Project Requirements Document

1. Introduction

The system is a podcast's feed platform . The main purpose of this system is to allow users to explore and listen to podcasts from a wide range of subjects and to provide digital-dust based podcasts suggestions for the users to discover.

1.1 Purpose of the system

As discussed before, the purpose of the system is to give the users the ability to listen to podcasts and explore new podcasts and shows based on their interests. The users will be able to play and rate the different podcasts on the platform and also will be allowed to maintain an actual user through the log-in system of the platform so the platform will be able to suggest podcasts based on their preferences.

1.2 Scope of the system

The scope of the system is to be responsible for the registration of users, managing the different podcasts (upvote, downvote and play) and searching a specific podacts's subject and its related subjects.

The system is not taking part in the content making of the items appearing on the platform. the platform will be free to use and will not obligate the users to pay for using it.

2. Actors and goals

User:

Type of actor: Primary.

Description: Every person that enrolled on the platform.

Goals: Registration, play podcasts , browse podcasts,rate podcasts and publish podcasts.

System’s Admin:

Type of actor: Primary.

Description: An enrolled user with ‘admin’ permissions

Goals: Watch all the enrolled users ,delete all the enrolled users , delete all podcasts ,and all user’s goals

System’s Manager:

Type of actor: Primary.

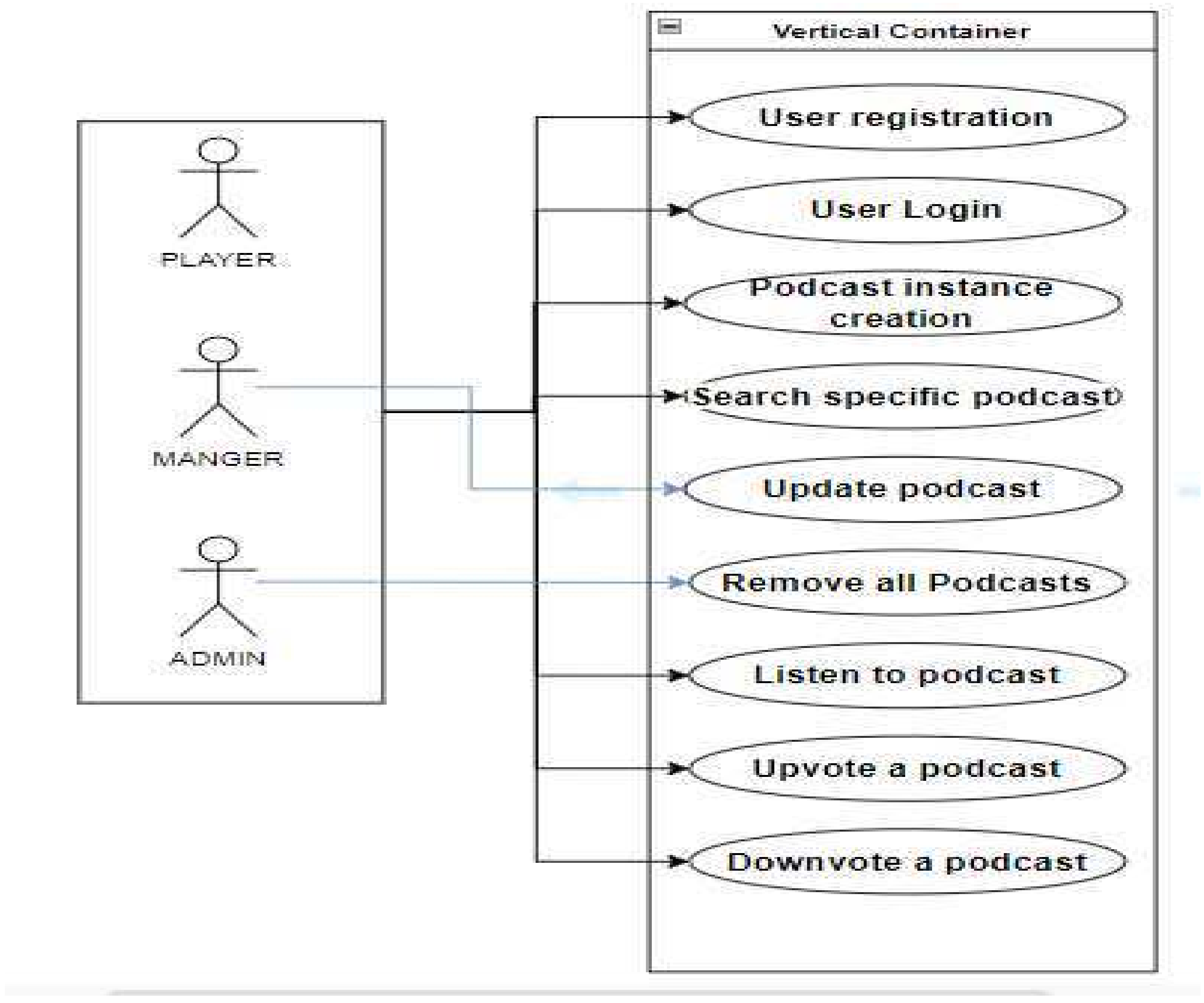
Description: An enrolled user with ‘manager’ permissions

Goals: Update existing podcast details and all user’s goals

3. Functional Requirements

1. Users will be required to log in to the system before using it.
2. The system will allow the search of a podcast’s subjects for every actor.
3. A user can see the previous podcasts he was listening to.
4. A user can manage a list of podcasts for future listening.
5. A user can rate the different podcasts.
6. A user can play a podcast.
7. A user can rate a podcast.
8. A user can publish a podcast.
9. An admin can watch all the enrolled users.
10. An admin can delete all the enrolled users.
11. An admin can delete all podcasts.
12. A manager can update existing podcast details.

3.1 Use case diagram



3.2 Use cases

Use Case: User registration

Goal: Registering to the platform

Participating actors: User (PLAYER,MANAGER,ADMIN)

Main flow:

1. User fills registration form.
2. System verifies that there is no existing user with the same ID
3. System notifies the user of successful registration

Alternate flow:

1a. The user didn’t fill all the form’s fields
1a1. System notifies the user with the proper notification.

Alternate flow:

2a. The user already registered
2a1. System notifies the user with the proper notification

Use Case: User Login to the system

Goal: Login into the platform

Participating actors: User (PLAYER,MANAGER,ADMIN)

Main flow:

1. User asks to login to the platform.
2. System verifies that there is an existing user with the specified ID.
3. System notifies the user of successful login.

Alternate flow:

1a. The user didn’t provide the correct login details.
1a1. System notifies the user with the proper notification.

Use Case: Podcast instance creation

Goal: Post podcast onto the platform

Participating actors: User (PLAYER,MANAGER,ADMIN)

Main flow:

1. User asks to publish podcast instance onto the platform
2. System provides the wished podcast.

Alternate flow:
2a. The user didn't fill all the form's fields (aka. name,author,genre)
2a1. System notifies the user with the proper notification.

Alternate flow:
2a. The user provided the wrong user's details
2a1. System notifies the user with the proper notification.

Use Case: Search specific podcast

Goal: the user will navigate successfully to the wished podcast

Participating actors: User (PLAYER,MANAGER,ADMIN)

Main flow:

1. User asks for a specific podcast according to the provided searching field
2. System provides the wished podcast.

Alternate flow:

2a. The wished podcast does not exist on the platform.
2a1. System notifies the user with the proper notification.

Alternate flow:

2a. The user didn't provide a searching method .
2a1. System notifies the user with the proper notification.

Use case: Update podcast

Goal: Updating any detail of a podcast

Participating actors: User (MANAGER)

Main flow:

1. Manager logs in to the system
2. System verifies Manager has an existing user
3. Manager provides the updated podcast instance
4. System verifies that details are valid
5. System updates the podcast in any relevant instance

Alternate flow:
1a. The user didn't provide the correct login details.

1a1. System notifies the user with the proper notification.

Alternate flow:

2a. User is non ‘Manager’ type

2a1. The system notifies the user that only Manager can update podcast instances

Alternate flow:

3a. User provides updated podcast instance with missing essential details

3a1. The system notifies the user with the proper notification.

Use case: **Remove all Podcasts**

Goal: **Remove all podcasts instances from the platform**

Participating actors: **User (ADMIN)**

Main flow:

1. Admin logs in to the system
2. System verifies Admin has an existing user
3. Admin asks the system to remove all podcast instances
4. System removes the podcast from the course list
5. System notifies of successful removal

Alternate flow:

1a. The user didn’t provide the correct login details.

1a1. System notifies the user with the proper notification.

Alternate flow:

2a. User is non ‘Admin’ type

2a1. The system notifies the user that only Admin can remove podcast instances

---

Use Case: **Listen to podcast**

Goal: **Let the user listen to his wished podcast**

Participating actors: **User (PLAYER,MANAGER,ADMIN)**

Main flow:

1. User logs in to the system
2. System verifies User has an existing user
3. User asks to listen to specific podcast
4. System checks if the podcast exists
5. System update the listener’s counter
6. System provides the requested podcast

Alternate flow:
2a. The user didn’t provide the correct login details.
2a1. System notifies the user with the proper notification.

Alternate flow:
3a. The user didn’t specify the proper command.
3a1. System notifies the user that the command is not properly defined.

Alternate flow:
4a. The Podcast does not exist:
4a1. System notifies that the podcast does not exist.

Use Case: **Upvote a podcast**

Goal: **Upvoting the podcast’s rating**

Participating actors: **User (PLAYER,MANAGER,ADMIN)**

Main flow:
1. User logs in to the system
2. System verifies User has an existing user
3. User asks to upvote the rating to specific podcast
4. System checks if the podcast exists
5. System update the podcast’s upvotes counter
6. System provides the updated podcast

Alternate flow:
2a. The user didn’t provide the correct login details.
2a1. System notifies the user with the proper notification.

Alternate flow:
3a. The user didn’t specify the proper command.
3a1. System notifies the user that the command is not properly defined.

Alternate flow:
4a. The podcast does not exist:
4a1. System notifies that the podcast does not exist.

Use Case: **Downvote a podcast**

Goal: **Downvoting the podcast’s rating**

Participating actors: **User (PLAYER,MANAGER,ADMIN)**

Main flow:
1. User logs in to the system
2. System verifies User has an existing user
3. User asks to downvote the rating to specific podcast

4. System checks if the podcast exists
5. System update the podcast’s downvotes counter
6. System provides the updated podcast

Alternate flow:

2a. The user didn’t provide the correct login details.
2a1. System notifies the user with the proper notification.

Alternate flow:

3a. The user didn’t specify the proper command.
3a1. System notifies the user that the command is not properly defined.

Alternate flow:

4a. The podcast does not exist:
4a1. System notifies that the podcast does not exist.

4. Non - Functional Requirements

Requirement type	**Requirement Description**	**Requirement Number**
Usability - U	The system should be easy to learn.	1
Performance - P	The system should allow multiple users to listen and vote podcast up/down at the same time.	2
Supportability - S	System should run on any Browser.	3

Technologies

Lombok -

What is it?

Java library tool that is used to minimize or remove the boilerplate code.
Lombok generates boilerplate codes without presenting lines of code, but instead of creating boilerplate codes inside our source code, Lombok adds all these boilerplate codes at the compile-time in the ".class" file.

Why did we decide to work with it?

Lombok saves time for the developers. In addition to it, it also increases the readability of the source code and saves space.

Where can we find it in our project?

We can find Lombok usage in our boundaries.

Gradle -

What is it?

Build automation tool that is used to automate the creation of applications. The building process includes compiling, linking, and packaging the code. Gradle provides building, testing, and deploying software on several platforms.
The tool is popular for building any software and large projects.

Why did we decide to work with it?

The process becomes more consistent with the help of building automation tools.
Also, it has better performance, is more flexible, and provides better dependencies management (than its competition - Maven).

Where can we find it in our project?

Project level helps us with adding\ updating dependencies.

React -

<u>What is it?</u>

An open-source JavaScript library that is used for building user interfaces. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

<u>Why did we decide to work with it?</u>

React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application. In addition, some of our team members are already experienced with React.

<u>Where can we find it in our project?</u>

Client code.

MSSQL -

<u>What is it?</u>

MSSQL is a suite of database software published by Microsoft . it includes a relational database engine, which stores data in tables, columns, and rows, Integration Services (SSIS), which is a data movement tool for importing, exporting, and transforming data.

<u>Why did we decide to work with it?</u>

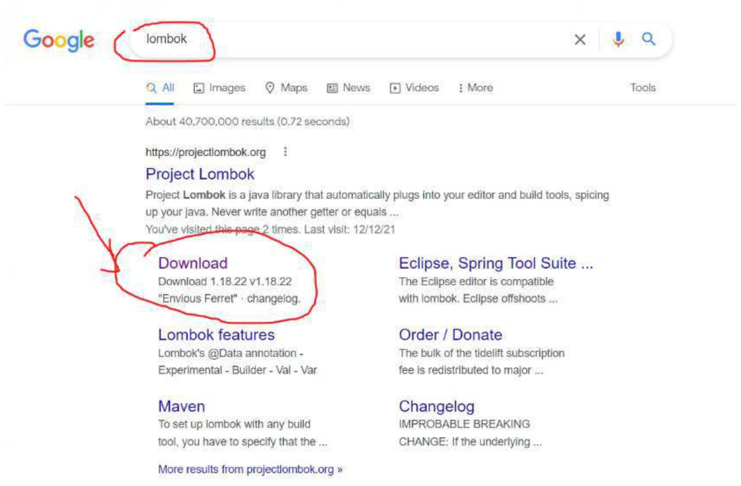
The Spring Framework provides extensive support for working with SQL databases, from direct JDBC access using JdbcTemplate to complete “object relational mapping” technologies such as Hibernate.
Therefore, it was just logical for us to go for relational SQL rather than NOSQL technologies.
Some of the team members feel much more confident engaging with SQL because all of us
Took the ‘Databases 101’ course during our second year at the college.

<u>Where can we find it in our project?</u>

The database tables.

Lombok installation guide

1. Search for Lombok in google and choose the Download link.



About 40,700,000 results (0.72 seconds)

https://projectlombok.org

Project Lombok

Project **Lombok** is a java library that automatically plugs into your editor and build tools, spicing up your java. Never write another getter or equals ...

You've visited this page 2 times. Last visit: 12/12/21

Download
Download 1.18.22 v1.18.22 "Envious Ferret" · changelog.

Eclipse, Spring Tool Suite ...
The Eclipse editor is compatible with lombok. Eclipse offshoots ...

Lombok features
Lombok's @Data annotation - Experimental - Builder - Val - Var

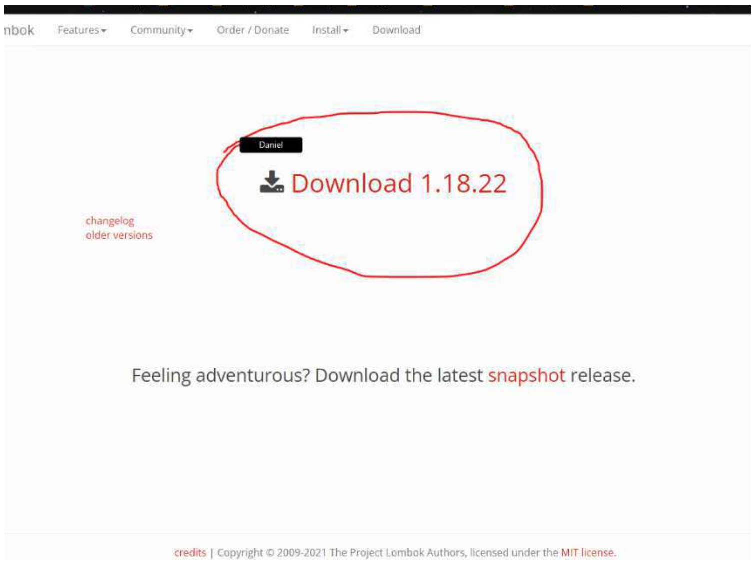
Order / Donate
The bulk of the tidelift subscription fee is redistributed to major ...

Maven
To set up lombok with any build tool, you have to specify that the ...

Changelog
IMPROBABLE BREAKING CHANGE: If the underlying ...

More results from projectlombok.org »

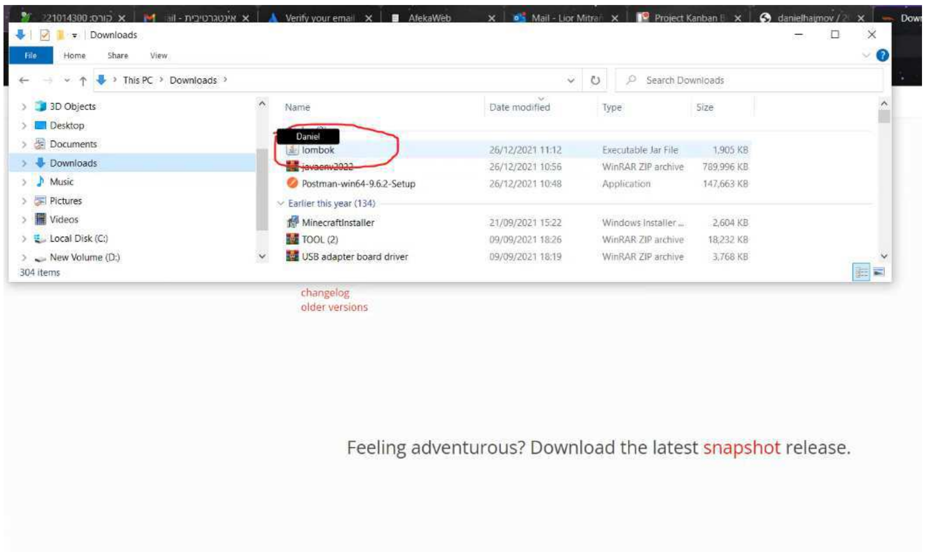
2. Download the latest major 1 version (1.+) of Lombok.



Feeling adventurous? Download the latest snapshot release.

credits | Copyright © 2009-2021 The Project Lombok Authors, licensed under the MIT license.

3. Go to the download path and open the Lombok.

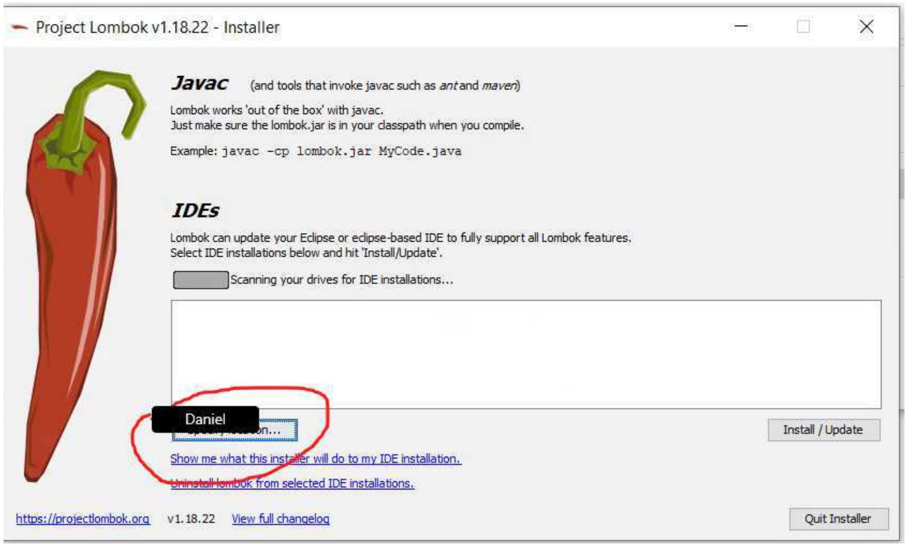


Name	Date modified	Type	Size
lombok	26/12/2021 11:12	Executable Jar File	1,905 KB
javaenv2022	26/12/2021 10:56	WinRAR ZIP archive	789,996 KB
Postman-win64-9.6.2-Setup	26/12/2021 10:48	Application	147,663 KB
Earlier this year (134)			
MinecraftInstaller	21/09/2021 15:22	Windows Installer ...	2,604 KB
TOOL (2)	09/09/2021 18:26	WinRAR ZIP archive	18,232 KB
USB adapter board driver	09/09/2021 18:19	WinRAR ZIP archive	3,768 KB

changelog
older versions

Feeling adventurous? Download the latest snapshot release.

4. Click "Specify location…" Button.



Project Lombok v1.18.22 - Installer

Javac (and tools that invoke javac such as *ant* and *maven*)

Lombok works 'out of the box' with javac.
Just make sure the lombok.jar is in your classpath when you compile.

Example: `javac -cp lombok.jar MyCode.java`

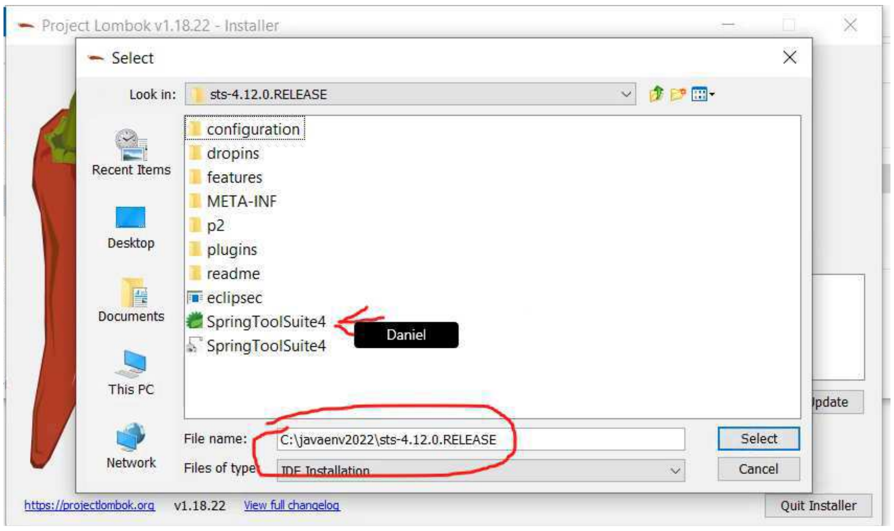
IDEs

Lombok can update your Eclipse or eclipse-based IDE to fully support all Lombok features.
Select IDE installations below and hit 'Install/Update'.

Scanning your drives for IDE installations…

Daniel
…on…
Install / Update
Show me what this installer will do to my IDE installation.
Uninstall lombok from selected IDE installations.
https://projectlombok.org
v1.18.22
View full changelog
Quit Installer

5. Go to the SpringToolSuite Directory (Usually C:\javaenv2022\sts-4.12.0.RELEASE). Select the SpringToolSuite.


Project Lombok v1.18.22 - Installer

Select

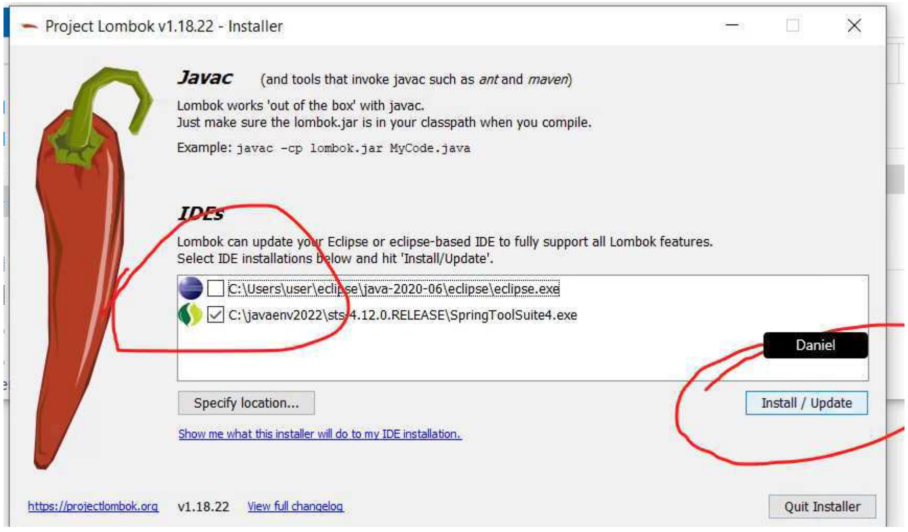
Look in: sts-4.12.0.RELEASE

configuration
dropins
features
META-INF
p2
plugins
readme
eclipsec
SpringToolSuite4
SpringToolSuite4

File name: C:\javaenv2022\sts-4.12.0.RELEASE

Files of type: IDE Installation

6. Remove all the unnecessary IDEs and click the “Install / Update” Button.


Project Lombok v1.18.22 - Installer

Javac (and tools that invoke javac such as *ant* and *maven*)

Lombok works 'out of the box' with javac.
Just make sure the lombok.jar is in your classpath when you compile.

Example: `javac -cp lombok.jar MyCode.java`

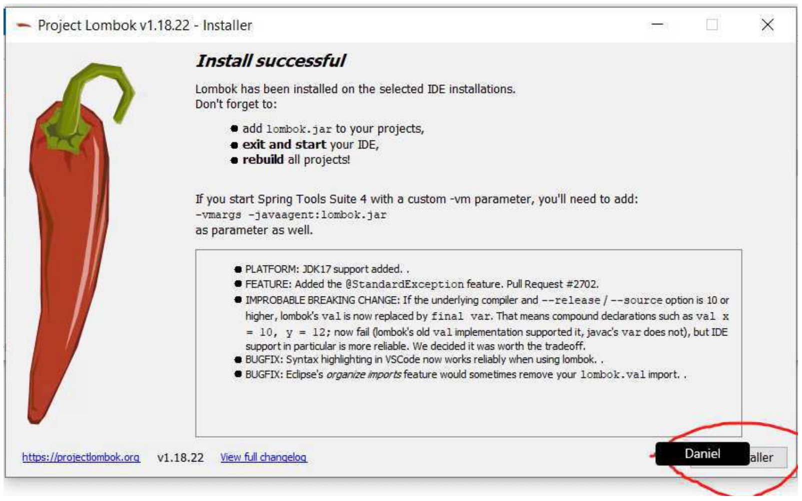
IDEs

Lombok can update your Eclipse or eclipse-based IDE to fully support all Lombok features.
Select IDE installations below and hit 'Install/Update'.

[ ] C:\Users\user\eclipse\java-2020-06\eclipse\eclipse.exe
[✓] C:\javaenv2022\sts-4.12.0.RELEASE\SpringToolSuite4.exe

Daniel
Specify location...
Install / Update
Show me what this installer will do to my IDE installation.
https://projectlombok.org
v1.18.22
View full changelog
Quit Installer

7. Quit the installer.



Project Lombok v1.18.22 - Installer

Install successful

Lombok has been installed on the selected IDE installations.
Don't forget to:

- add `lombok.jar` to your projects,
- **exit and start** your IDE,
- **rebuild** all projects!

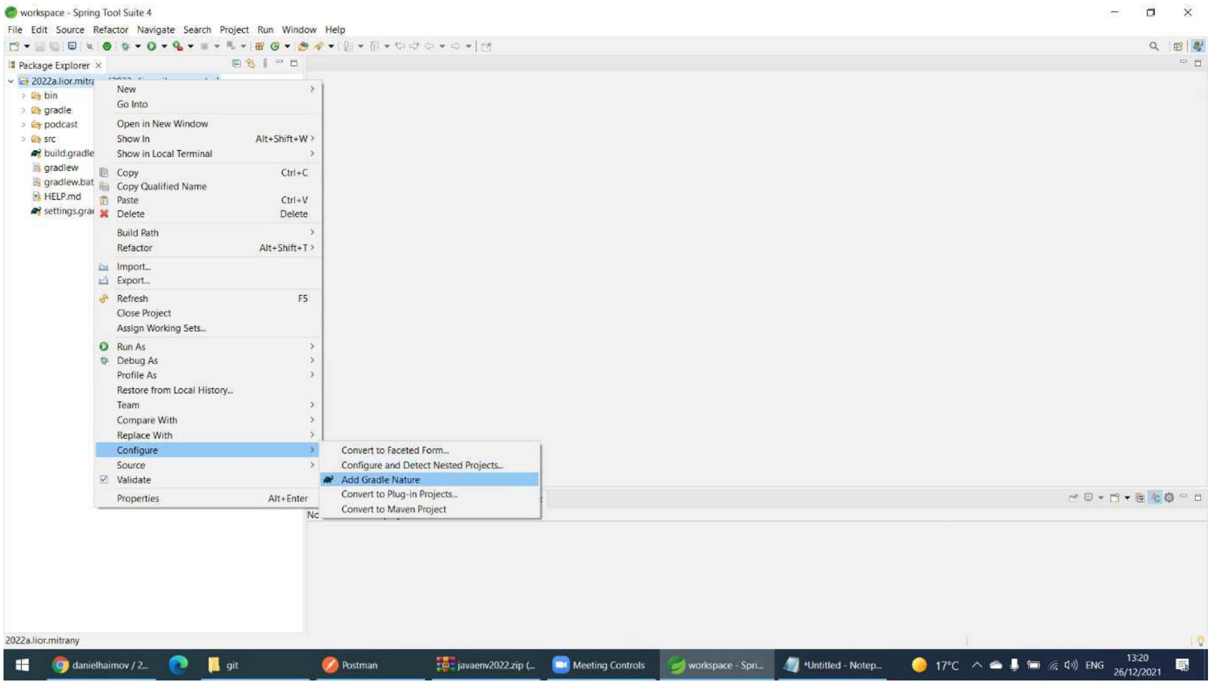
If you start Spring Tools Suite 4 with a custom -vm parameter, you'll need to add:
`-vmargs -javaagent:lombok.jar`
as parameter as well.

- PLATFORM: JDK17 support added. .
- FEATURE: Added the `@StandardException` feature. Pull Request #2702.
- IMPROBABLE BREAKING CHANGE: If the underlying compiler and `--release` / `--source` option is 10 or higher, lombok's `val` is now replaced by `final var`. That means compound declarations such as `val x = 10, y = 12;` now fail (lombok's old `val` implementation supported it, javac's `var` does not), but IDE support in particular is more reliable. We decided it was worth the tradeoff.
- BUGFIX: Syntax highlighting in VSCode now works reliably when using lombok. .
- BUGFIX: Eclipse's *organize imports* feature would sometimes remove your `lombok.val` import. .

https://projectlombok.org v1.18.22 View full changelog

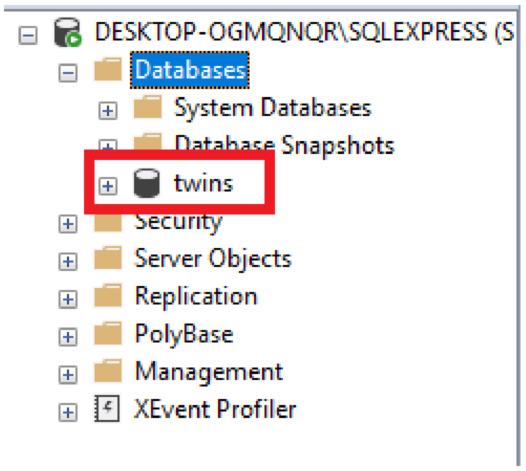
8. Now you can Clone the project (Notice: if the project is already cloned so rebuild the project).

9. Right click the project -> Configure -> Add Gradle Nature.

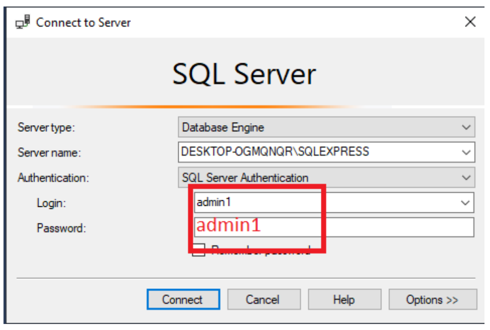


לאחר יצירת משתמש בשרת:

לאחר יצירת הDB בשרת:

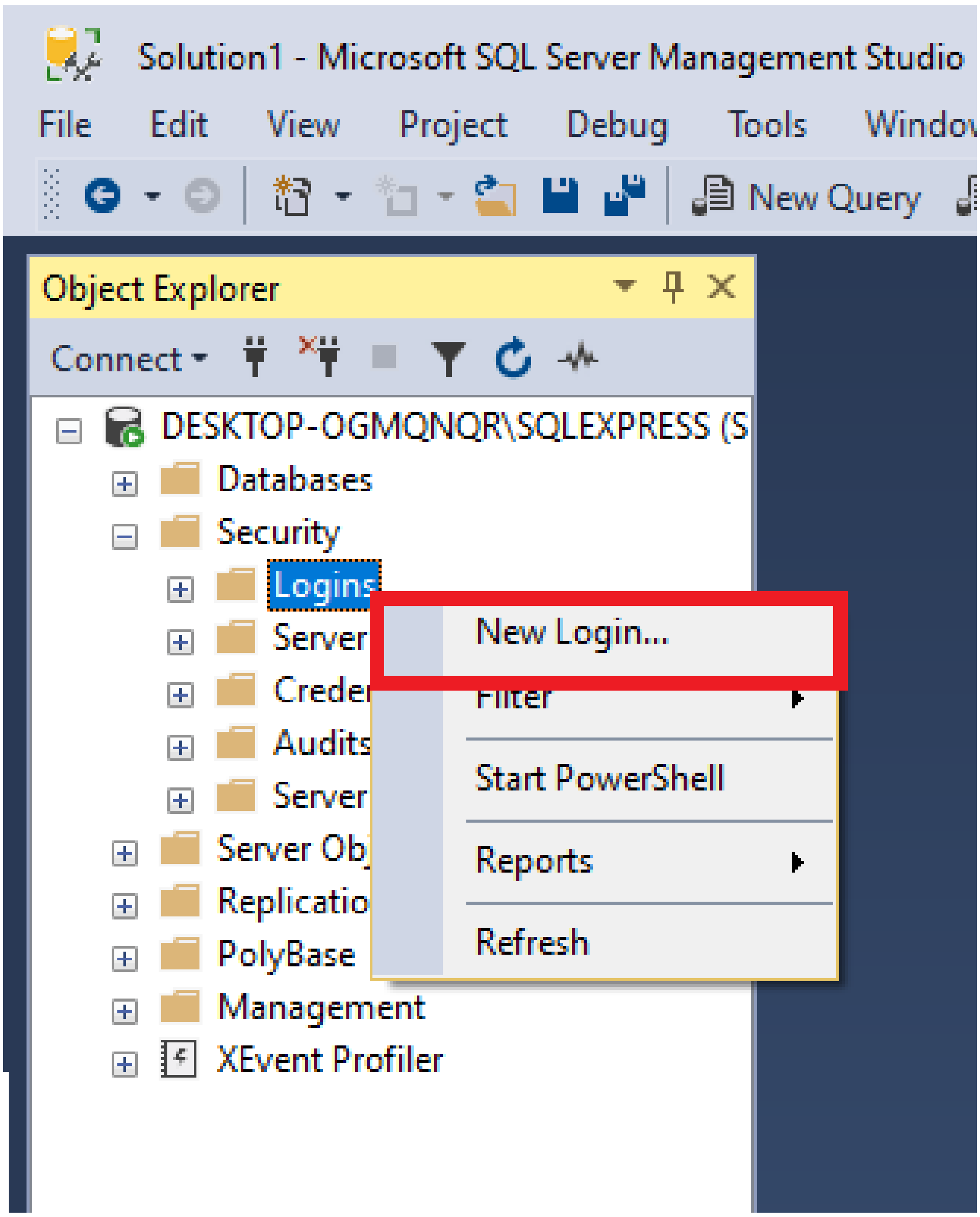


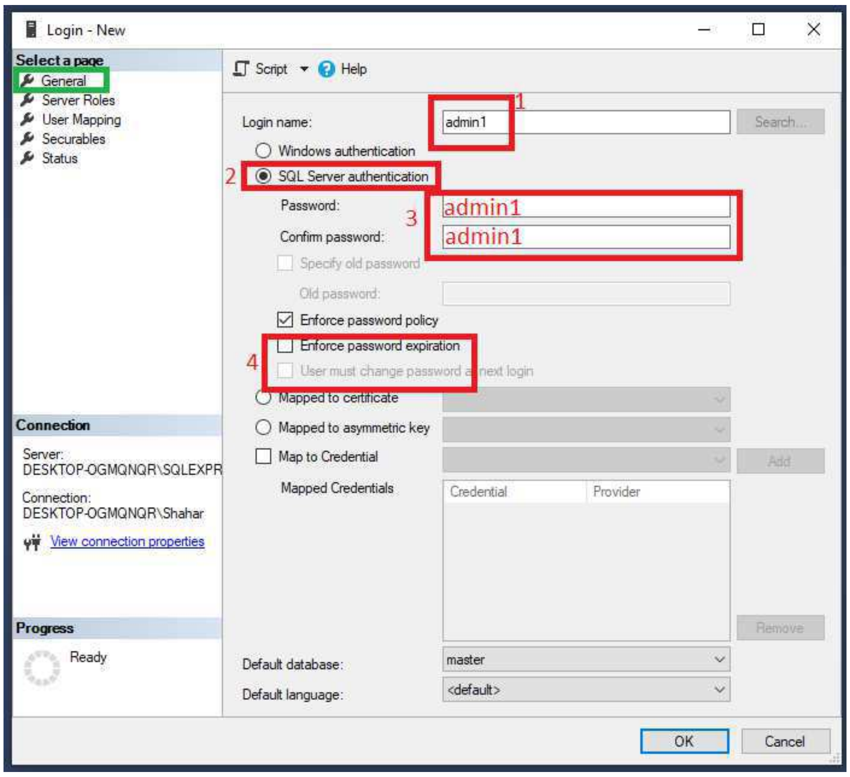
נרצה ליצור משתמש בשרת שאיתו נוכל להתחבר בצורה מרוחקת.



Server type: Database Engine
Server name: DESKTOP-OGMQNQR\SQLEXPRESS
Authentication: SQL Server Authentication
Login: admin1
Password: admin1

<u>יצירת המשתמש :</u>


Solution1 - Microsoft SQL Server Management Studio
File
Edit
View
Project
Debug
Tools
Windo
New Query
Object Explorer
Connect
DESKTOP-OGMQNQR\SQLEXPRESS (S
Databases
Security
Logins
Server
Creder
Audits
Server
Server Obj
Replicatio
PolyBase
Management
XEvent Profiler
New Login...
Filter
Start PowerShell
Reports
Refresh



Login name: admin1

Windows authentication

SQL Server authentication ●

Password: admin1

Confirm password: admin1

Specify old password ☐

Old password:

Enforce password policy ✓

Enforce password expiration ☐

User must change password at next login ☐

Mapped to certificate

Mapped to asymmetric key

Map to Credential ☐

Mapped Credentials

Credential	Provider

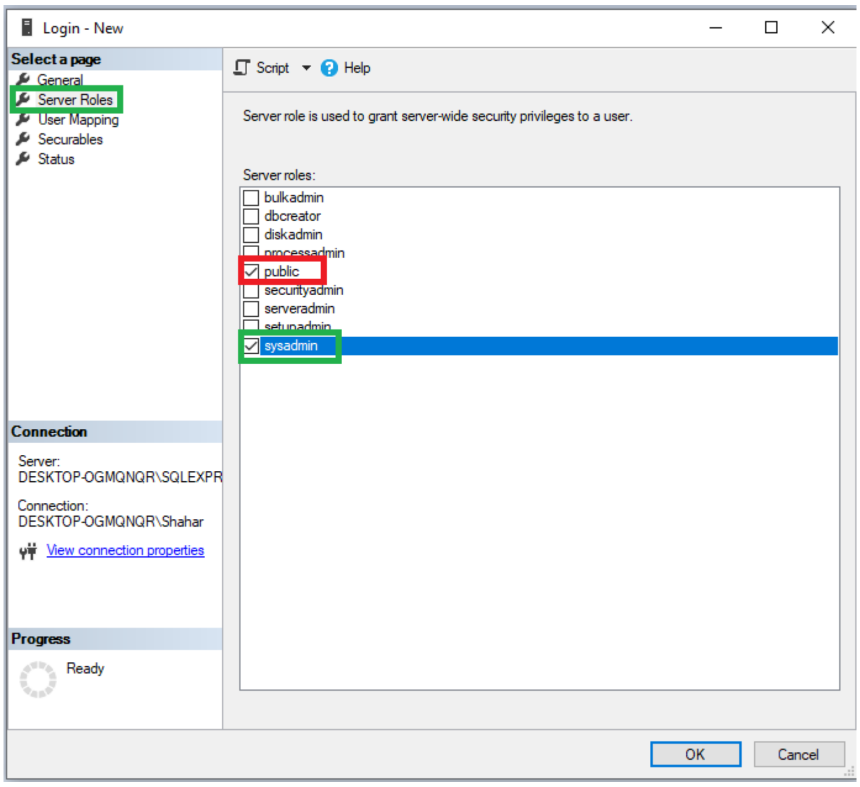
Default database: master

Default language: <default>

Server: DESKTOP-OGMQNQR\SQLEXPR

Connection: DESKTOP-OGMQNQR\Shahar

View connection properties



Server role is used to grant server-wide security privileges to a user.

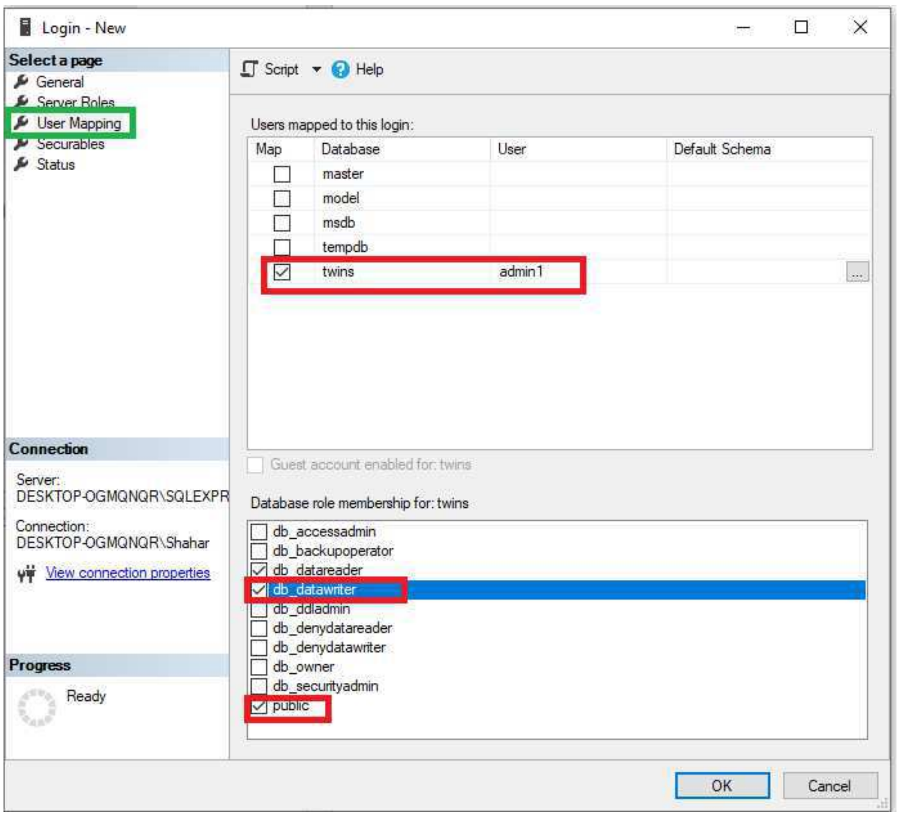
Server roles:

- ☐ bulkadmin
- ☐ dbcreator
- ☐ diskadmin
- ☐ processadmin
- ✓ public
- ☐ securityadmin
- ☐ serveradmin
- ☐ setupadmin
- ✓ sysadmin

Server: DESKTOP-OGMQNQR\SQLEXPR

Connection: DESKTOP-OGMQNQR\Shahar

View connection properties


Login - New

Select a page: General, Server Roles, User Mapping, Securables, Status

Users mapped to this login:

Map	Database	User	Default Schema
☐	master		
☐	model		
☐	msdb		
☐	tempdb		
✓	twins	admin1	

Guest account enabled for: twins

Database role membership for: twins

- ☐ db_accessadmin
- ☐ db_backupoperator
- ✓ db_datareader
- ✓ db_datawriter
- ☐ db_ddladmin
- ☐ db_denydatareader
- ☐ db_denydatawriter
- ☐ db_owner
- ☐ db_securityadmin
- ✓ public

Connection

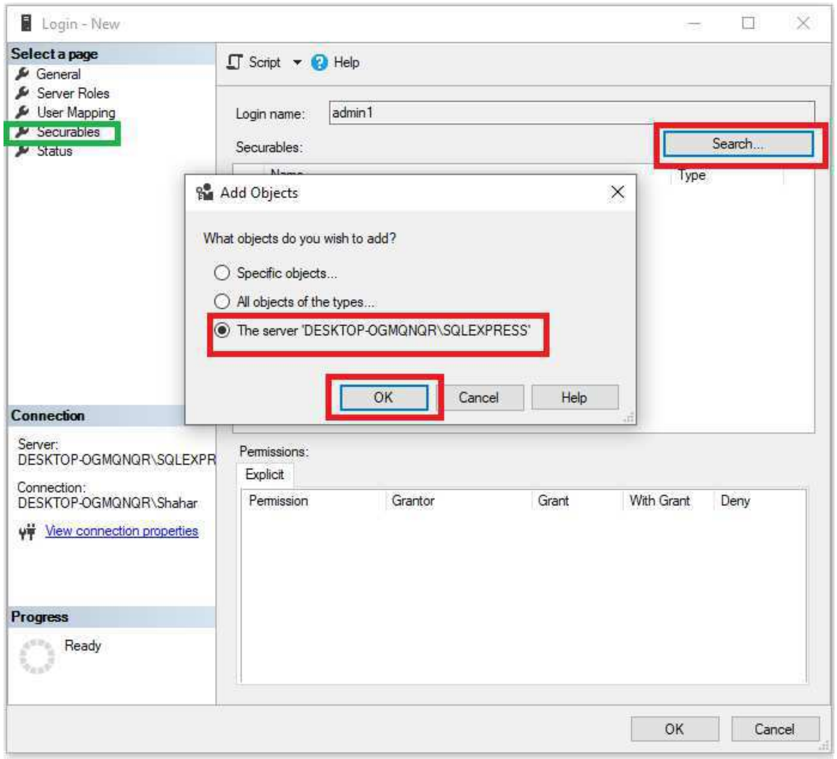
Server:
DESKTOP-OGMQNQR\SQLEXPR

Connection:
DESKTOP-OGMQNQR\Shahar

View connection properties

Progress

Ready


Login - New

Select a page: General, Server Roles, User Mapping, Securables, Status

Login name: admin1

Securables: Search...

Add Objects

What objects do you wish to add?

- ○ Specific objects...
- ○ All objects of the types...
- ● The server 'DESKTOP-OGMQNQR\SQLEXPRESS'

OK Cancel Help

Permissions: Explicit

Permission	Grantor	Grant	With Grant	Deny

Connection

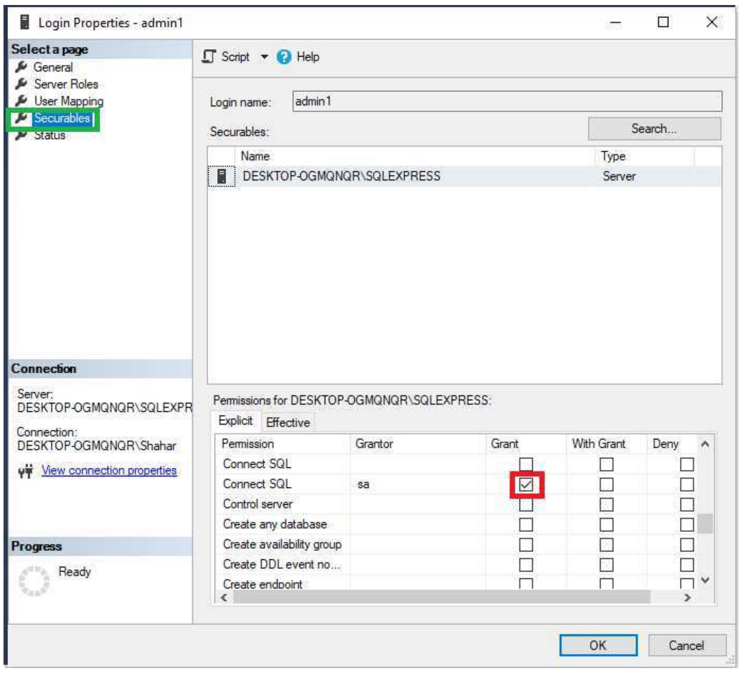
Server:
DESKTOP-OGMQNQR\SQLEXPR

Connection:
DESKTOP-OGMQNQR\Shahar

View connection properties

Progress

Ready



Login Properties - admin1

Select a page
- General
- Server Roles
- User Mapping
- Securables
- Status

Script ▾ Help

Login name: admin1

Securables: Search...

Name	Type
DESKTOP-OGMQNQR\SQLEXPRESS	Server

Permissions for DESKTOP-OGMQNQR\SQLEXPRESS:

Explicit | Effective

Permission	Grantor	Grant	With Grant	Deny
Connect SQL		☐	☐	☐
Connect SQL	sa	☑	☐	☐
Control server		☐	☐	☐
Create any database		☐	☐	☐
Create availability group		☐	☐	☐
Create DDL event no...		☐	☐	☐
Create endpoint		☐	☐	☐

Connection

Server:
DESKTOP-OGMQNQR\SQLEXPR

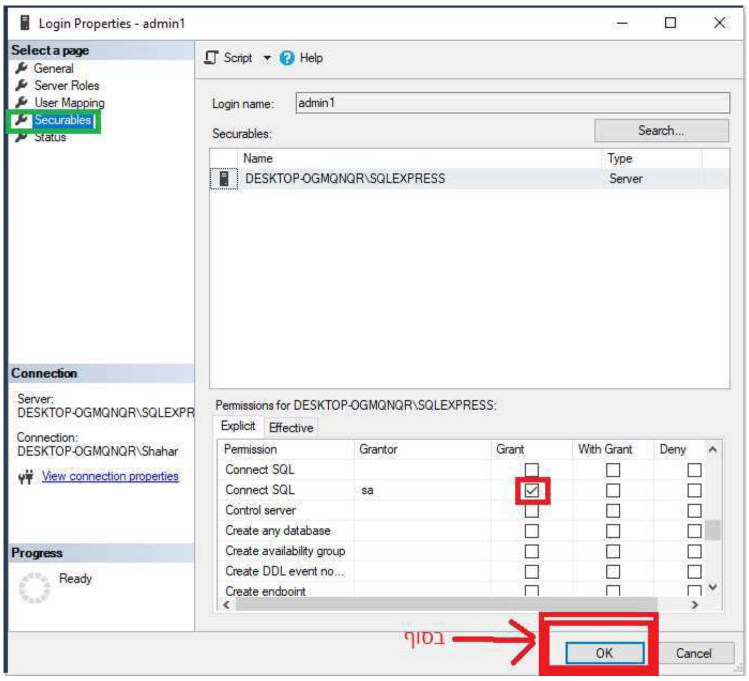
Connection:
DESKTOP-OGMQNQR\Shahar

View connection properties

Progress

Ready

OK | Cancel



Login Properties - admin1

Select a page
- General
- Server Roles
- User Mapping
- Securables
- Status

Script ▾ Help

Login name: admin1

Securables: Search...

Name	Type
DESKTOP-OGMQNQR\SQLEXPRESS	Server

Permissions for DESKTOP-OGMQNQR\SQLEXPRESS:

Explicit | Effective

Permission	Grantor	Grant	With Grant	Deny
Connect SQL		☐	☐	☐
Connect SQL	sa	☑	☐	☐
Control server		☐	☐	☐
Create any database		☐	☐	☐
Create availability group		☐	☐	☐
Create DDL event no...		☐	☐	☐
Create endpoint		☐	☐	☐

Connection

Server:
DESKTOP-OGMQNQR\SQLEXPR

Connection:
DESKTOP-OGMQNQR\Shahar

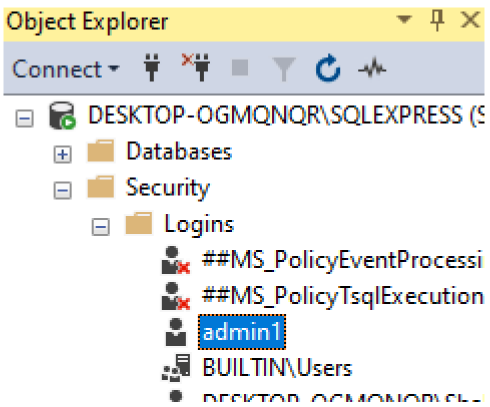
View connection properties

Progress

Ready

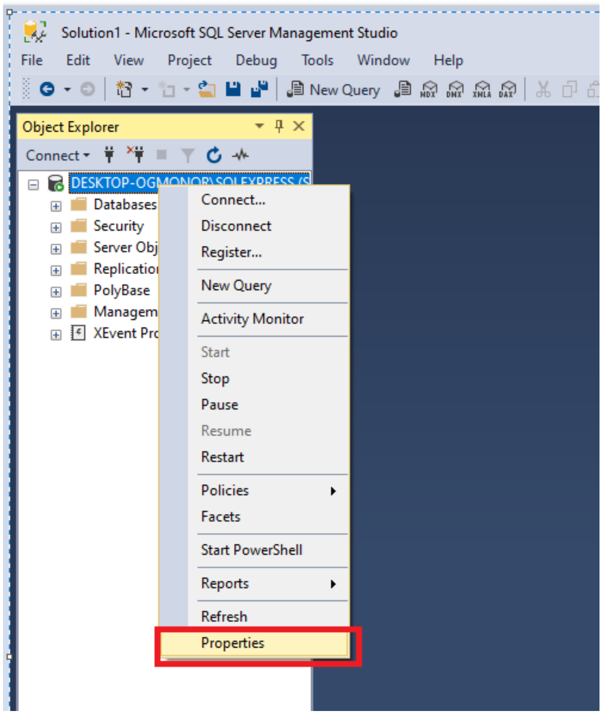
OK | Cancel

כעת נוצר משתמש admin1 admin1:

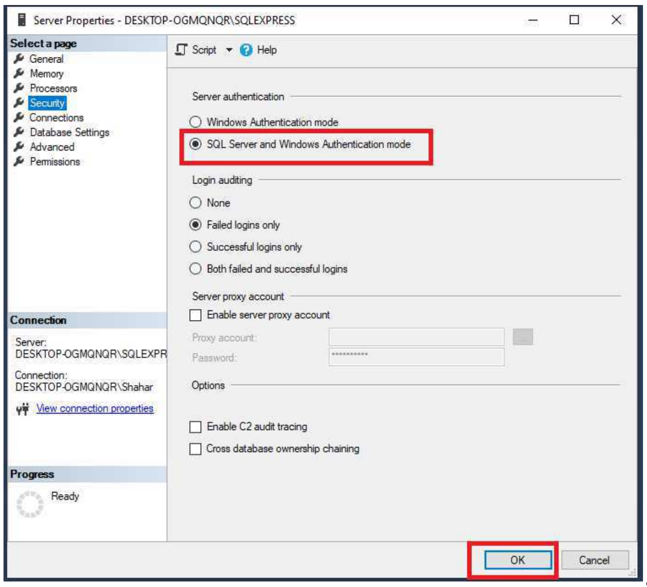


<u>עדכון ההגדרה להתחברות מרחוק:</u>

כניסה להגדרות השרת:



עדכון ההגדרות:


Server Properties - DESKTOP-OGMQNQR\SQLEXPRESS

Select a page: General, Memory, Processors, Security, Connections, Database Settings, Advanced, Permissions

Server authentication
- Windows Authentication mode
- [✓ SQL Server and Windows Authentication mode]

Login auditing
- None
- [✓ Failed logins only]
- Successful logins only
- Both failed and successful logins

Server proxy account
- Enable server proxy account

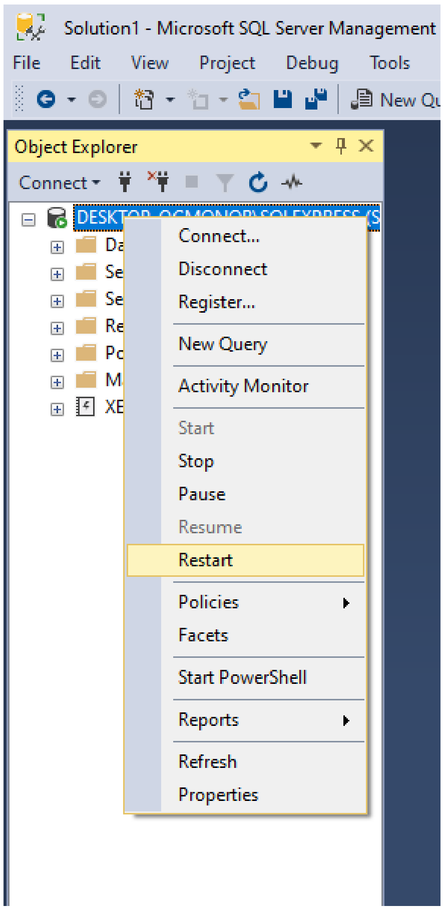
Proxy account:
Password: ***********

Options
- Enable C2 audit tracing
- Cross database ownership chaining

Connection
Server:
DESKTOP-OGMQNQR\SQLEXPR
Connection:
DESKTOP-OGMQNQR\Shahar
View connection properties

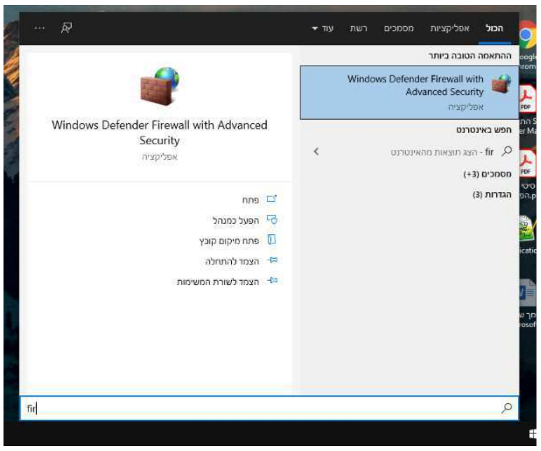
Progress
Ready

אתחול השרת:

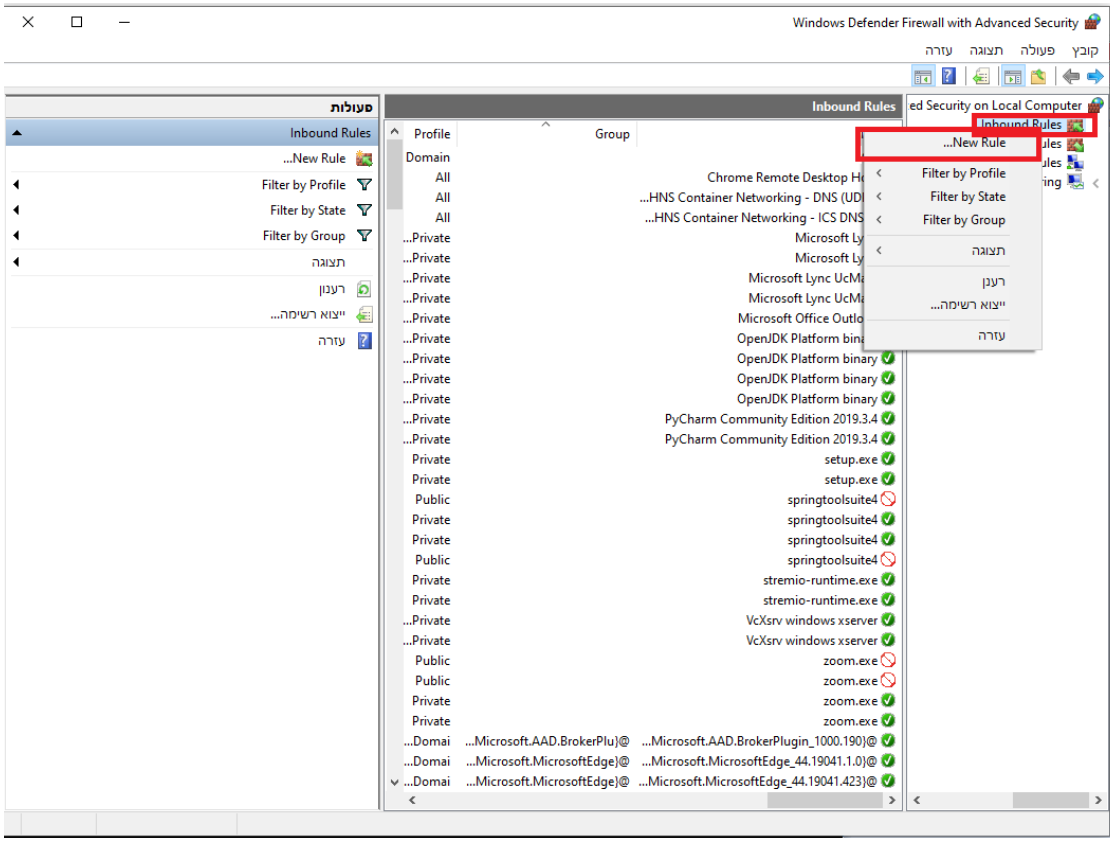

Solution1 - Microsoft SQL Server Management
File
Edit
View
Project
Debug
Tools
New Qu
Object Explorer
Connect
DESKTOP-OCMONOP\SQLEXPRESS (S
Connect...
Disconnect
Register...
New Query
Activity Monitor
Start
Stop
Pause
Resume
Restart
Policies
Facets
Start PowerShell
Reports
Refresh
Properties

<u>אי חסימת פורט 1433 (הפורט הדיפולטי של MsSQL Server) ע"י הfirewall:</u>

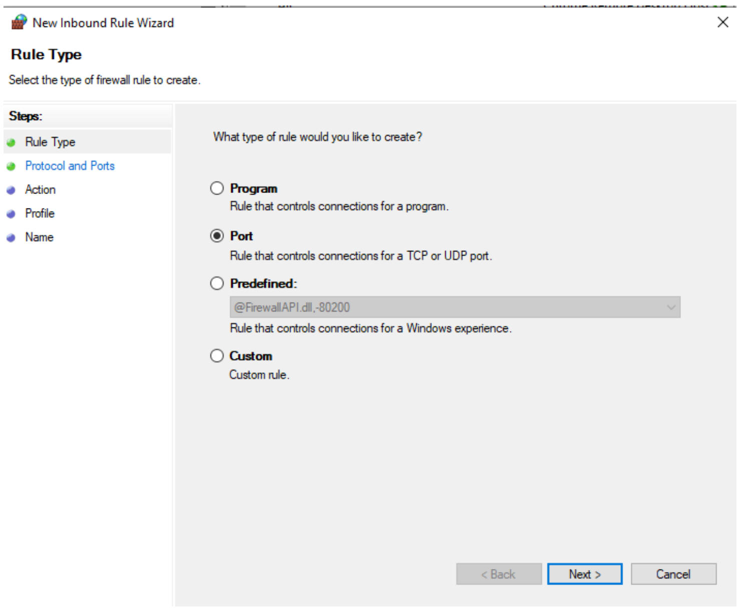
כנס להגדרות firewall:



לאחר מכן:



יש למלא את החלון לפי ההוראות הבאות:


New Inbound Rule Wizard

Rule Type

Select the type of firewall rule to create.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What type of rule would you like to create?

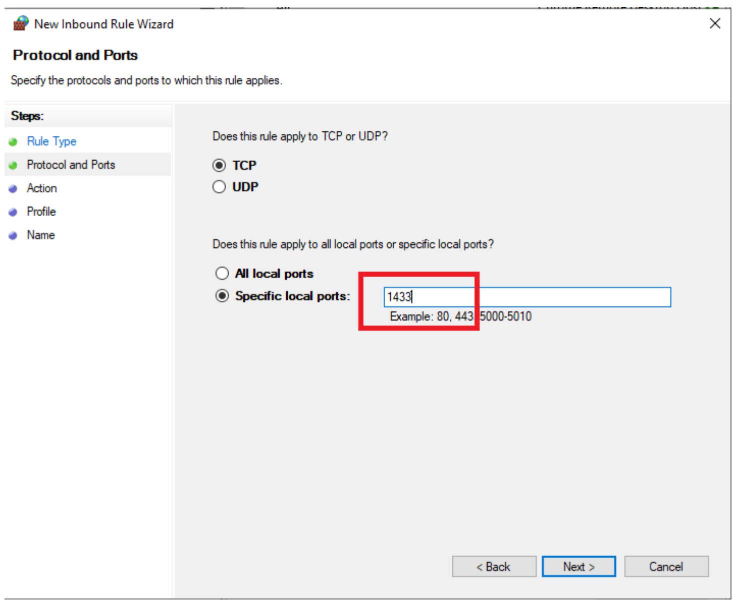
( ) **Program**
Rule that controls connections for a program.

(●) **Port**
Rule that controls connections for a TCP or UDP port.

( ) **Predefined:**
@FirewallAPI.dll,-80200
Rule that controls connections for a Windows experience.

( ) **Custom**
Custom rule.

< Back | Next > | Cancel


New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

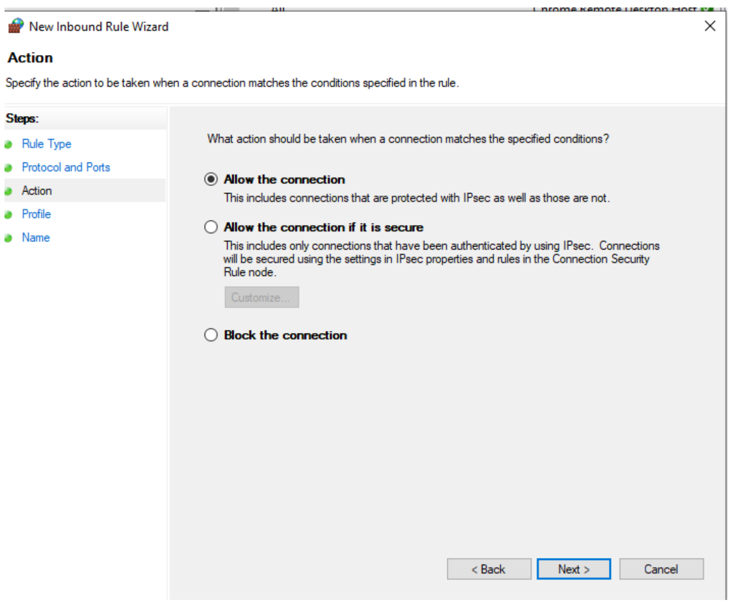
Does this rule apply to TCP or UDP?

(●) **TCP**
( ) **UDP**

Does this rule apply to all local ports or specific local ports?

( ) **All local ports**
(●) **Specific local ports:** 1433
Example: 80, 443, 5000-5010

< Back | Next > | Cancel



New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

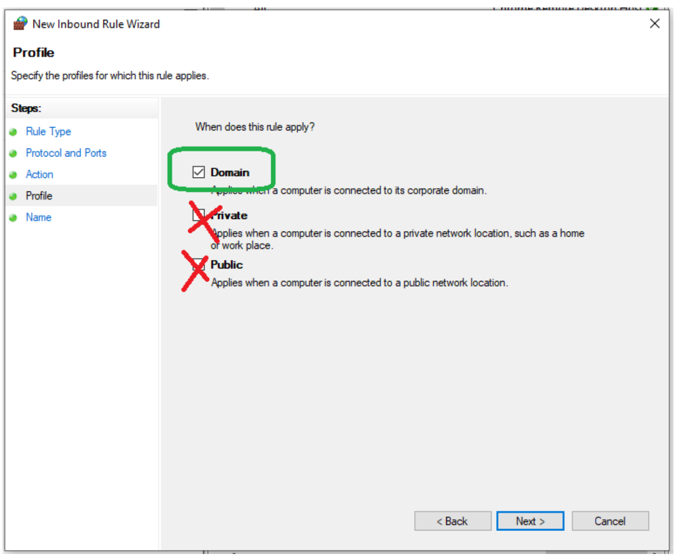
Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What action should be taken when a connection matches the specified conditions?

● **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

○ **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

○ **Block the connection**



New Inbound Rule Wizard

Profile

Specify the profiles for which this rule applies.

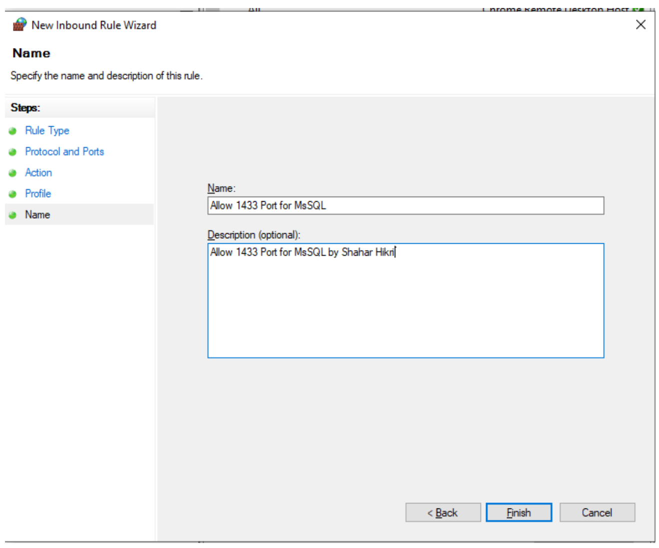
Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

When does this rule apply?

☑ **Domain**
Applies when a computer is connected to its corporate domain.

☐ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☐ **Public**
Applies when a computer is connected to a public network location.



New Inbound Rule Wizard

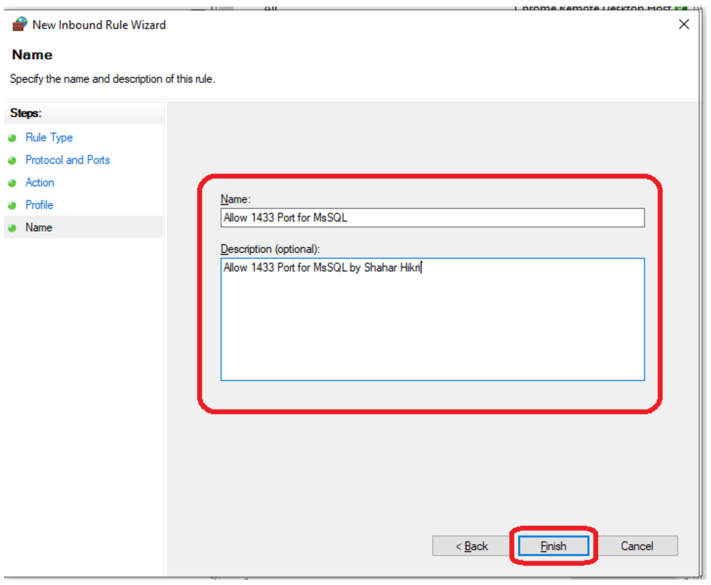
Name

Specify the name and description of this rule.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Name:
Allow 1433 Port for MsSQL

Description (optional):
Allow 1433 Port for MsSQL by Shahar Hikri


New Inbound Rule Wizard

Name

Specify the name and description of this rule.

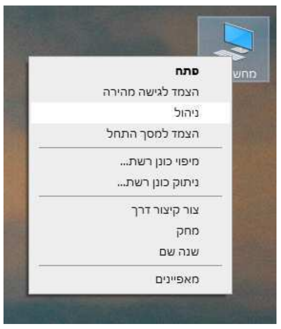
Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Name:
Allow 1433 Port for MsSQL

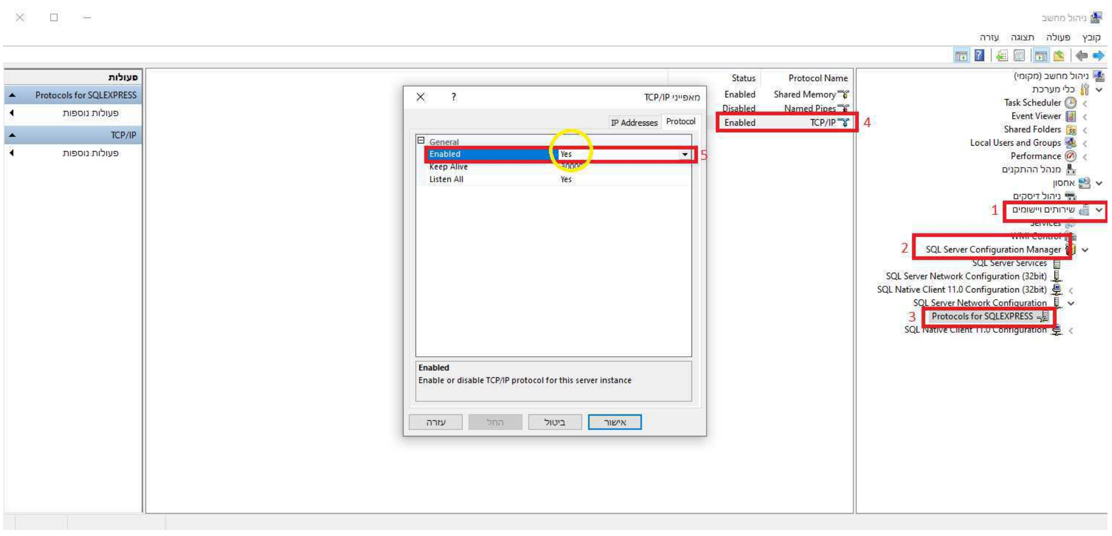
Description (optional):
Allow 1433 Port for MsSQL by Shahar Hikri

הפעלת הפורט 1433 לתקשורת עם השרת

כנס למסך "ניהול מחשב" ע"י לחצן ימני על "מחשב זה" (This PC) ואז על "ניהול":



לאחר מכן:



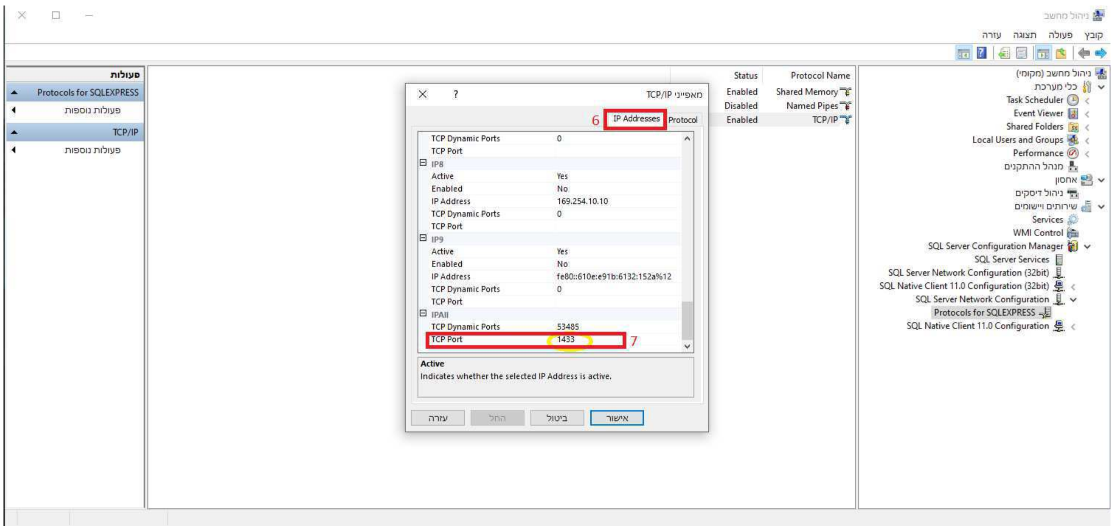
Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Disabled
TCP/IP	Enabled

מאפייני TCP/IP

Protocol | IP Addresses

General	
Enabled	Yes
Keep Alive	30000
Listen All	Yes

Enabled
Enable or disable TCP/IP protocol for this server instance



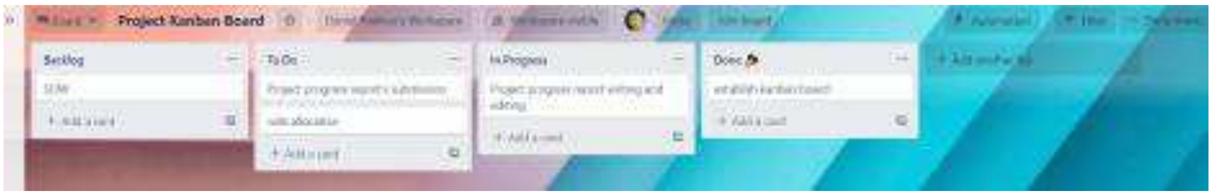
Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Disabled
TCP/IP	Enabled

```
TCP Dynamic Ports       0
TCP Port
⊟ IP8
Active                  Yes
Enabled                 No
IP Address              169.254.10.10
TCP Dynamic Ports       0
TCP Port
⊟ IP9
Active                  Yes
Enabled                 No
IP Address              fe80::610e:e91b:6132:152a%12
TCP Dynamic Ports       0
TCP Port
⊟ IPAll
TCP Dynamic Ports       53485
TCP Port                1433
```

Active
Indicates whether the selected IP Address is active.

Sprint 1

project initiation:



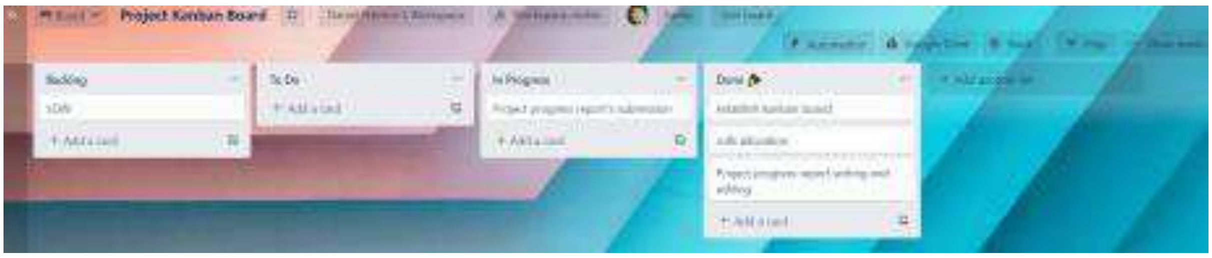
Backlog: SOW

To Do: Project progress report's submission; role allocation

In Progress: Project progress report writing and editing

Done: establish kanban board

after Project progress report submission:



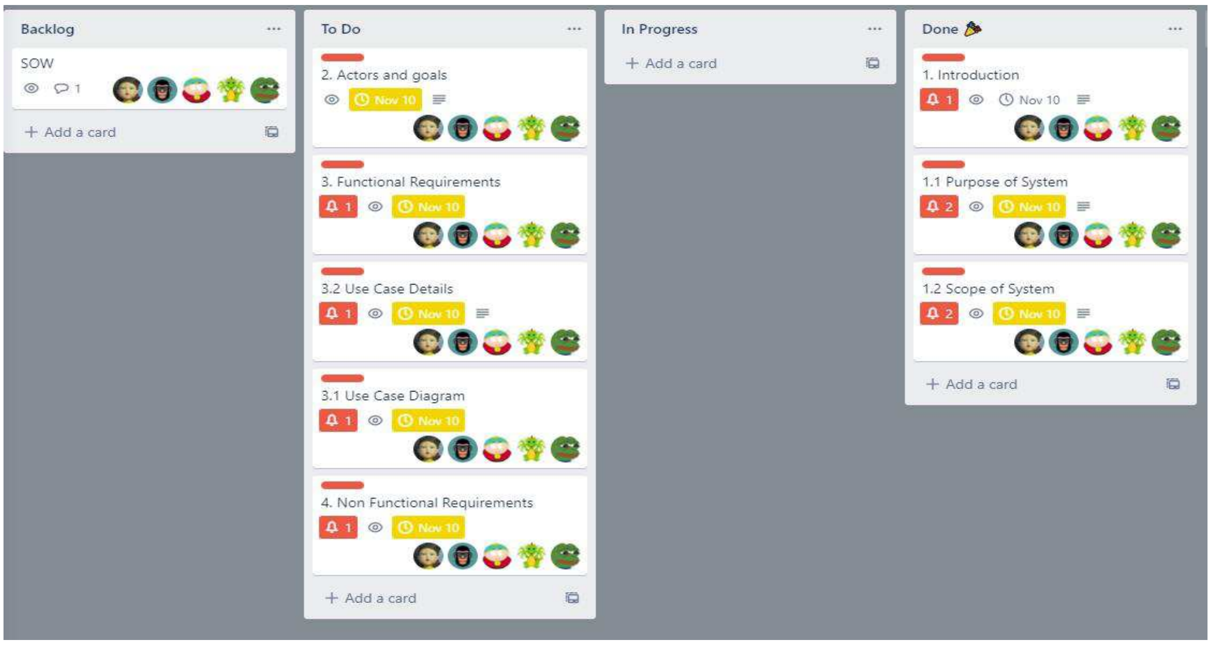
Backlog: SOW

In Progress: Project progress report's submission

Done: establish kanban board; role allocation; Project progress report writing and editing

Sprint 2

Start of the sprint - 27.10 :



Backlog

- SOW — 1
- + Add a card

To Do

- 2. Actors and goals — Nov 10
- 3. Functional Requirements — 1 — Nov 10
- 3.2 Use Case Details — 1 — Nov 10
- 3.1 Use Case Diagram — 1 — Nov 10
- 4. Non Functional Requirements — 1 — Nov 10
- + Add a card

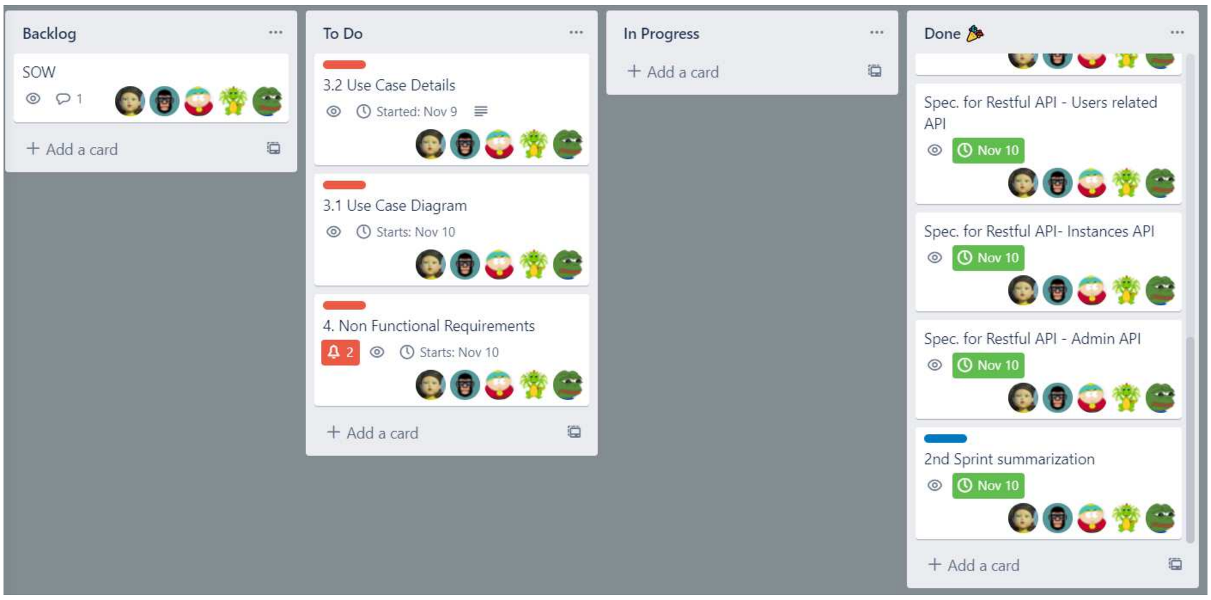
In Progress

- + Add a card

Done

- 1. Introduction — 1 — Nov 10
- 1.1 Purpose of System — 2 — Nov 10
- 1.2 Scope of System — 2 — Nov 10
- + Add a card

End of the sprint - 09.11:



Backlog

- SOW — 1
- + Add a card

To Do

- 3.2 Use Case Details — Started: Nov 9
- 3.1 Use Case Diagram — Starts: Nov 10
- 4. Non Functional Requirements — 2 — Starts: Nov 10
- + Add a card

In Progress

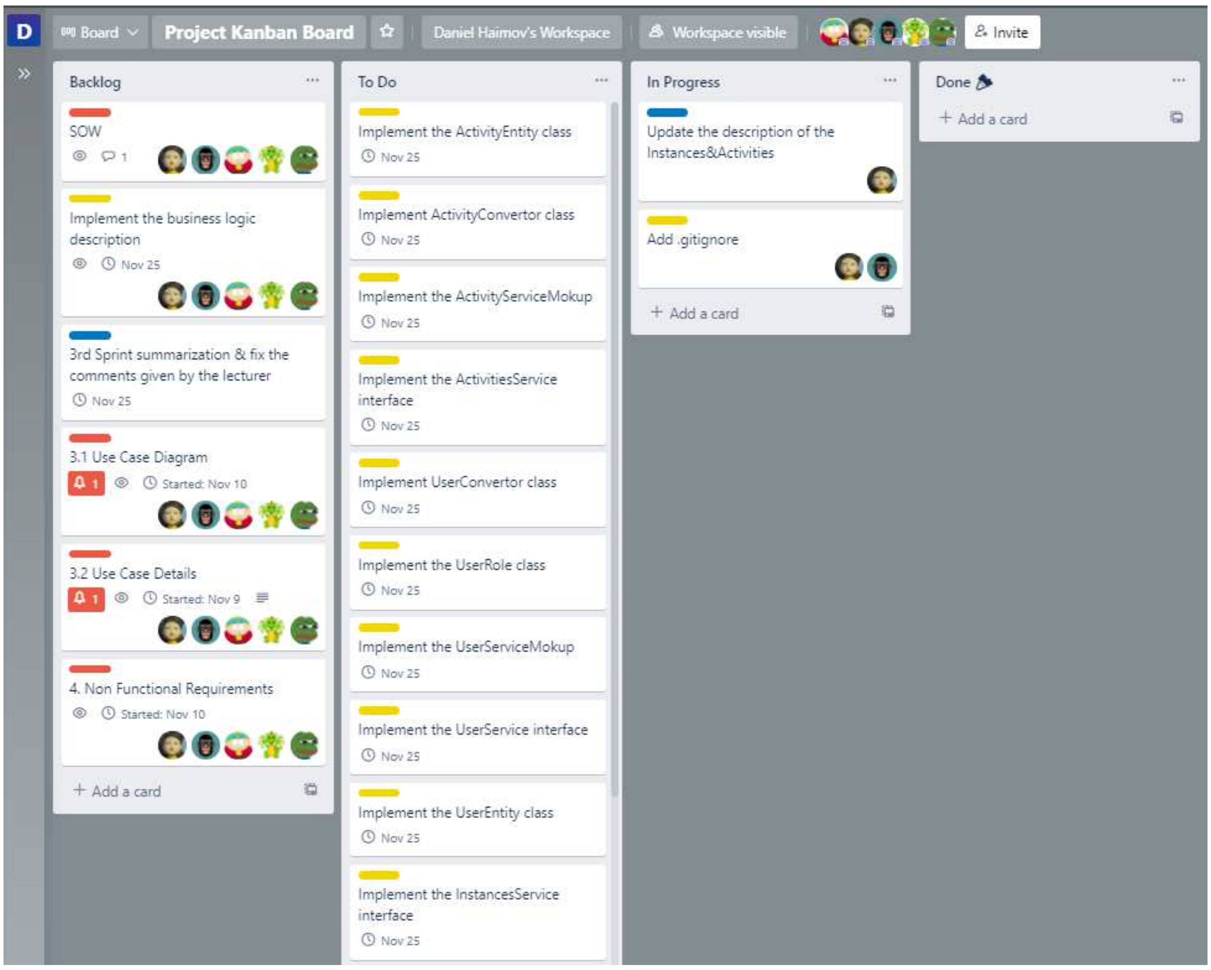
- + Add a card

Done

- Spec. for Restful API - Users related API — Nov 10
- Spec. for Restful API- Instances API — Nov 10
- Spec. for Restful API - Admin API — Nov 10
- 2nd Sprint summarization — Nov 10
- + Add a card

Sprint 3

Start of the sprint - 10.11.21 :

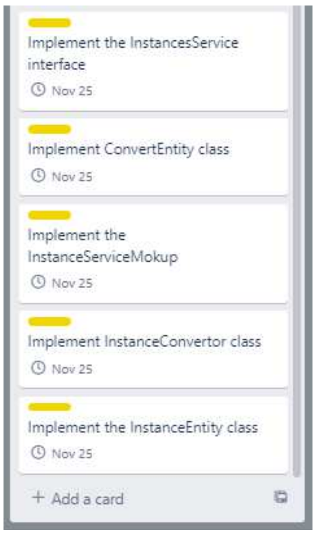


Backlog
- SOW — 1
- Implement the business logic description — Nov 25
- 3rd Sprint summarization & fix the comments given by the lecturer — Nov 25
- 3.1 Use Case Diagram — 1 — Started: Nov 10
- 3.2 Use Case Details — 1 — Started: Nov 9
- 4. Non Functional Requirements — Started: Nov 10
- + Add a card

To Do
- Implement the ActivityEntity class — Nov 25
- Implement ActivityConvertor class — Nov 25
- Implement the ActivityServiceMokup — Nov 25
- Implement the ActivitiesService interface — Nov 25
- Implement UserConvertor class — Nov 25
- Implement the UserRole class — Nov 25
- Implement the UserServiceMokup — Nov 25
- Implement the UserService interface — Nov 25
- Implement the UserEntity class — Nov 25
- Implement the InstancesService interface — Nov 25

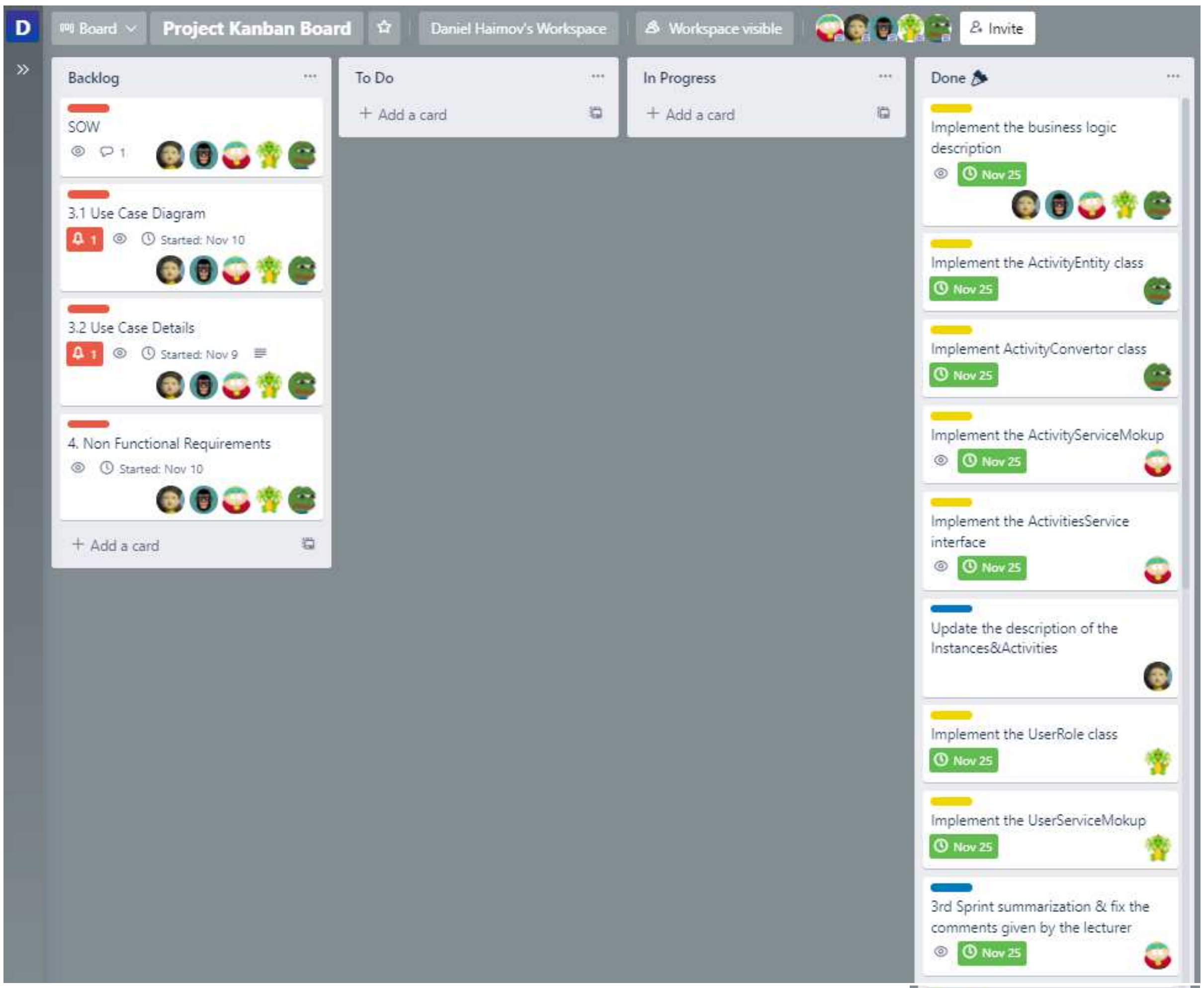
In Progress
- Update the description of the Instances&Activities
- Add .gitignore
- + Add a card

Done
- + Add a card



- Implement the InstancesService interface — Nov 25
- Implement ConvertEntity class — Nov 25
- Implement the InstanceServiceMokup — Nov 25
- Implement InstanceConvertor class — Nov 25
- Implement the InstanceEntity class — Nov 25
- + Add a card

End of the sprint - 24.11.21:



Backlog

- SOW — 1
- 3.1 Use Case Diagram — 1 — Started: Nov 10
- 3.2 Use Case Details — 1 — Started: Nov 9
- 4. Non Functional Requirements — Started: Nov 10

+ Add a card

To Do

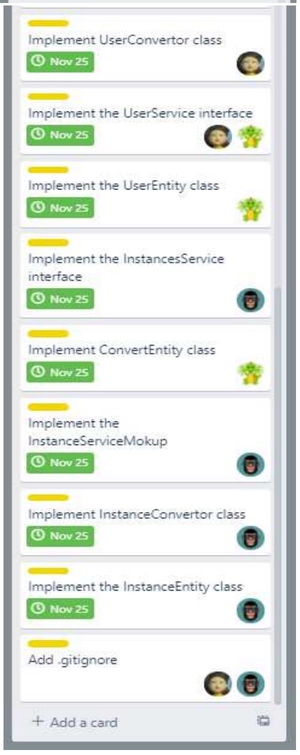
+ Add a card

In Progress

+ Add a card

Done

- Implement the business logic description — Nov 25
- Implement the ActivityEntity class — Nov 25
- Implement ActivityConvertor class — Nov 25
- Implement the ActivityServiceMokup — Nov 25
- Implement the ActivitiesService interface — Nov 25
- Update the description of the Instances&Activities
- Implement the UserRole class — Nov 25
- Implement the UserServiceMokup — Nov 25
- 3rd Sprint summarization & fix the comments given by the lecturer — Nov 25

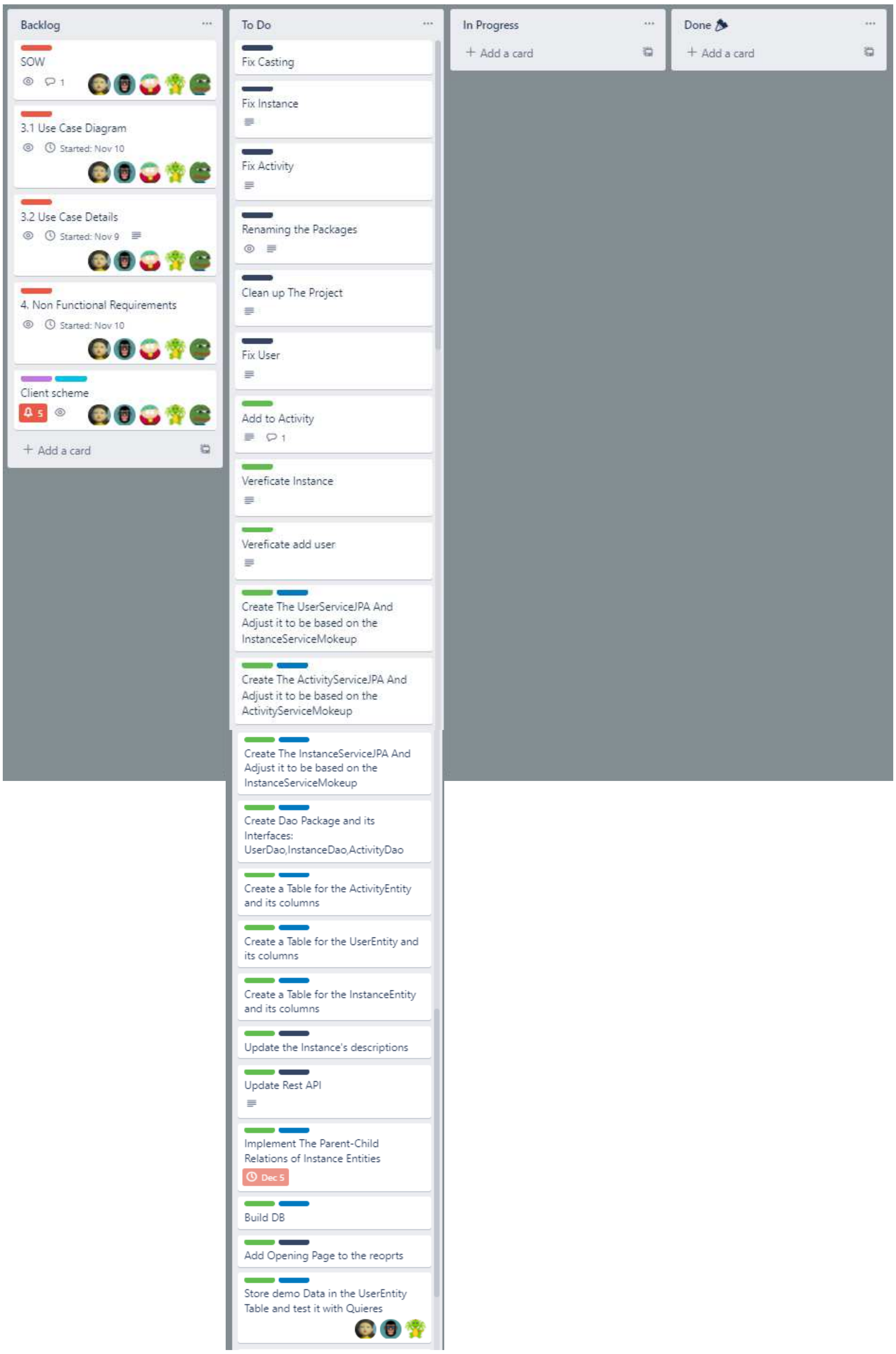


- Implement UserConvertor class — Nov 25
- Implement the UserService interface — Nov 25
- Implement the UserEntity class — Nov 25
- Implement the InstancesService interface — Nov 25
- Implement ConvertEntity class — Nov 25
- Implement the InstanceServiceMokup — Nov 25
- Implement InstanceConvertor class — Nov 25
- Implement the InstanceEntity class — Nov 25
- Add .gitignore

+ Add a card

Sprint 4

Start of the sprint - 24.11.21 :



Backlog

- SOW — 1
- 3.1 Use Case Diagram — Started: Nov 10
- 3.2 Use Case Details — Started: Nov 9
- 4. Non Functional Requirements — Started: Nov 10
- Client scheme — 5
- + Add a card

To Do

- Fix Casting
- Fix Instance
- Fix Activity
- Renaming the Packages
- Clean up The Project
- Fix User
- Add to Activity — 1
- Vereficate Instance
- Vereficate add user
- Create The UserServiceJPA And Adjust it to be based on the InstanceServiceMokeup
- Create The ActivityServiceJPA And Adjust it to be based on the ActivityServiceMokeup
- Create The InstanceServiceJPA And Adjust it to be based on the InstanceServiceMokeup
- Create Dao Package and its Interfaces: UserDao,InstanceDao,ActivityDao
- Create a Table for the ActivityEntity and its columns
- Create a Table for the UserEntity and its columns
- Create a Table for the InstanceEntity and its columns
- Update the Instance's descriptions
- Update Rest API
- Implement The Parent-Child Relations of Instance Entities — Dec 5
- Build DB
- Add Opening Page to the reoprts
- Store demo Data in the UserEntity Table and test it with Quieres

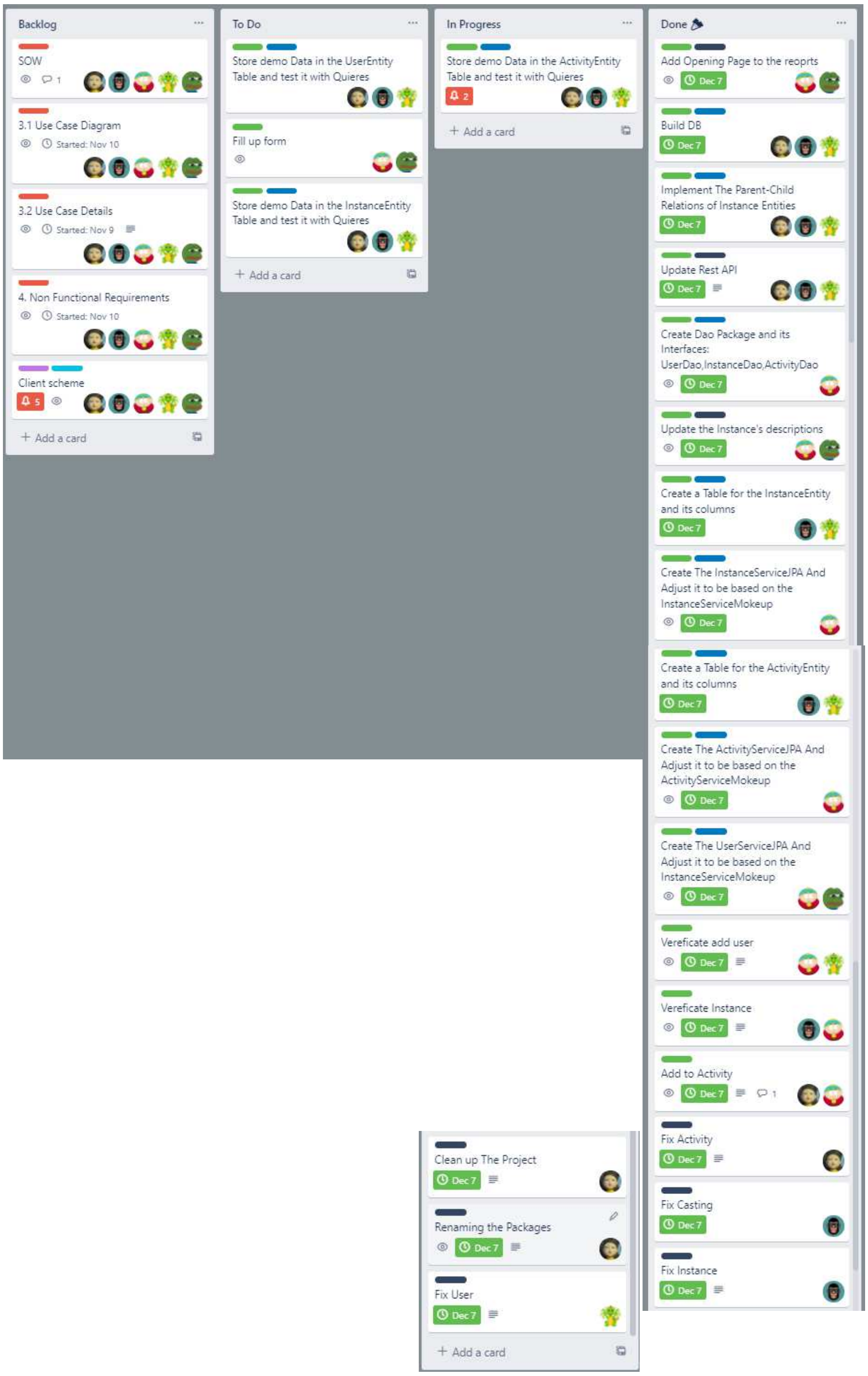
In Progress

- + Add a card

Done

- + Add a card

End of the sprint - 07.12.21:



Backlog

- SOW — 1
- 3.1 Use Case Diagram — Started: Nov 10
- 3.2 Use Case Details — Started: Nov 9
- 4. Non Functional Requirements — Started: Nov 10
- Client scheme — 5
- + Add a card

To Do

- Store demo Data in the UserEntity Table and test it with Quieres
- Fill up form
- Store demo Data in the InstanceEntity Table and test it with Quieres
- + Add a card

In Progress

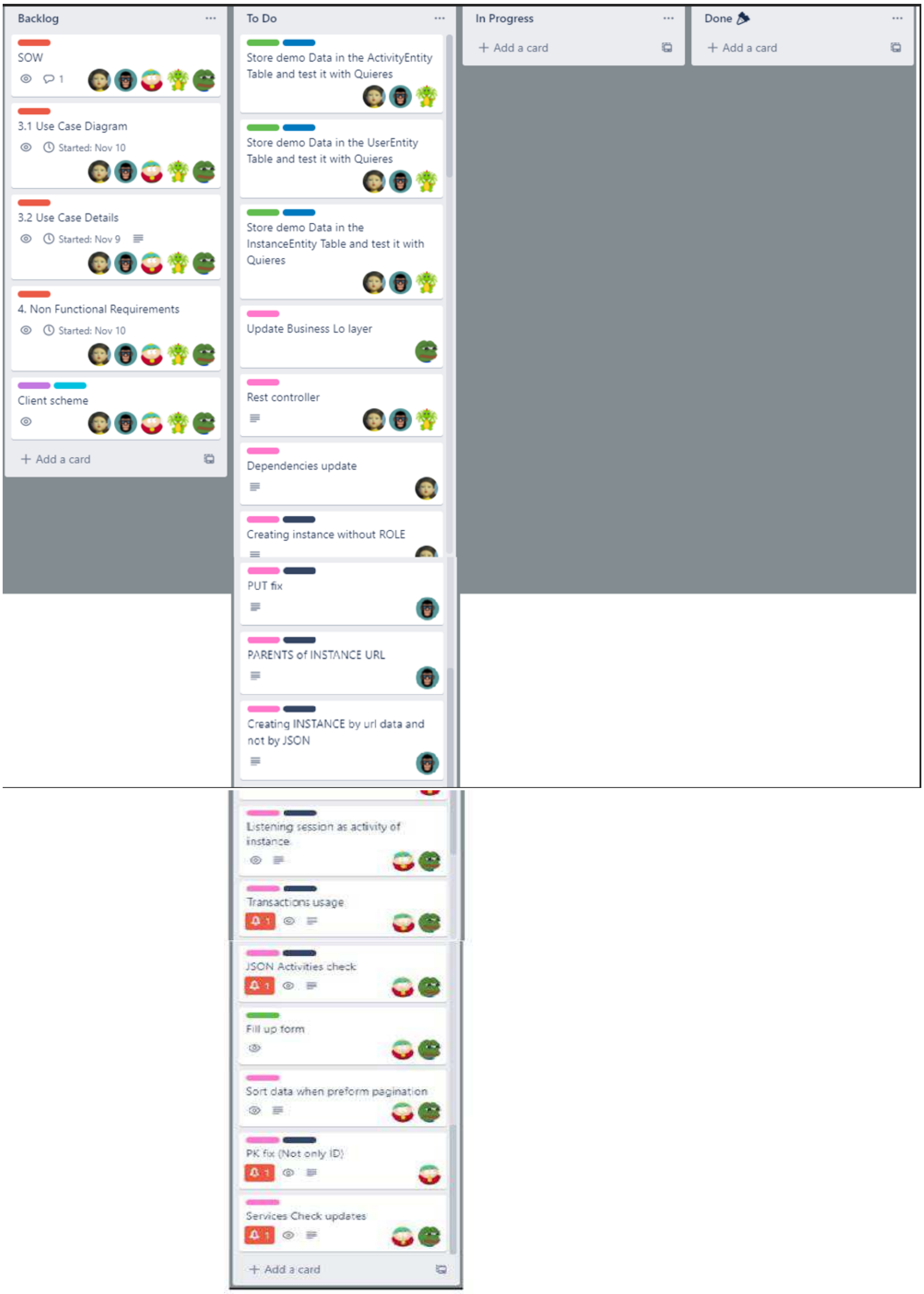
- Store demo Data in the ActivityEntity Table and test it with Quieres — 2
- + Add a card
- Clean up The Project — Dec 7
- Renaming the Packages — Dec 7
- Fix User — Dec 7
- + Add a card

Done

- Add Opening Page to the reoprts — Dec 7
- Build DB — Dec 7
- Implement The Parent-Child Relations of Instance Entities — Dec 7
- Update Rest API — Dec 7
- Create Dao Package and its Interfaces: UserDao,InstanceDao,ActivityDao — Dec 7
- Update the Instance's descriptions — Dec 7
- Create a Table for the InstanceEntity and its columns — Dec 7
- Create The InstanceServiceJPA And Adjust it to be based on the InstanceServiceMokeup — Dec 7
- Create a Table for the ActivityEntity and its columns — Dec 7
- Create The ActivityServiceJPA And Adjust it to be based on the ActivityServiceMokeup — Dec 7
- Create The UserServiceJPA And Adjust it to be based on the InstanceServiceMokeup — Dec 7
- Vereficate add user — Dec 7
- Vereficate Instance — Dec 7
- Add to Activity — Dec 7 — 1
- Fix Activity — Dec 7
- Fix Casting — Dec 7
- Fix Instance — Dec 7

Sprint 5

Start of the sprint - 8.12.21 :


Backlog

SOW

1

3.1 Use Case Diagram

Started: Nov 10

3.2 Use Case Details

Started: Nov 9

4. Non Functional Requirements

Started: Nov 10

Client scheme

+ Add a card

To Do

Store demo Data in the ActivityEntity Table and test it with Quieres

Store demo Data in the UserEntity Table and test it with Quieres

Store demo Data in the InstanceEntity Table and test it with Quieres

Update Business Lo layer

Rest controller

Dependencies update

Creating instance without ROLE

PUT fix

PARENTS of INSTANCE URL

Creating INSTANCE by url data and not by JSON

Listening session as activity of instance

Transactions usage

1

JSON Activities check

1

Fill up form

Sort data when preform pagination

PK fix (Not only ID)

1

Services Check updates

1

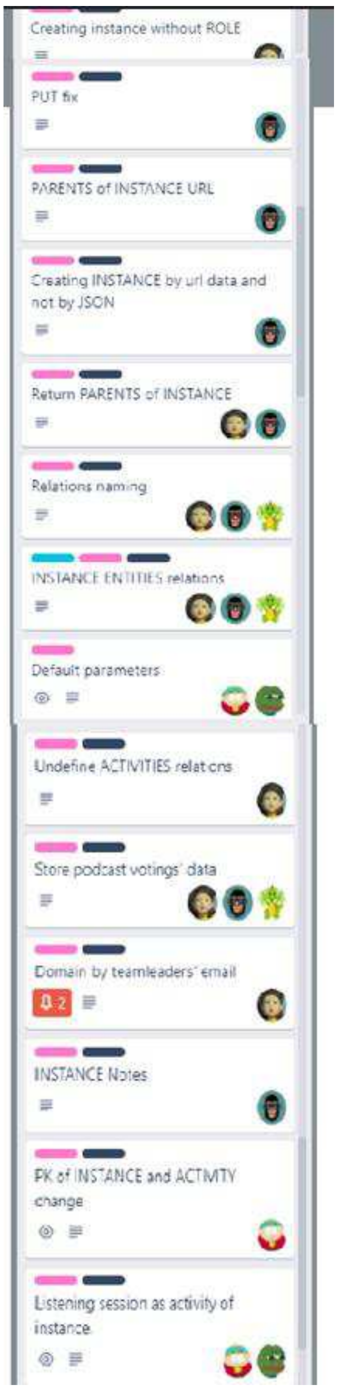
+ Add a card

In Progress

+ Add a card

Done

+ Add a card



Creating instance without ROLE

PUT fix

PARENTS of INSTANCE URL

Creating INSTANCE by url data and not by JSON

Return PARENTS of INSTANCE

Relations naming

INSTANCE ENTITIES relations

Default parameters

Undefine ACTIVITIES relations

Store podcast votings' data

Domain by teamleaders' email

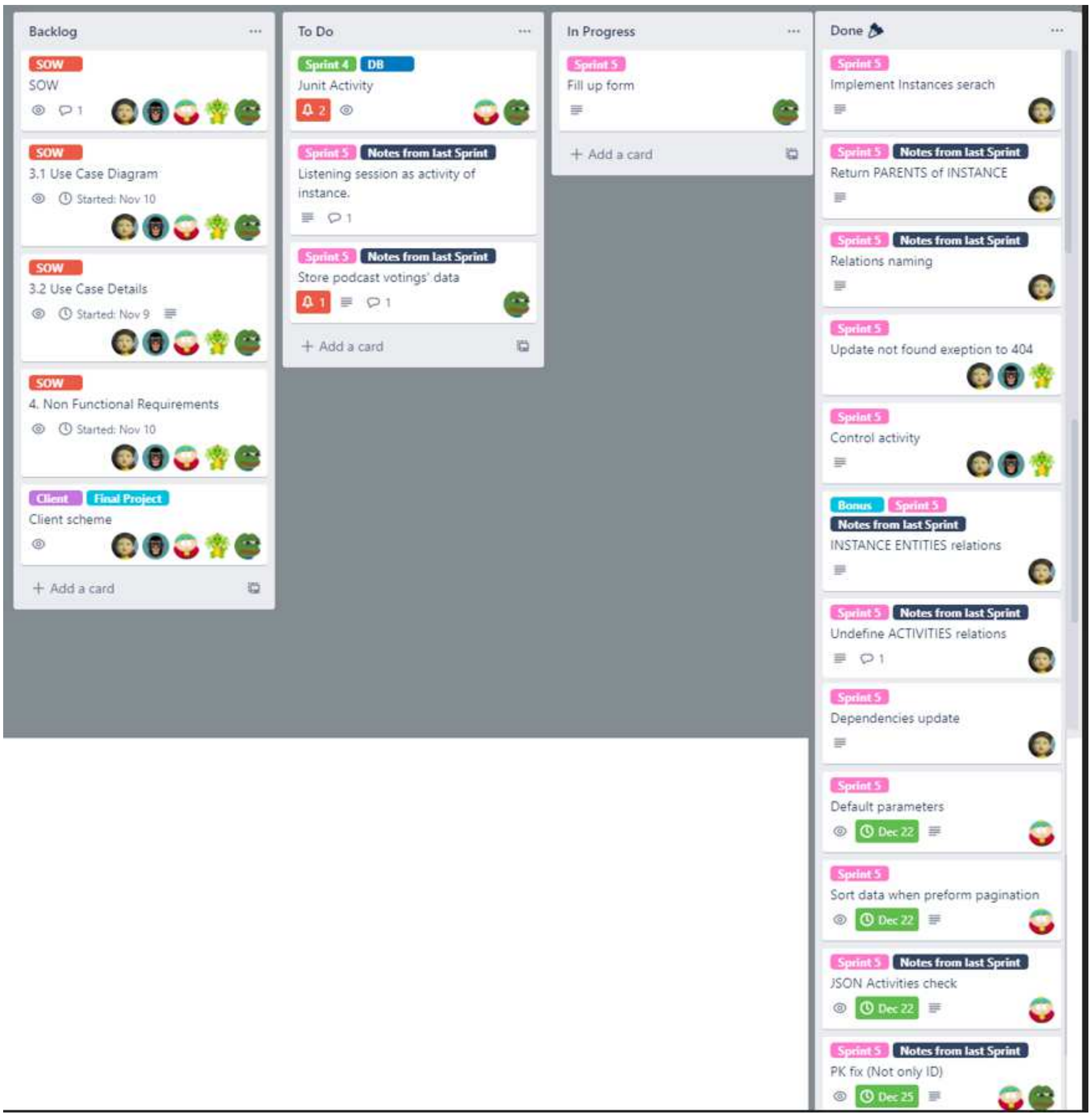
2

INSTANCE Notes

PK of INSTANCE and ACTIVITY change

Listening session as activity of instance

End of the sprint - 21.12.21:



Backlog

- SOW
  SOW
  1
- SOW
  3.1 Use Case Diagram
  Started: Nov 10
- SOW
  3.2 Use Case Details
  Started: Nov 9
- SOW
  4. Non Functional Requirements
  Started: Nov 10
- Client | Final Project
  Client scheme

+ Add a card

To Do

- Sprint 4 | DB
  Junit Activity
  2
- Sprint 5 | Notes from last Sprint
  Listening session as activity of instance.
  1
- Sprint 5 | Notes from last Sprint
  Store podcast votings' data
  1 | 1

+ Add a card

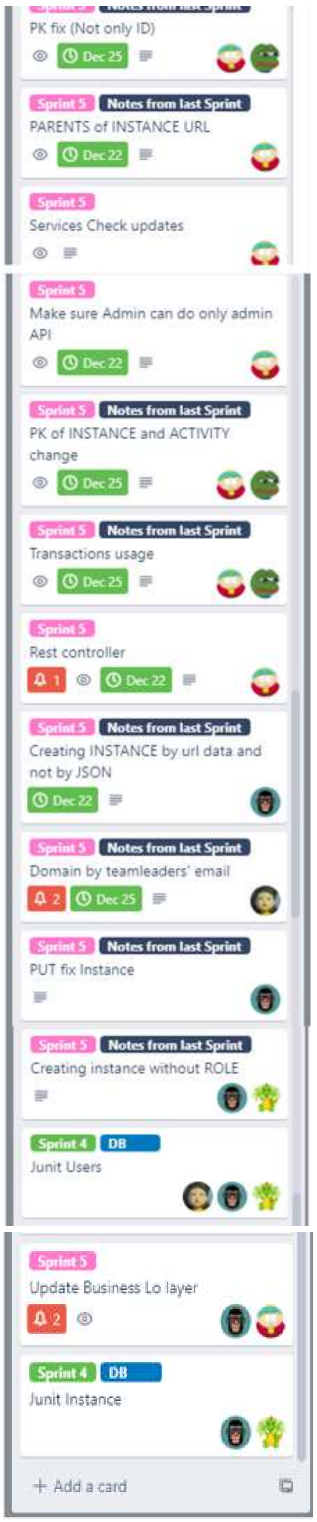
In Progress

- Sprint 5
  Fill up form

+ Add a card

Done

- Sprint 5
  Implement Instances serach
- Sprint 5 | Notes from last Sprint
  Return PARENTS of INSTANCE
- Sprint 5 | Notes from last Sprint
  Relations naming
- Sprint 5
  Update not found exeption to 404
- Sprint 5
  Control activity
- Bonus | Sprint 5 | Notes from last Sprint
  INSTANCE ENTITIES relations
- Sprint 5 | Notes from last Sprint
  Undefine ACTIVITIES relations
  1
- Sprint 5
  Dependencies update
- Sprint 5
  Default parameters
  Dec 22
- Sprint 5
  Sort data when preform pagination
  Dec 22
- Sprint 5 | Notes from last Sprint
  JSON Activities check
  Dec 22
- Sprint 5 | Notes from last Sprint
  PK fix (Not only ID)
  Dec 25



PK fix (Not only ID)
Dec 25

Sprint 5 | Notes from last Sprint
PARENTS of INSTANCE URL
Dec 22

Sprint 5
Services Check updates

Sprint 5
Make sure Admin can do only admin API
Dec 22

Sprint 5 | Notes from last Sprint
PK of INSTANCE and ACTIVITY change
Dec 25

Sprint 5 | Notes from last Sprint
Transactions usage
Dec 25

Sprint 5
Rest controller
1 | Dec 22

Sprint 5 | Notes from last Sprint
Creating INSTANCE by url data and not by JSON
Dec 22

Sprint 5 | Notes from last Sprint
Domain by teamleaders' email
2 | Dec 25

Sprint 5 | Notes from last Sprint
PUT fix Instance

Sprint 5 | Notes from last Sprint
Creating instance without ROLE

Sprint 4 | DB
Junit Users

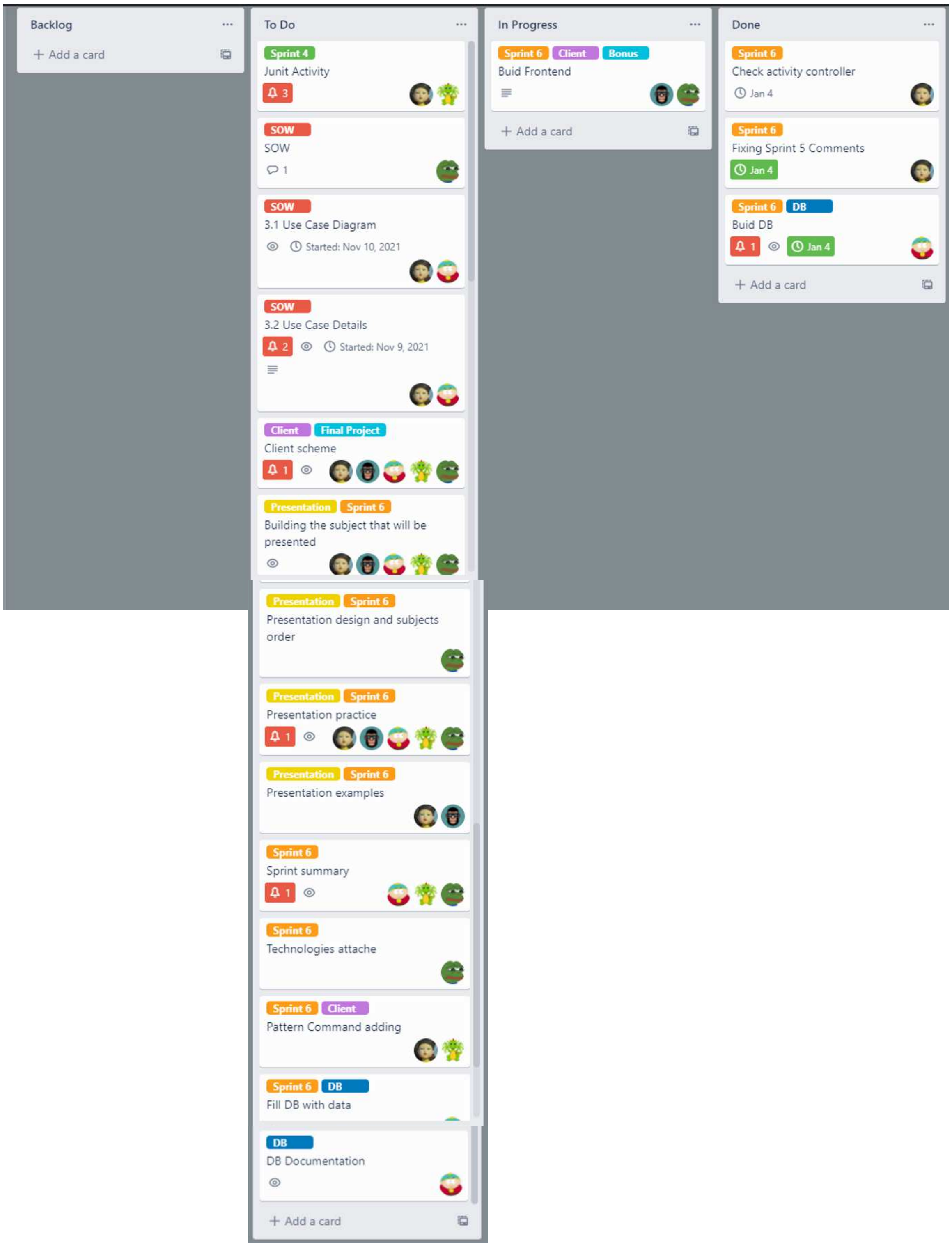
Sprint 5
Update Business Lo layer
2

Sprint 4 | DB
Junit Instance

+ Add a card

Sprint 6

Start of the sprint - 22.12.21 :



Backlog

- + Add a card

To Do

- Sprint 4 — Junit Activity — 3
- SOW — SOW — 1
- SOW — 3.1 Use Case Diagram — Started: Nov 10, 2021
- SOW — 3.2 Use Case Details — 2 — Started: Nov 9, 2021
- Client, Final Project — Client scheme — 1
- Presentation, Sprint 6 — Building the subject that will be presented
- Presentation, Sprint 6 — Presentation design and subjects order
- Presentation, Sprint 6 — Presentation practice — 1
- Presentation, Sprint 6 — Presentation examples
- Sprint 6 — Sprint summary — 1
- Sprint 6 — Technologies attache
- Sprint 6, Client — Pattern Command adding
- Sprint 6, DB — Fill DB with data
- DB — DB Documentation
- + Add a card

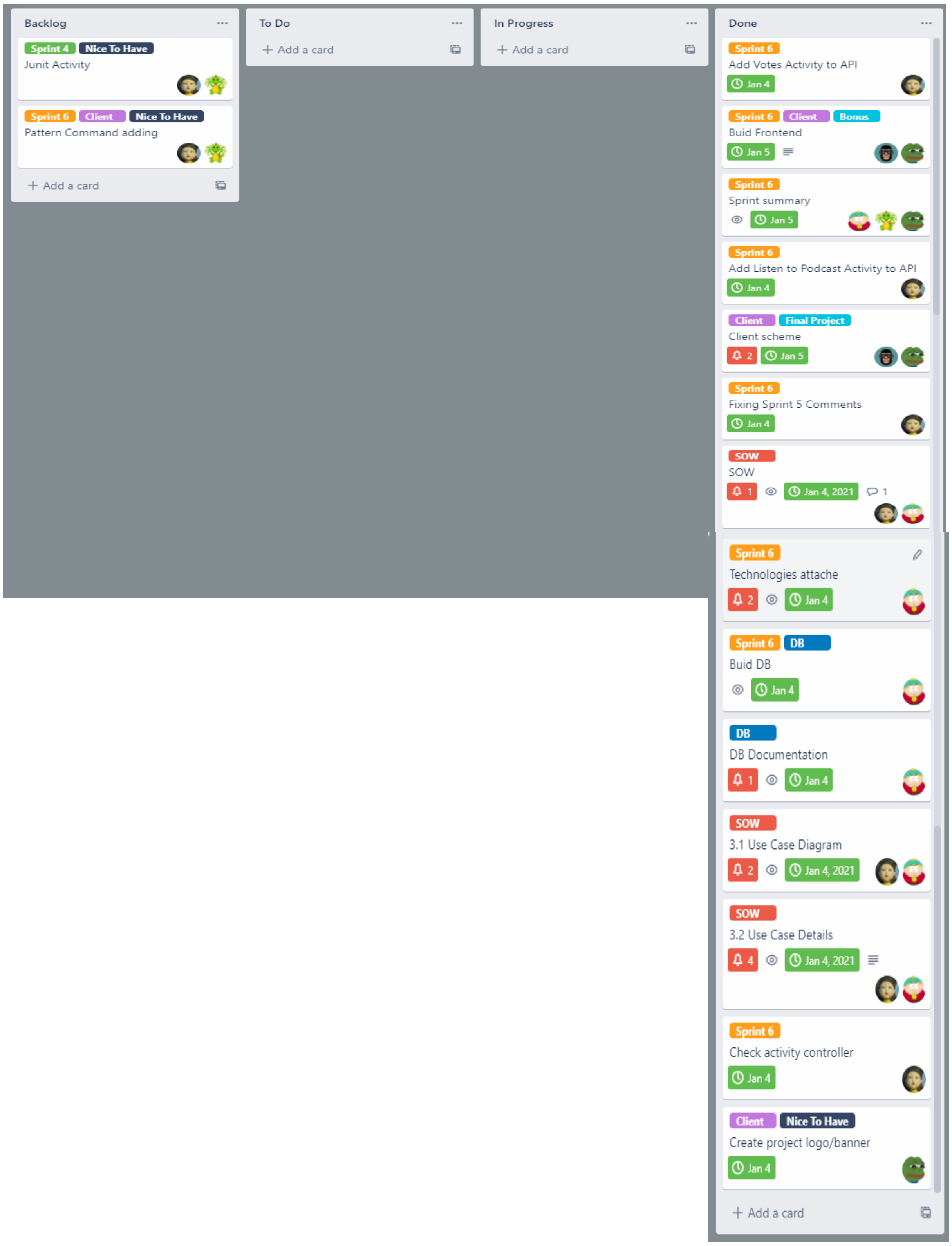
In Progress

- Sprint 6, Client, Bonus — Buid Frontend
- + Add a card

Done

- Sprint 6 — Check activity controller — Jan 4
- Sprint 6 — Fixing Sprint 5 Comments — Jan 4
- Sprint 6, DB — Buid DB — 1 — Jan 4
- + Add a card

End of the sprint - 04.01.22:



Backlog

- Sprint 4 | Nice To Have — Junit Activity
- Sprint 6 | Client | Nice To Have — Pattern Command adding
- + Add a card

To Do

- + Add a card

In Progress

- + Add a card

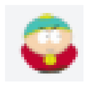
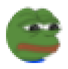
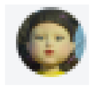
Done

- Sprint 6 — Add Votes Activity to API — Jan 4
- Sprint 6 | Client | Bonus — Buid Frontend — Jan 5
- Sprint 6 — Sprint summary — Jan 5
- Sprint 6 — Add Listen to Podcast Activity to API — Jan 4
- Client | Final Project — Client scheme — 2 — Jan 5
- Sprint 6 — Fixing Sprint 5 Comments — Jan 4
- SOW — SOW — 1 — Jan 4, 2021 — 1
- Sprint 6 — Technologies attache — 2 — Jan 4
- Sprint 6 | DB — Buid DB — Jan 4
- DB — DB Documentation — 1 — Jan 4
- SOW — 3.1 Use Case Diagram — 2 — Jan 4, 2021
- SOW — 3.2 Use Case Details — 4 — Jan 4, 2021
- Sprint 6 — Check activity controller — Jan 4
- Client | Nice To Have — Create project logo/banner — Jan 4
- + Add a card

Project in Integrative Software Engineering.
Sprint-6.

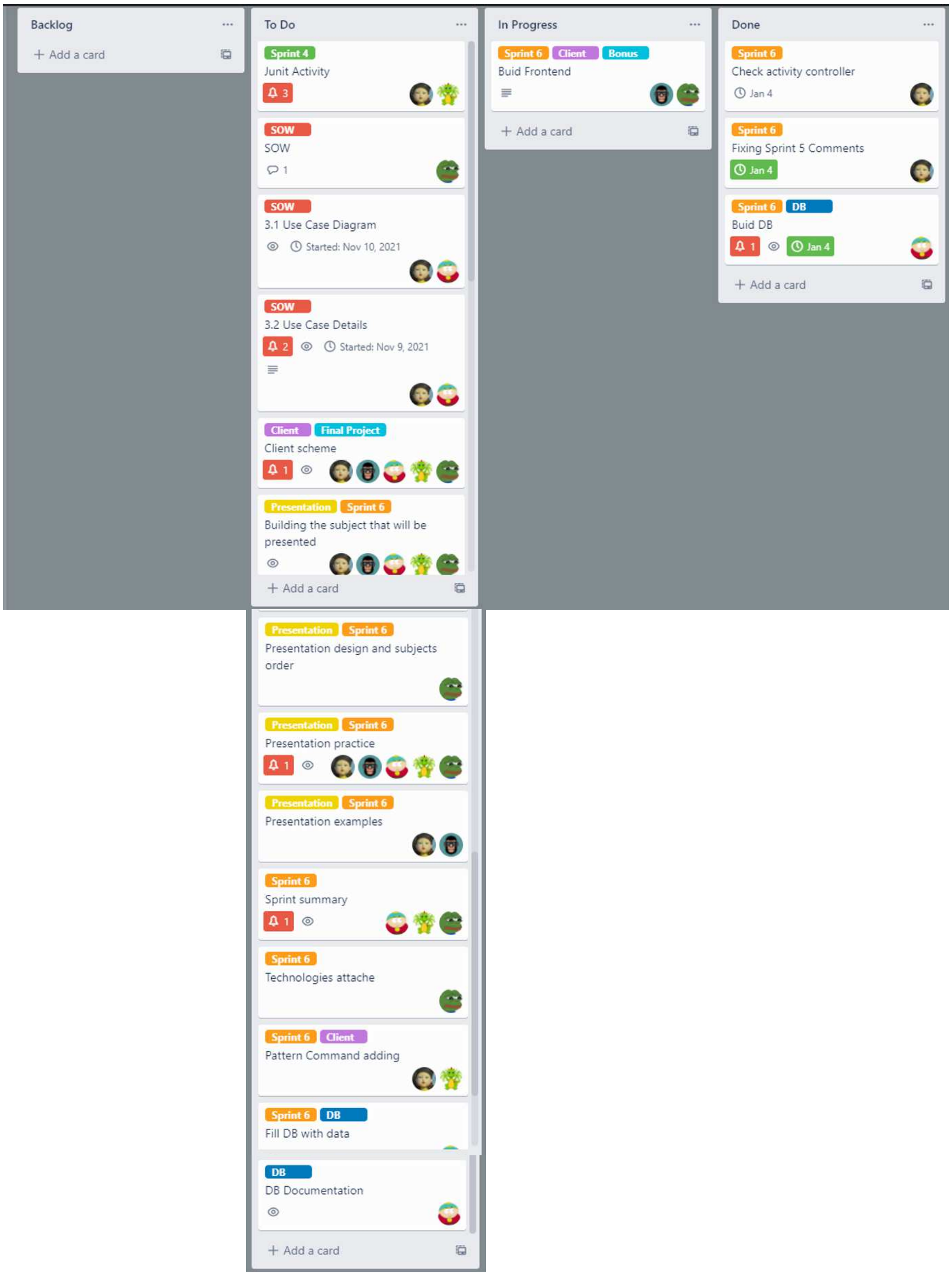
Date of submission: 05/01/22.
Course name: Integrative Software Engineering.
Course code: 10143.
Lecturer: Mr. Eisenstein Eyal.
Presenters: Lior Mitrany, Omer Ratsaby, Daniel Haimov, Shlomi Dori, Stav Rabinovich

List of Students:

Name	ID	Role	Avatar
Omer Ratsaby	312274780	DBA Team	
Stav Rabinovich	208661090	Scrum Master UI/UX Designer Team	
Lior Mitrany	205478258	Team Leader QA Engineer Team	
Daniel Haimov	207949058	Technical Writer Devops Team	
Shlomi Dori	316584044	Product Owner Team	

Kanban Board:

Start of the sprint - 22.12.21 :



Backlog

+ Add a card

To Do

- Sprint 4 — Junit Activity — 3
- SOW — SOW — 1
- SOW — 3.1 Use Case Diagram — Started: Nov 10, 2021
- SOW — 3.2 Use Case Details — 2 — Started: Nov 9, 2021
- Client, Final Project — Client scheme — 1
- Presentation, Sprint 6 — Building the subject that will be presented
- Presentation, Sprint 6 — Presentation design and subjects order
- Presentation, Sprint 6 — Presentation practice — 1
- Presentation, Sprint 6 — Presentation examples
- Sprint 6 — Sprint summary — 1
- Sprint 6 — Technologies attache
- Sprint 6, Client — Pattern Command adding
- Sprint 6, DB — Fill DB with data
- DB — DB Documentation

+ Add a card

In Progress

- Sprint 6, Client, Bonus — Buid Frontend

+ Add a card

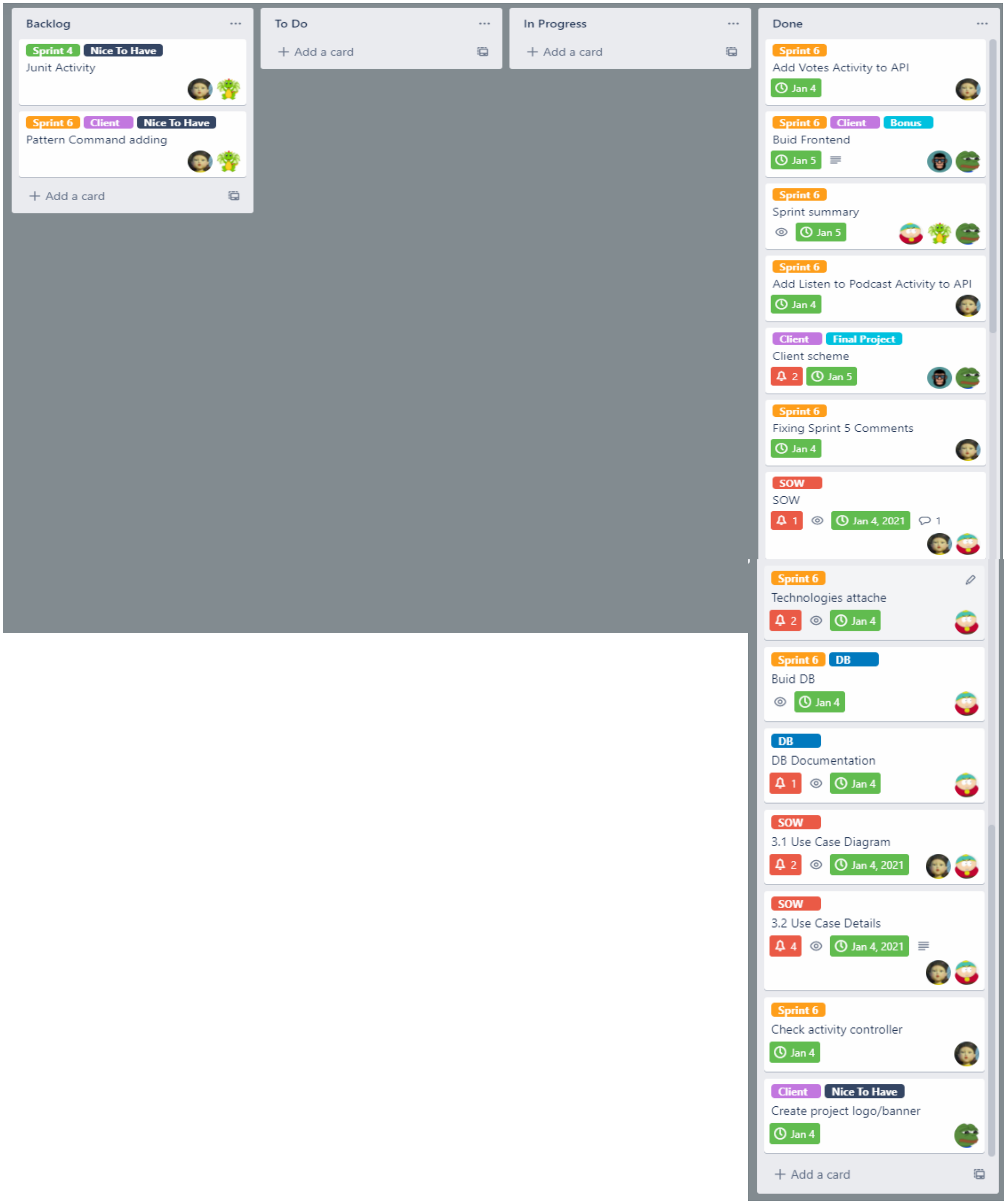
Done

- Sprint 6 — Check activity controller — Jan 4
- Sprint 6 — Fixing Sprint 5 Comments — Jan 4
- Sprint 6, DB — Buid DB — 1 — Jan 4

+ Add a card

Kanban Board:

End of the sprint - 04.01.22:



Backlog

Sprint 4 | Nice To Have
Junit Activity

Sprint 6 | Client | Nice To Have
Pattern Command adding

+ Add a card

To Do

+ Add a card

In Progress

+ Add a card

Done

Sprint 6
Add Votes Activity to API
Jan 4

Sprint 6 | Client | Bonus
Buid Frontend
Jan 5

Sprint 6
Sprint summary
Jan 5

Sprint 6
Add Listen to Podcast Activity to API
Jan 4

Client | Final Project
Client scheme
2 | Jan 5

Sprint 6
Fixing Sprint 5 Comments
Jan 4

SOW
SOW
1 | Jan 4, 2021 | 1

Sprint 6
Technologies attache
2 | Jan 4

Sprint 6 | DB
Buid DB
Jan 4

DB
DB Documentation
1 | Jan 4

SOW
3.1 Use Case Diagram
2 | Jan 4, 2021

SOW
3.2 Use Case Details
4 | Jan 4, 2021

Sprint 6
Check activity controller
Jan 4

Client | Nice To Have
Create project logo/banner
Jan 4

+ Add a card

General Summary of Work:

What went well for the team and should be continued on the next phases of work?
-- We decided to continue with dividing the last sprint's corrections and new sprint's missions into tasks and sectioned each task to both couples individuals and individuals, based on their knowledge, the size of the task, and by the chemistry of the team members.

In this sprint, we met almost every day in order to get the best results. The members of the team Were all determined to do their best and get the project done.

What problems did the team encounter through this phase of work
-- The absence of our team leader (Army's duty), had a little impact on the communication and And the usual work schedules (although he did his best to contribute).
As the semester goes on and each course that we participate in requires more time and effort, and in addition to that, some of our team members are going through work appliance processes, we've found it much more challenging to schedule appointments.
We have overcome the mentioned challenges by defining more accurately the tasks on the Kanban board.

Why did we not complete all planned work
-- We did.

<u>**Client Cheat Sheet**</u>

```
Create User:
        path = "/iob/users",
        method = POST.
        Body = {
                        "email":"user@demo.com",
                        "role":"ROLE",
                        "username":"Demo User",
                        "avatar":"XXX"
                }
        response =
                {
                        "userId":{
                                "domain":"2022a.demo",
                                "email":"user@demo.com"
                        },
                        "role":"ROLE",
                        "username":"Demo User",
                        "avatar":"XXX"
                }
        .then(response):
                path = "/iob/instances/{userDomain}/{userEmail}"
                method = POST
                Body = {
                                "type":"USER",
                                "name":"{userEmail}",
                                "active":true,
                                "createdBy":{
                                        "userId":{
                                                "domain":"{userDomain}",
                                                "email":"{userEmail}"
                                        }
                                },
                                "instanceAttributes":{
                                        "Genre": ["Genre0","Genre1"]
                                }
                        }
```

Login User:

```
path = "/iob/users/login/{userDomain}/{userEmail}",
method = GET
response =
{
	"userId":{
		"domain":"2022a.demo",
		"email":"user@demo.com"
	},
	"role":"PLAYER",
	"username":"Demo User",
	"avatar":"J"
}
```

Create Podcast:

```
        path = "/iob/instances/{userDomain}/{userEmail}"
        method = POST
        Body = {
                        "type":"PODCAST",
                        "name":"{PodcastName}",
                        "active":true,
                        "createdBy":{
                                "userId":{
                                        "domain":"{userDomain}",
                                        "email":"{userEmail}"
                                }
                        },
                        "instanceAttributes":{
                                "Author":"XXX",
                                “Genre”: [“Genre0”, “Genre1”],
                                "URL" : "http://"
                        }
                }
        response =
                instanceBoundary
        .then(response):
                path = "/iob/activities"
                method = POST
                Body = {
                                "type":"PODCAST",
                                "instance":{
                                        "instanceId":{
                                                "domain":"{podcastDomain}",
                                                "id":"{podcastId}"
                                        }
                                },
                                "invokedBy":{
                                        "userId":{
                                                "domain":"{appName}",
                                                "email":"{ONLY - PLAYER email}"
                                        }
                                },
                                "activityAttributes":{
                                        "upvotes":0,
                                        "downvotes":0,
                                        "listeners":0
                                }
                        }
```

Vote Podcast:

```
        path = "/iob/activities/vote"
        method = POST
        Body = {
                        "activityId":{
                                "domain":"{appName}",
                                "id":"{activityId}"
                        },
                        "instance":{
                                "instanceId":{
                                        "domain":"{appName}",
                                        "id":"{podcastId}"
                                }
                        },
                        "activityAttributes":{
                                "command":”upvotes” or “downvotes”
                        }
                }
```

Listen Podcast:

```
        path = "/iob/activities/listen"
        method = POST
        Body = {
                        "activityId":{
                                "domain":"{appName}",
                                "id":"{activityId}"
                        },
                        "instance":{
                                "instanceId":{
                                        "domain":"{appName}",
                                        "id":"{podcastId}"
                                }
                        },
                        "activityAttributes":{
                                "command":”listeners””
                        }
                }
```

Get X Podcasts from page Y:

```
path="/iob/instances/{userDomain}/{userEmail}/search/byType/PODCAST?size={X}&
page={Y}"
method = GET
```