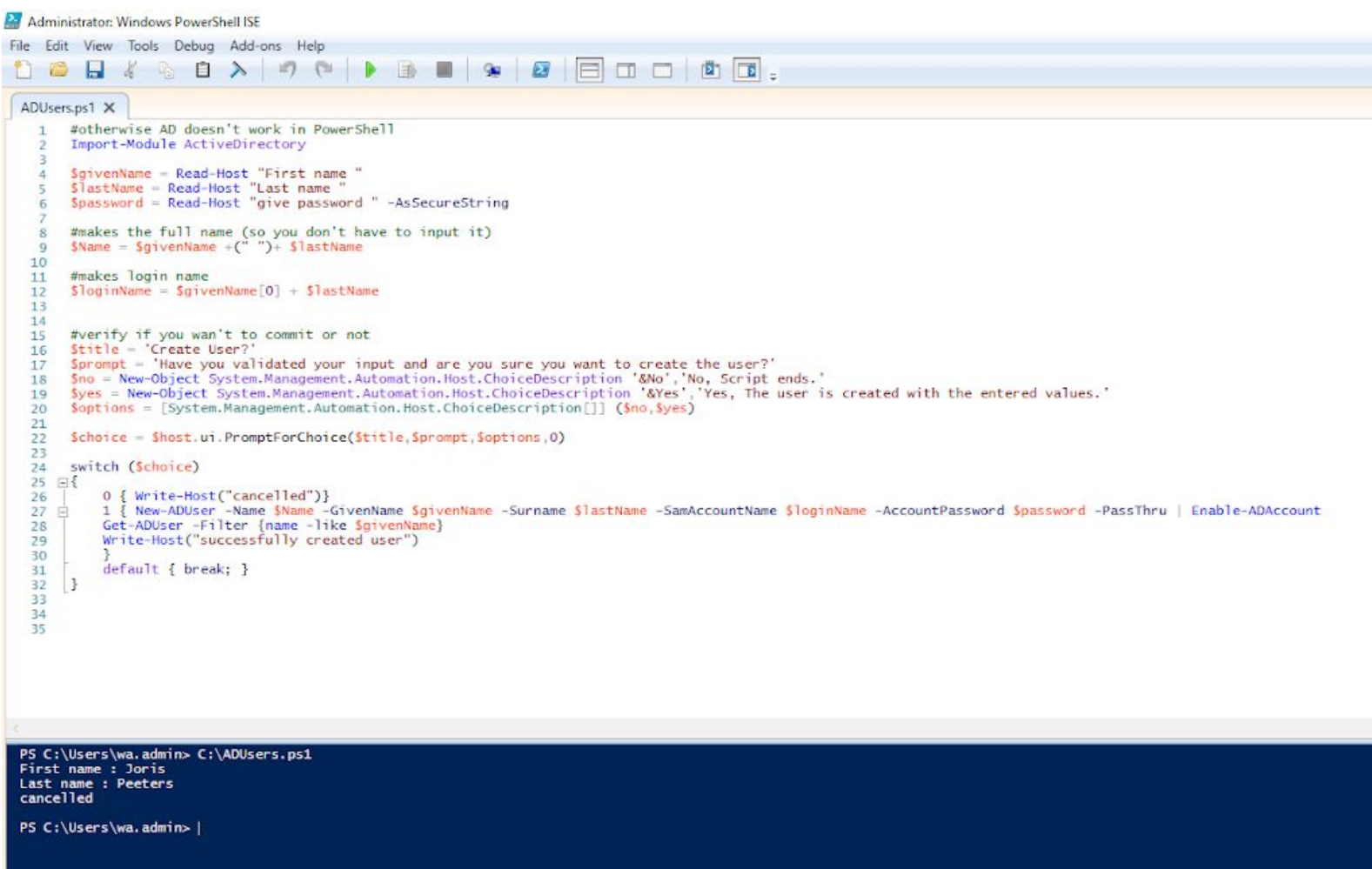
```
1 #otherwise AD doesn't work in PowerShell
2 Import-Module ActiveDirectory
3
4 $givenName = Read-Host "First name "
5 $lastName = Read-Host "Last name "
6 $password = Read-Host "give password " -AsSecureString
7
8 #makes the full name (so you don't have to input it)
9 $Name = $givenName + (" ") + $lastName
10
11 #makes login name
12 $loginName = $givenName[0] + $lastName
13
14
15 #verify if you want to commit or not
16 $title = 'Create User?'
17 $prompt = 'Have you validated your input and are you sure you want to create the user?'
18 $no = New-Object System.Management.Automation.Host.ChoiceDescription '&No', 'No, Script ends.'
19 $yes = New-Object System.Management.Automation.Host.ChoiceDescription '&Yes', 'Yes, The user is created with the entered values.'
20 $options = [System.Management.Automation.Host.ChoiceDescription[]] ($no,$yes)
21
22 $choice = $host.ui.PromptForChoice($title,$prompt,$options,0)
23
24 switch ($choice)
25 {
26 0 { Write-Host("cancelled")}
27 1 { New-ADUser -Name $Name -GivenName $givenName -Surname $lastName -SamAccountName $loginName -AccountPassword $password -PassThru | Enable-ADAccount
28     Get-ADUser -Filter {name -like $givenName}
29     Write-Host("successfully created user")
30 }
31 default { break; }
32 }
33
34
35
```

A dialog box titled 'Create User?' is displayed, asking 'Have you validated your input and are you sure you want to create the user?'. It has 'No' and 'Yes' buttons.

The command prompt at the bottom shows the execution of the script:

```
PS C:\Users\wa.admin> C:\ADUsers.ps1
First name : Joris
Last name : Peeters
```

Vervolgens springt er een venster open dat vraagt of u zeker door wilt gaan. Dit is handig voor als je een fout hebt getypt en je wilt de user niet aanmaken.



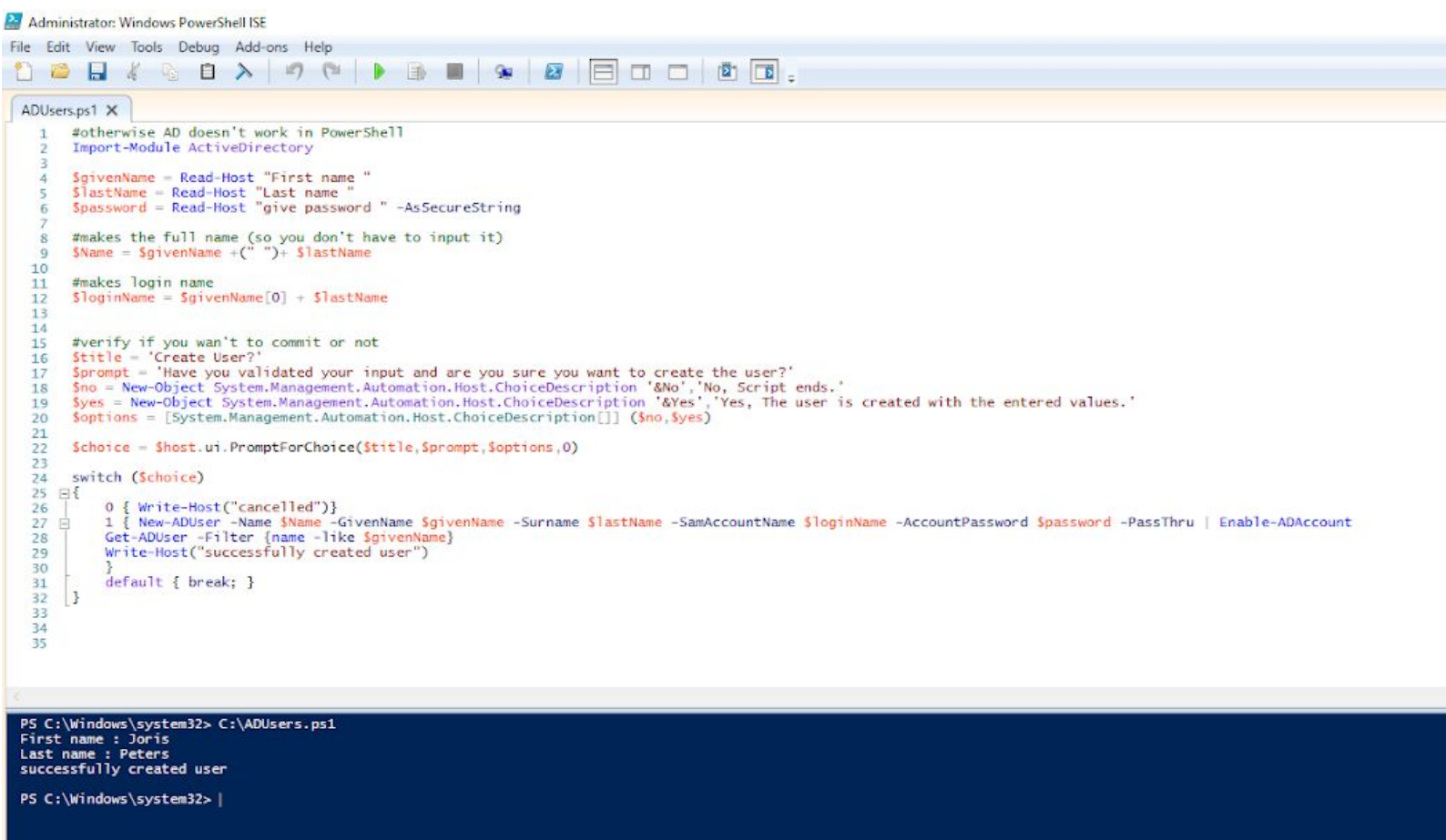
```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help

ADUsers.ps1 X
1 #otherwise AD doesn't work in PowerShell
2 Import-Module ActiveDirectory
3
4 $givenName = Read-Host "First name "
5 $lastName = Read-Host "Last name "
6 $password = Read-Host "give password " -AsSecureString
7
8 #makes the full name (so you don't have to input it)
9 $Name = $givenName +(" ") + $lastName
10
11 #makes login name
12 $loginName = $givenName[0] + $lastName
13
14
15 #verify if you wan't to commit or not
16 $title = 'Create User?'
17 $prompt = 'Have you validated your input and are you sure you want to create the user?'
18 $no = New-Object System.Management.Automation.Host.ChoiceDescription '&No','No, Script ends.'
19 $yes = New-Object System.Management.Automation.Host.ChoiceDescription '&Yes','Yes, The user is created with the entered values.'
20 $options = [System.Management.Automation.Host.ChoiceDescription[]] ($no,$yes)
21
22 $choice = $host.ui.PromptForChoice($title,$prompt,$options,0)
23
24 switch ($choice)
25 {
26 0 { Write-Host("cancelled")}
27 1 { New-ADUser -Name $Name -GivenName $givenName -Surname $lastName -SamAccountName $loginName -AccountPassword $password -PassThru | Enable-ADAccount
28 Get-ADUser -Filter {name -like $givenName}
29 Write-Host("successfully created user")
30 }
31 default { break; }
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

PS C:\Users\wa.admin> C:\ADUsers.ps1
First name : Joris
Last name : Peeters
cancelled

PS C:\Users\wa.admin> |
```

Als je bij het vorig venster op No klikt gebeurt er dit en wordt het gecancelled.



The screenshot shows the Windows PowerShell ISE interface. The title bar reads "Administrator: Windows PowerShell ISE". The menu bar includes "File", "Edit", "View", "Tools", "Debug", "Add-ons", and "Help". The toolbar contains icons for file operations and execution. The script editor shows a file named "ADUsers.ps1" with the following code:

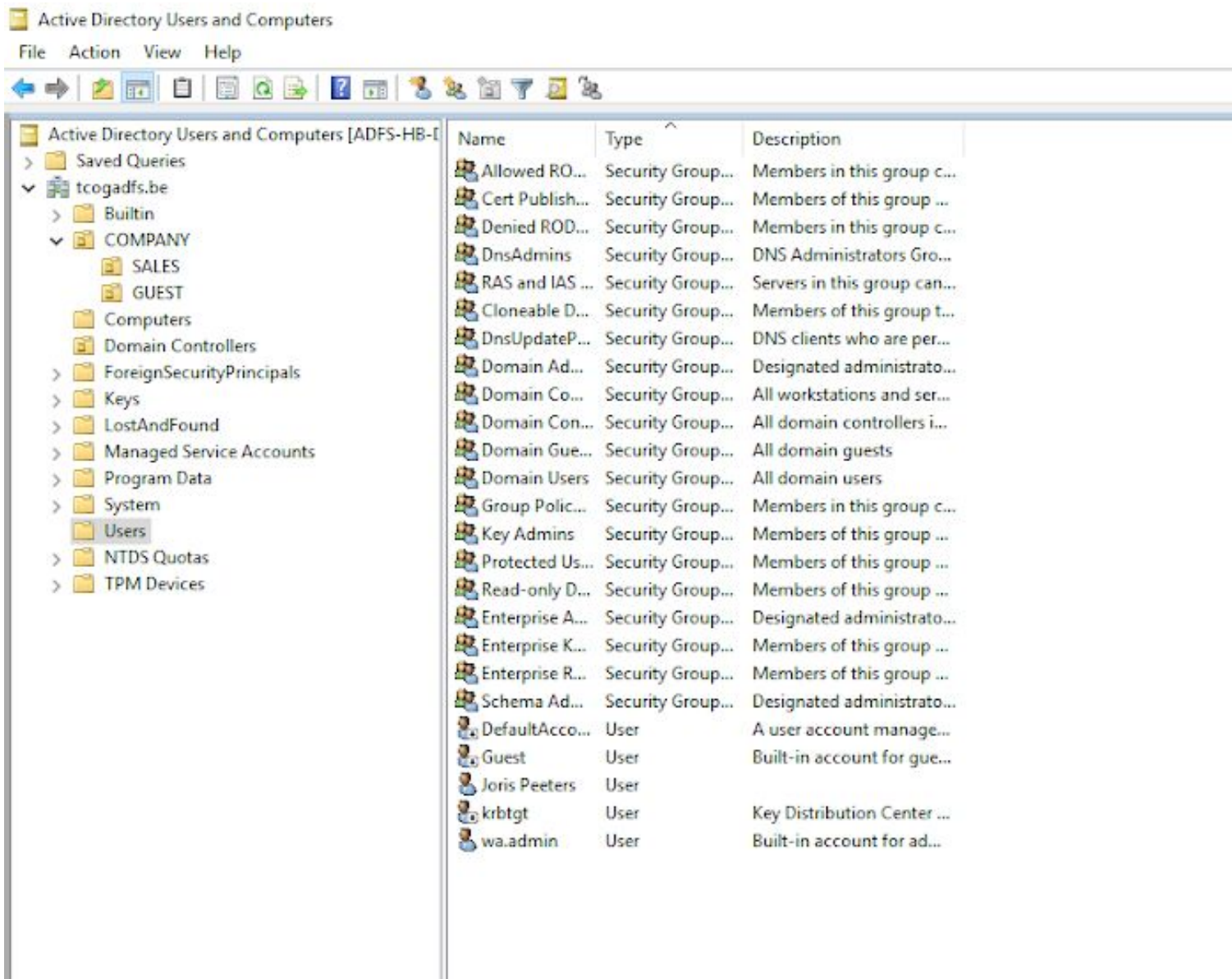
```
1 #otherwise AD doesn't work in PowerShell
2 Import-Module ActiveDirectory
3
4 $givenName = Read-Host "First name "
5 $lastName = Read-Host "Last name "
6 $password = Read-Host "give password " -AsSecureString
7
8 #makes the full name (so you don't have to input it)
9 $Name = $givenName +(" ") + $lastName
10
11 #makes login name
12 $loginName = $givenName[0] + $lastName
13
14
15 #verify if you want't to commit or not
16 $title = 'Create User?'
17 $prompt = 'Have you validated your input and are you sure you want to create the user?'
18 $no = New-Object System.Management.Automation.Host.ChoiceDescription '&No','No, Script ends.'
19 $yes = New-Object System.Management.Automation.Host.ChoiceDescription '&Yes','Yes, The user is created with the entered values.'
20 $options = [System.Management.Automation.Host.ChoiceDescription[]] ($no,$yes)
21
22 $choice = $host.ui.PromptForChoice($title,$prompt,$options,0)
23
24 switch ($choice)
25 {
26     0 { Write-Host("cancelled")}
27     1 { New-ADUser -Name $Name -GivenName $givenName -Surname $lastName -SamAccountName $loginName -AccountPassword $password -PassThru | Enable-ADAccount
28         Get-ADUser -Filter {name -like $givenName}
29         Write-Host("successfully created user")
30     }
31     default { break; }
32 }
33
34
35
```

The console window at the bottom shows the execution of the script:

```
PS C:\Windows\system32> C:\ADUsers.ps1
First name : Joris
Last name : Peters
successfully created user
PS C:\Windows\system32> |
```

Als je op Yes klikt wordt de user aangemaakt en wordt er gezegd of het gelukt is.





Zoals u hier ziet is het aanmaken van de user wel degelijk gelukt.

Zo dit is het script dat ik gemaakt heb. Dit script is natuurlijk nog een simpele versie, maar je kan hier nog veel kanten mee uit bv. ervoor zorgen dat alle gegevens worden uitgelezen uit een csv file of er kunne meerdere gegevens ingevuld worden en je kan het ook Remote draaien op een desktop. Je kan er dus nog heel veel mee doen maar met de tijd die ik had ben ik tot hier geraakt.

In de tijd dat ik hier mee bezig ben geweest heb ik toch veel bijgeleerd. Vooral voor wat script gebruikt worden in de praktijk en dat het echt handig kan zijn. Ook tijdens het schrijven van het script heb ik veel functies van PowerShell leren kennen. Dit was dus een zeer interessante en leerrijke taak.