# Sprint Reports

**Table of content**

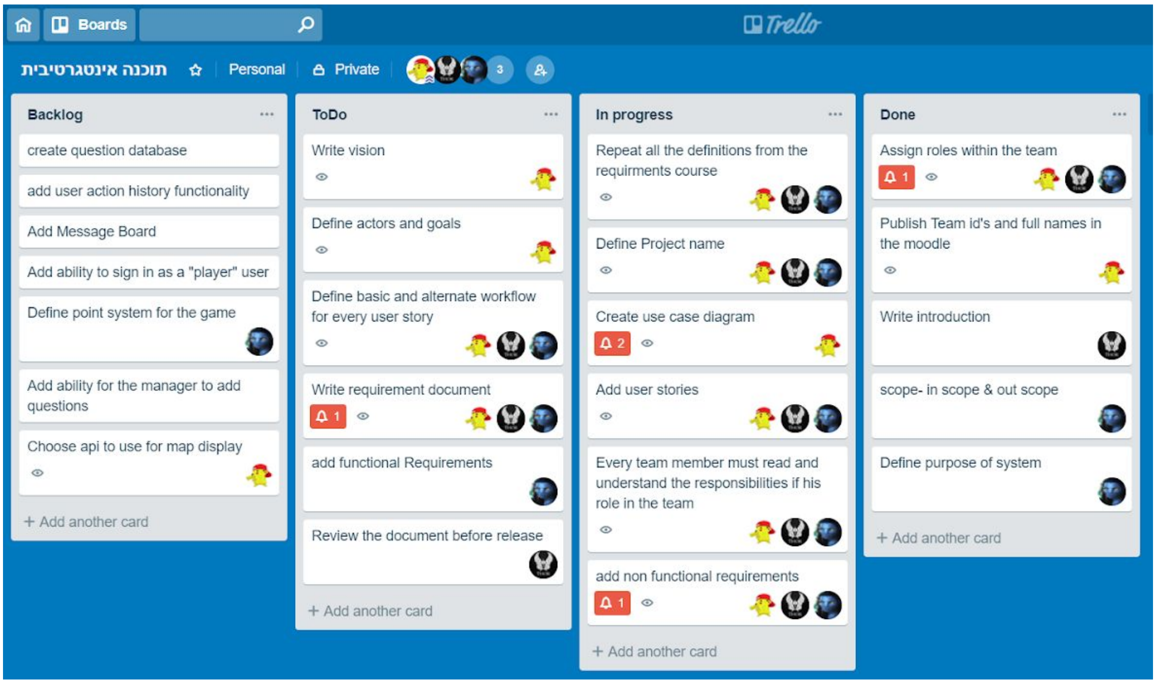
# Sprint Report #1

# (14.10.18 - 28.10.18)

**Topic of the Sprint:** Requirements document

**Link to scrum board:** trello board

**Scrum board at the beginning of the sprint (14.10.18):**



**Backlog**
- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "player" user
- Define point system for the game
- Add ability for the manager to add questions
- Choose api to use for map display
- + Add another card

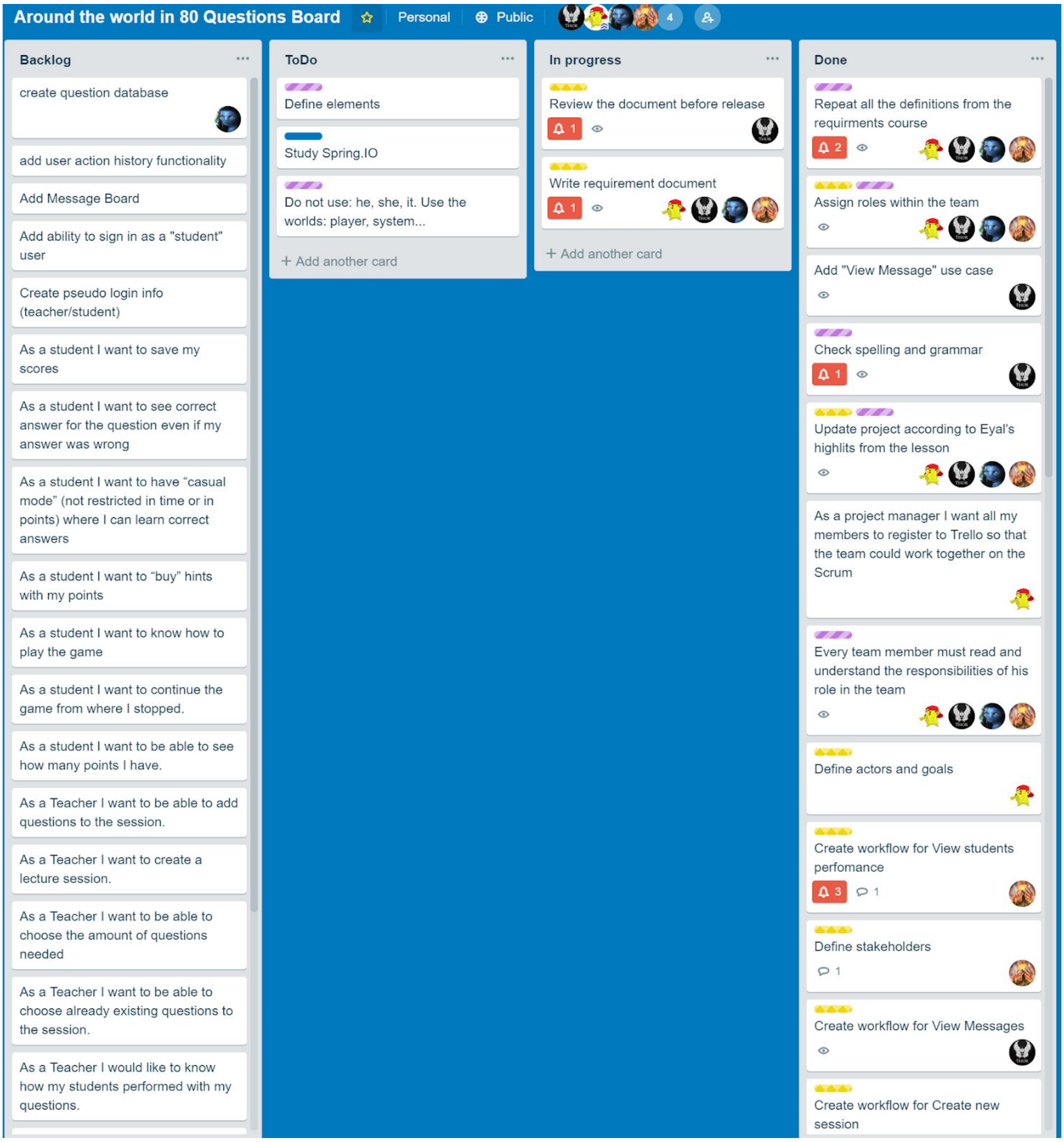
**ToDo**
- Write vision
- Define actors and goals
- Define basic and alternate workflow for every user story
- Write requirement document
- add functional Requirements
- Review the document before release
- + Add another card

**In progress**
- Repeat all the definitions from the requirments course
- Define Project name
- Create use case diagram
- Add user stories
- Every team member must read and understand the responsibilities if his role in the team
- add non functional requirements
- + Add another card

**Done**
- Assign roles within the team
- Publish Team id's and full names in the moodle
- Write introduction
- scope- in scope & out scope
- Define purpose of system
- + Add another card

**Scrum board at the end of sprint(28.10.2018):**

Part 1:



**Backlog**

- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have “casual mode” (not restricted in time or in points) where I can learn correct answers
- As a student I want to “buy” hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.

**ToDo**

- Define elements
- Study Spring.IO
- Do not use: he, she, it. Use the worlds: player, system...

+ Add another card

**In progress**

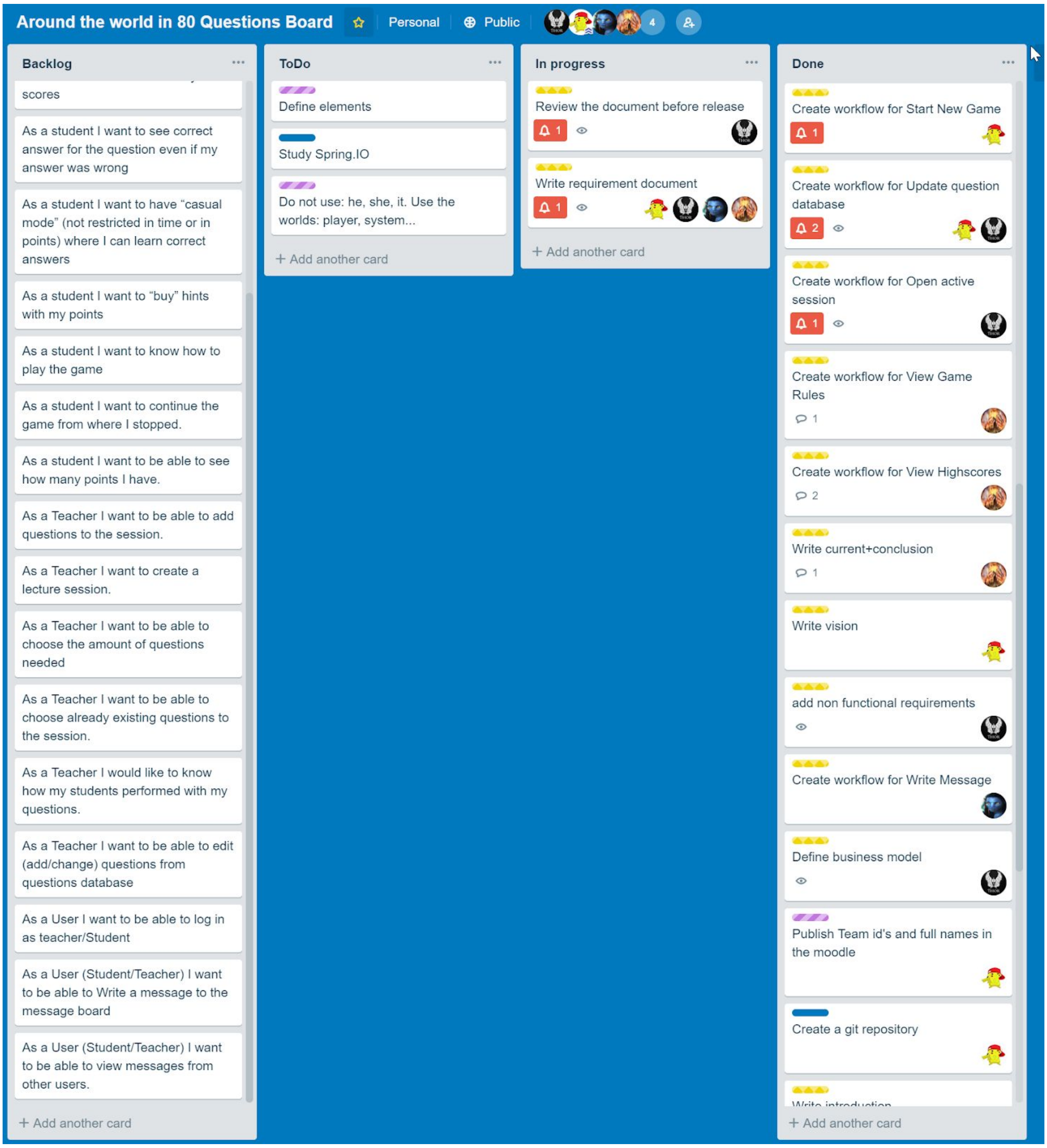
- Review the document before release 🔔 1
- Write requirement document 🔔 1

+ Add another card

**Done**

- Repeat all the definitions from the requirments course 🔔 2
- Assign roles within the team
- Add "View Message" use case
- Check spelling and grammar 🔔 1
- Update project according to Eyal's highlits from the lesson
- As a project manager I want all my members to register to Trello so that the team could work together on the Scrum
- Every team member must read and understand the responsibilities of his role in the team
- Define actors and goals
- Create workflow for View students perfomance 🔔 3 💬 1
- Define stakeholders 💬 1
- Create workflow for View Messages
- Create workflow for Create new session

## Part 2:



Around the world in 80 Questions Board ☆ Personal Public 4

**Backlog**

- scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have “casual mode” (not restricted in time or in points) where I can learn correct answers
- As a student I want to “buy” hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.
- As a Teacher I want to be able to edit (add/change) questions from questions database
- As a User I want to be able to log in as teacher/Student
- As a User (Student/Teacher) I want to be able to Write a message to the message board
- As a User (Student/Teacher) I want to be able to view messages from other users.

+ Add another card

**ToDo**

- Define elements
- Study Spring.IO
- Do not use: he, she, it. Use the worlds: player, system...

+ Add another card

**In progress**

- Review the document before release 🔔 1
- Write requirement document 🔔 1

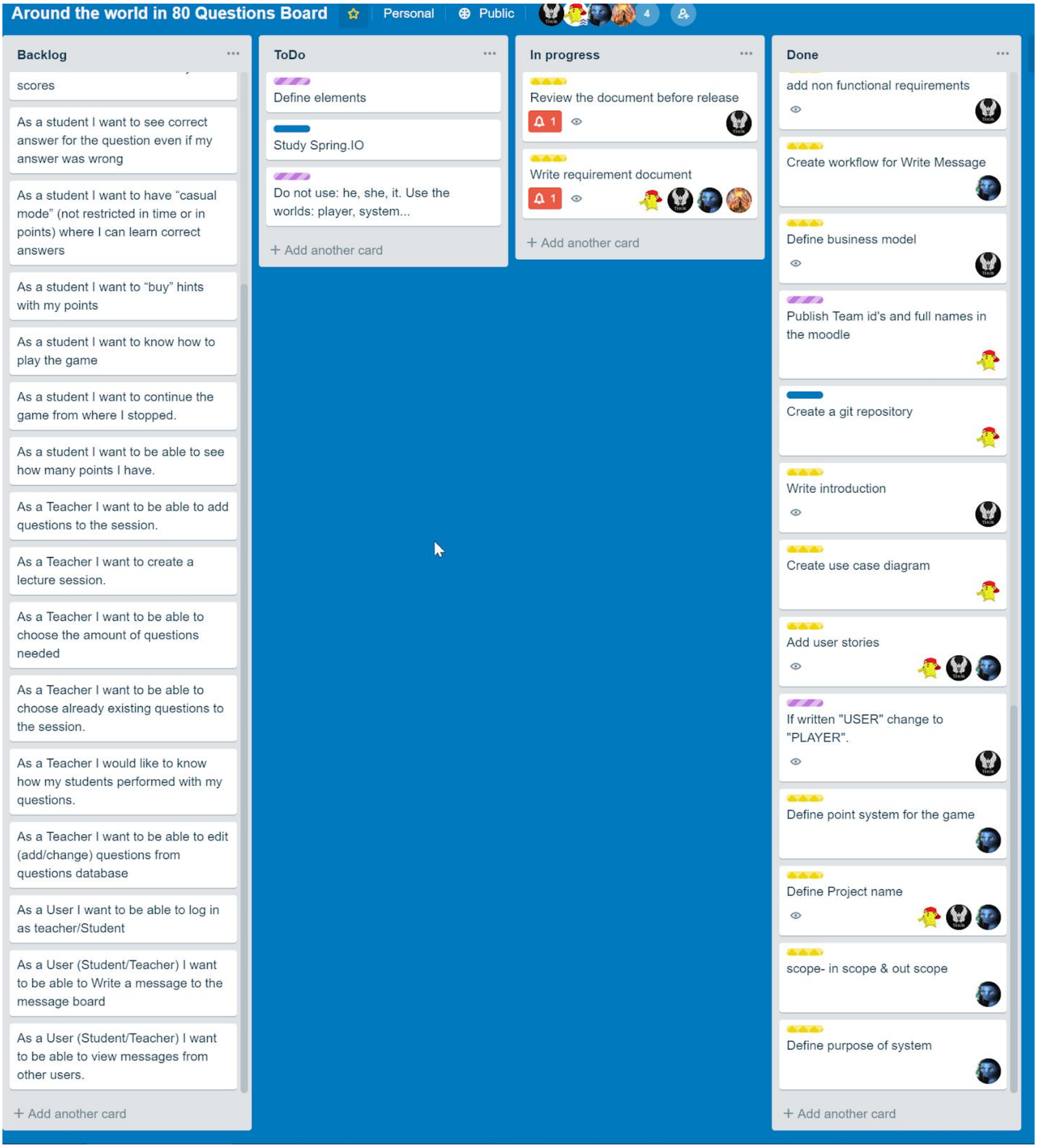
+ Add another card

**Done**

- Create workflow for Start New Game 🔔 1
- Create workflow for Update question database 🔔 2
- Create workflow for Open active session 🔔 1
- Create workflow for View Game Rules 💬 1
- Create workflow for View Highscores 💬 2
- Write current+conclusion 💬 1
- Write vision
- add non functional requirements
- Create workflow for Write Message
- Define business model
- Publish Team id's and full names in the moodle
- Create a git repository
- Write introduction

+ Add another card

Part 3:



**Backlog**

- scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have “casual mode” (not restricted in time or in points) where I can learn correct answers
- As a student I want to “buy” hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.
- As a Teacher I want to be able to edit (add/change) questions from questions database
- As a User I want to be able to log in as teacher/Student
- As a User (Student/Teacher) I want to be able to Write a message to the message board
- As a User (Student/Teacher) I want to be able to view messages from other users.

**ToDo**

- Define elements
- Study Spring.IO
- Do not use: he, she, it. Use the worlds: player, system...

**In progress**

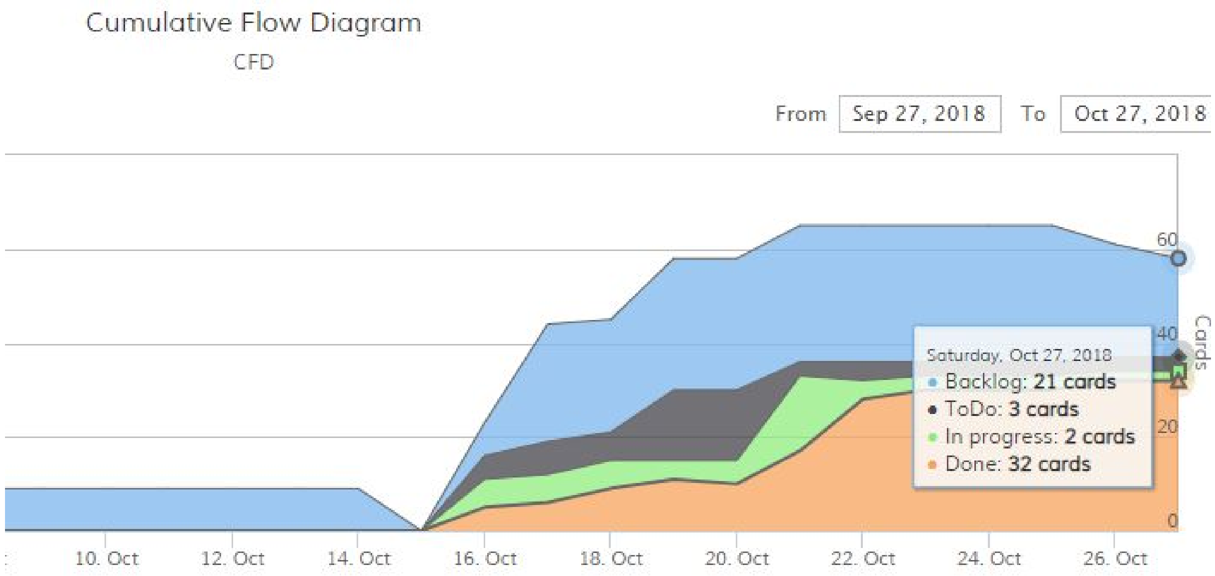
- Review the document before release
- Write requirement document

**Done**

- add non functional requirements
- Create workflow for Write Message
- Define business model
- Publish Team id's and full names in the moodle
- Create a git repository
- Write introduction
- Create use case diagram
- Add user stories
- If written "USER" change to "PLAYER".
- Define point system for the game
- Define Project name
- scope- in scope & out scope
- Define purpose of system

**Link to the Trello Board:**

trello.com/b/6rYbPb41



**Colors meaning in trello board:**

QA

Programming

Requirements document, Paperwork

moodle, general

**List of Students:**

1) Daniil Rolnik . ID: 334018009.

   Roles in the team: Product Manager, Product Owner, DevOps.

   Avatar:

   Github account: danrol

2) Elia Ben Anat . ID: 308048388.

   Role in the team: Database Administrator.

   Avatar:

   Github account: eliaba

3) Eden Dupont . ID: 204808596.

   Role in the team: QA Engineer.

   Avatar:

   Github account: eg7eg7

4) Eden Sharoni . ID: 315371906

Role in the team: Scrum Master.

Avatar:

Github account: EdenSharoni

**General summary of work:**

First meetings as a team. Topic was chosen and requirements document finished before deadline.

**What went well throughout the sprint?**

The team worked hard and achieved the main goal of finishing the document in time .

**What should be improved in team work?**

The communication between team members and time management.

**What  problems did the team encounter through the sprint?**

The main problem we had to face was the change of the application concept after the second lecture, that changed some of the requirements.

**Why did we not complete all planned work?**

We achieved all of our goals in this sprint.

**What is expected for the next sprint?**

We are planning on starting to build the first versions of our application and to perform the first tests of it . Every member of the team will study Spring.IO

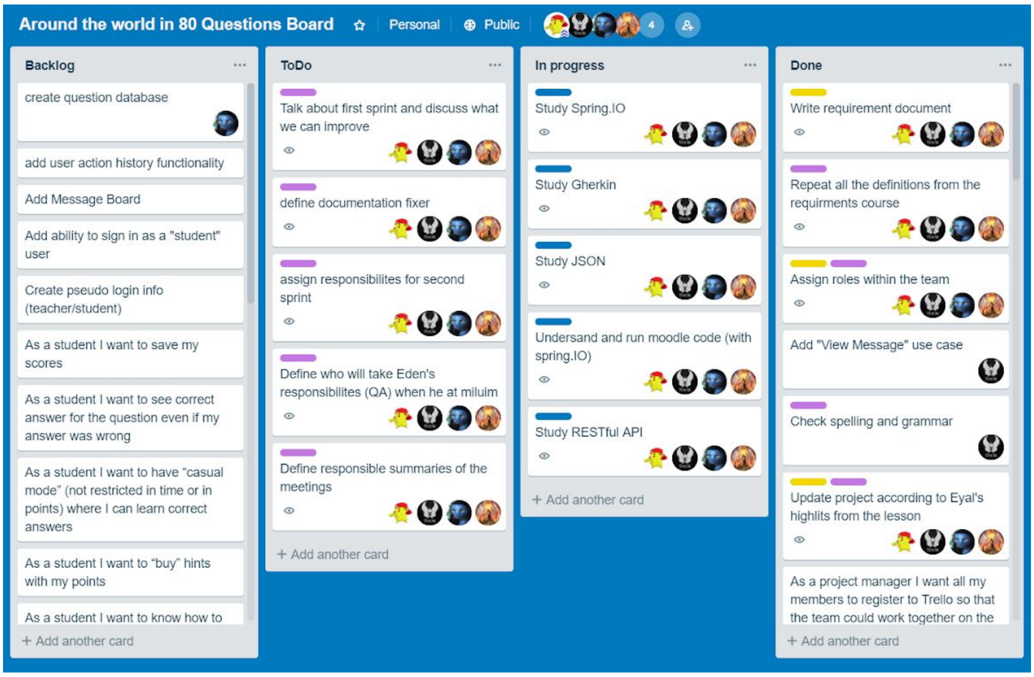
# Sprint Report #2

# (28.10.18 - 11.11.18)

**Topic of the Sprint:** Implement urls with REST api

**Link to scrum board:** trello board

### Scrum board at the beginning of the sprint (28.10.18):



Around the world in 80 Questions Board ☆ Personal Public 4

**Backlog**

- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to

+ Add another card

**ToDo**

- Talk about first sprint and discuss what we can improve
- define documentation fixer
- assign responsibilites for second sprint
- Define who will take Eden's responsibilites (QA) when he at miluim
- Define responsible summaries of the meetings

+ Add another card

**In progress**

- Study Spring.IO
- Study Gherkin
- Study JSON
- Undersand and run moodle code (with spring.IO)
- Study RESTful API

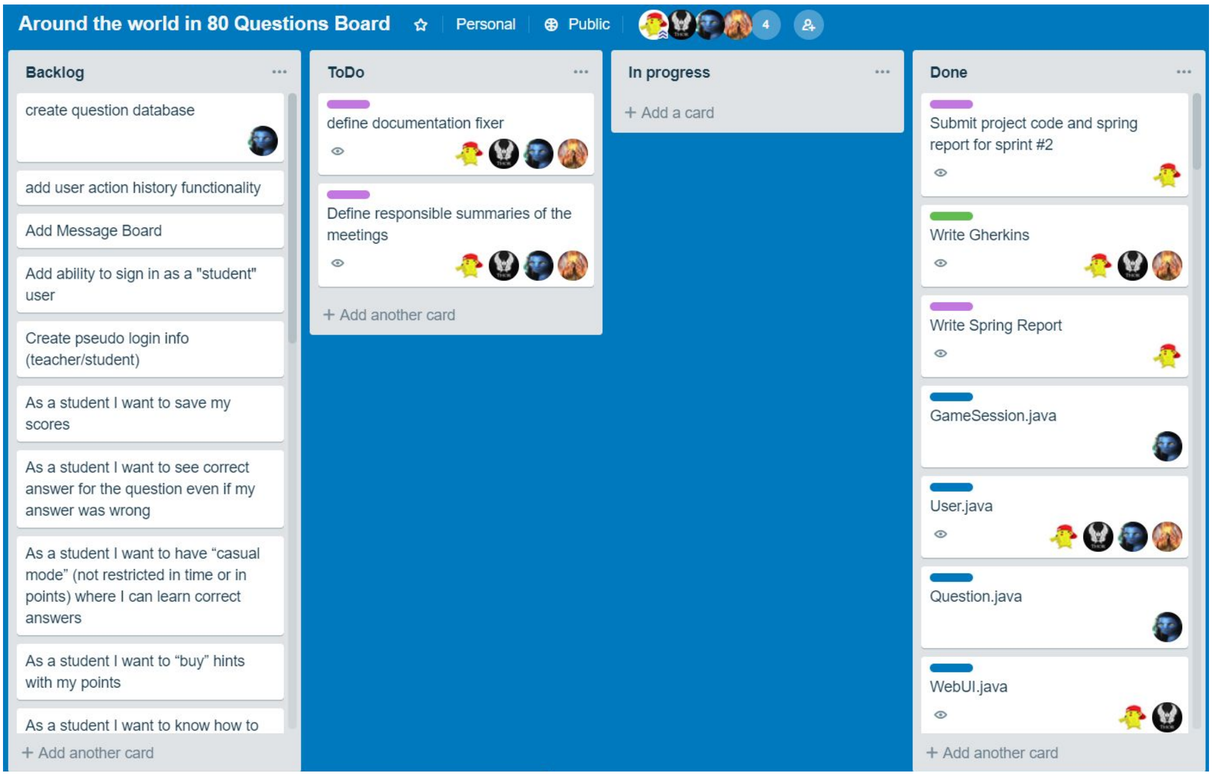
+ Add another card

**Done**

- Write requirement document
- Repeat all the definitions from the requirments course
- Assign roles within the team
- Add "View Message" use case
- Check spelling and grammar
- Update project according to Eyal's highlits from the lesson
- As a project manager I want all my members to register to Trello so that the team could work together on the

+ Add another card

### Scrum board at the end of the sprint (11.11.2018):



Around the world in 80 Questions Board ☆ Personal Public 4

**Backlog**

- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to

+ Add another card

**ToDo**

- define documentation fixer
- Define responsible summaries of the meetings

+ Add another card

**In progress**

+ Add a card

**Done**

- Submit project code and spring report for sprint #2
- Write Gherkins
- Write Spring Report
- GameSession.java
- User.java
- Question.java
- WebUI.java

+ Add another card

### General summary of work:

Project prototype. 4 meetings was performed. During week #2 of sprint #2 Eden Dupont was at miluim so his responsibilities was divided between another team members.

**What went well throughout the sprint?**

Working code submitted before deadline. Many meetings performed which helped in discussion of project and project's problems. All team members understand how Spring.io and Rest api work and how it connects to our project.

**What should be improved in team work?**

The communication between team members and time management.

**What  problems did the team encounter through the sprint?**

Team still lacks basic understanding of some requirements given by lecturer (project's client). Communication with client should be improved. Some urls wasn't implemented because team lacks understanding of desired result.

**Why did we not complete all planned work?**

Lack in understanding of some basic requirements from client.

**What is expected for the next sprint?**

Some basic project's code definitions should be changed to improve integrative side of our project. Talk with client and gain lacking knowledge.  Meet with client as team outside from lecture and get deeper understanding of project requirements. Get and perform  tasks before sprint #3 ends.

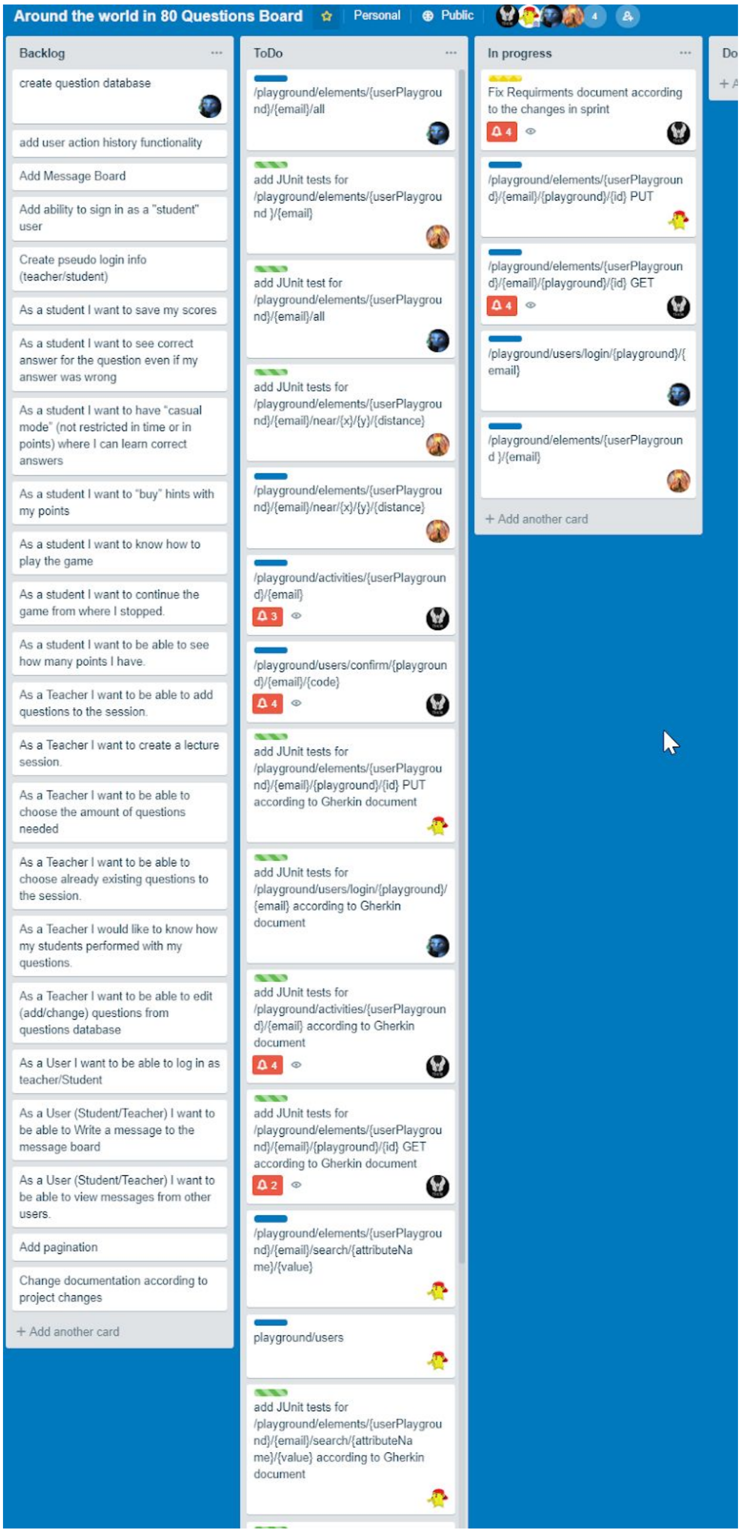
# Sprint Report #3

# (11.11.18 - 25.11.18)

**Topic of the Sprint:** Automated tests and stub services

## Link to scrum board: trello board

## Scrum board at the beginning of sprint(11.11.2018:



**Backlog**

- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.
- As a Teacher I want to be able to edit (add/change) questions from questions database
- As a User I want to be able to log in as teacher/Student
- As a User (Student/Teacher) I want to be able to Write a message to the message board
- As a User (Student/Teacher) I want to be able to view messages from other users.
- Add pagination
- Change documentation according to project changes

+ Add another card

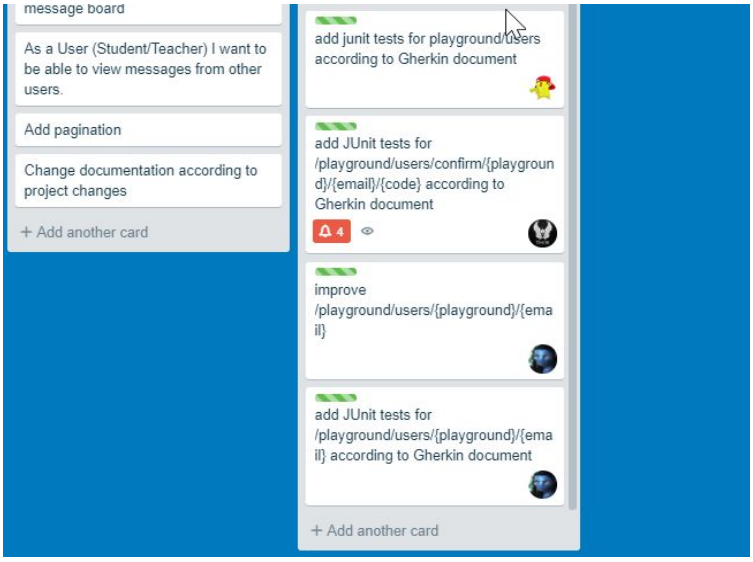
**ToDo**

- /playground/elements/{userPlayground}/{email}/all
- add JUnit tests for /playground/elements/{userPlayground }/{email}
- add JUnit test for /playground/elements/{userPlayground}/{email}/all
- add JUnit tests for /playground/elements/{userPlayground}/{email}/near/{x}/{y}/{distance}
- /playground/elements/{userPlayground}/{email}/near/{x}/{y}/{distance}
- /playground/activities/{userPlayground}/{email} 3
- /playground/users/confirm/{playground}/{email}/{code} 4
- add JUnit tests for /playground/elements/{userPlayground}/{email}/{playground}/{id} PUT according to Gherkin document
- add JUnit tests for /playground/users/login/{playground}/{email} according to Gherkin document
- add JUnit tests for /playground/activities/{userPlayground}/{email} according to Gherkin document 4
- add JUnit tests for /playground/elements/{userPlayground}/{email}/{playground}/{id} GET according to Gherkin document 2
- /playground/elements/{userPlayground}/{email}/search/{attributeName}/{value}
- playground/users
- add JUnit tests for /playground/elements/{userPlayground}/{email}/search/{attributeName}/{value} according to Gherkin document

**In progress**

- Fix Requirments document according to the changes in sprint 4
- /playground/elements/{userPlayground}/{email}/{playground}/{id} PUT
- /playground/elements/{userPlayground}/{email}/{playground}/{id} GET 4
- /playground/users/login/{playground}/{email}
- /playground/elements/{userPlayground }/{email}

+ Add another card



message board

As a User (Student/Teacher) I want to be able to view messages from other users.

Add pagination

Change documentation according to project changes

+ Add another card

add junit tests for playground/users according to Gherkin document

add JUnit tests for /playground/users/confirm/{playground}/{email}/{code} according to Gherkin document

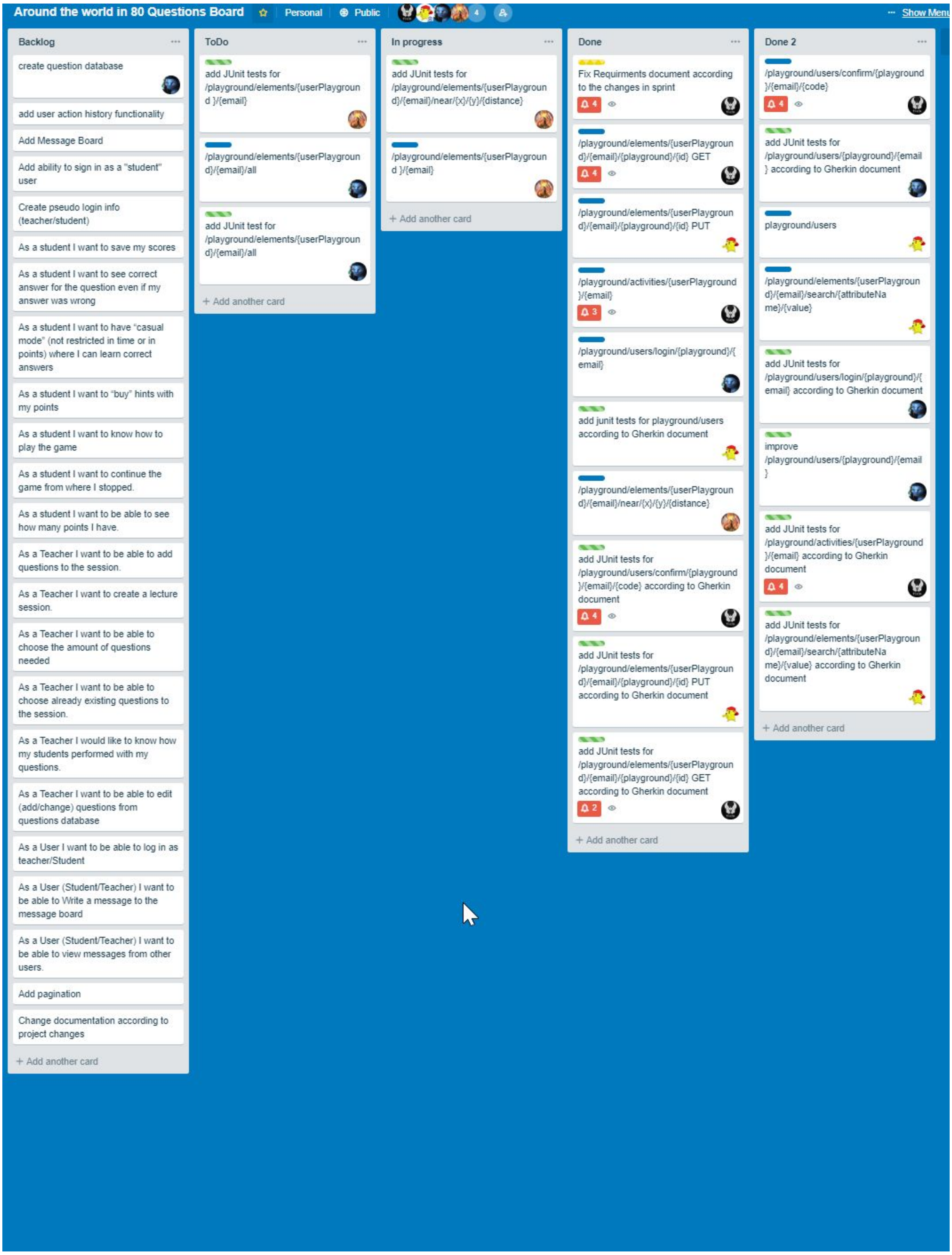
4

improve /playground/users/{playground}/{email}

add JUnit tests for /playground/users/{playground}/{email} according to Gherkin document

+ Add another card

## Scrum board at the end of sprint#3 (25.11.2018):



**Backlog**

- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.
- As a Teacher I want to be able to edit (add/change) questions from questions database
- As a User I want to be able to log in as teacher/Student
- As a User (Student/Teacher) I want to be able to Write a message to the message board
- As a User (Student/Teacher) I want to be able to view messages from other users.
- Add pagination
- Change documentation according to project changes

+ Add another card

**ToDo**

- add JUnit tests for /playground/elements/{userPlayground }/{email}
- /playground/elements/{userPlayground}/{email}/all
- add JUnit test for /playground/elements/{userPlayground}/{email}/all

+ Add another card

**In progress**

- add JUnit tests for /playground/elements/{userPlayground}/{email}/near/{x}/{y}/{distance}
- /playground/elements/{userPlayground }/{email}

+ Add another card

**Done**

- Fix Requirments document according to the changes in sprint 🔔 4
- /playground/elements/{userPlayground}/{email}/{playground}/{id} GET 🔔 4
- /playground/elements/{userPlayground}/{email}/{playground}/{id} PUT
- /playground/activities/{userPlayground}/{email} 🔔 3
- /playground/users/login/{playground}/{email}
- add junit tests for playground/users according to Gherkin document
- /playground/elements/{userPlayground}/{email}/near/{x}/{y}/{distance}
- add JUnit tests for /playground/users/confirm/{playground}/{email}/{code} according to Gherkin document 🔔 4
- add JUnit tests for /playground/elements/{userPlayground}/{email}/{playground}/{id} PUT according to Gherkin document
- add JUnit tests for /playground/elements/{userPlayground}/{email}/{playground}/{id} GET according to Gherkin document 🔔 2

+ Add another card

**Done 2**

- /playground/users/confirm/{playground}/{email}/{code} 🔔 4
- add JUnit tests for /playground/users/{playground}/{email} according to Gherkin document
- playground/users
- /playground/elements/{userPlayground}/{email}/search/{attributeName}/{value}
- add JUnit tests for /playground/users/login/{playground}/{email} according to Gherkin document
- improve /playground/users/{playground}/{email }
- add JUnit tests for /playground/activities/{userPlayground}/{email} according to Gherkin document 🔔 4
- add JUnit tests for /playground/elements/{userPlayground}/{email}/search/{attributeName}/{value} according to Gherkin document

+ Add another card

**List of Students:**

5) Daniil Rolnik . ID: 334018009.

   Roles in the team: Product Manager, Product Owner, DevOps.

   Avatar:

   Github account: danrol

6) Elia Ben Anat . ID: 308048388.

   Role in the team: Database Administrator.

   Avatar:

   Github account: eliaba

7) Eden Dupont . ID: 204808596.

   Role in the team: QA Engineer.

   Avatar:

   Github account: eg7eg7

8) Eden Sharoni . ID: 315371906

   Role in the team: Scrum Master.

   Avatar:

   Github account: EdenSharoni

**Link to Github repository:** https://github.com/danrol/ATW80Q

**General summary of work:**

Two meetings were performed. During those meetings we discussed:

- What needs to be done during sprint #3
- How to change our requirements to fit the API
- Division of labor. Every team member received what goals they need to achieve during sprint #3
- Web controller was improved. Logic added to url methods
- Basic JUnit tests were added, and tested for most API's
- Gherkin tests document improved

- Basic pagination using streams was added
- Temporary database through services was implemented

**What went well throughout the sprint?**

Team met with Eyal at the beginning of the sprint, this improved the team's understanding of the project which led to a more organized work and a more successful (and less stressful) sprint.

Working code submitted before deadline. Most needed gherkin test added to java code - still some are missing. Many meetings performed which helped in discussion of project and project's problems. All team members understand how Spring.io and Rest api work and how it connects to our project.

**What should be improved in team work?**

Some team members doing more work than others. Less hard working need to improve their time management and to become more active team members.

**What  problems did the team encounter through the sprint?**

Test that was built to check methods in controller returning ElementTO[] doesn't work. Team need to find out what change should to be done so tests will pass. Need to figure out how to make pagination work. Some problem with post methods consultation with Eyal needed. Team made bad decision about code division to classes.

**Why did we not complete all planned work?**

We need consultation from Eyal to solve some problems in tests

**What is expected for the next sprint?**

Temporary service database will be changed with constant solution (we'll get more details about this from lesson with Eyal). Requirements document need to be updated according to project changes.

## List of  used technologies:

1. Spring.io
2. RESTful api
3. jdk 1.8
4. JUnit
5. Eclipse IDE
6. Github repository

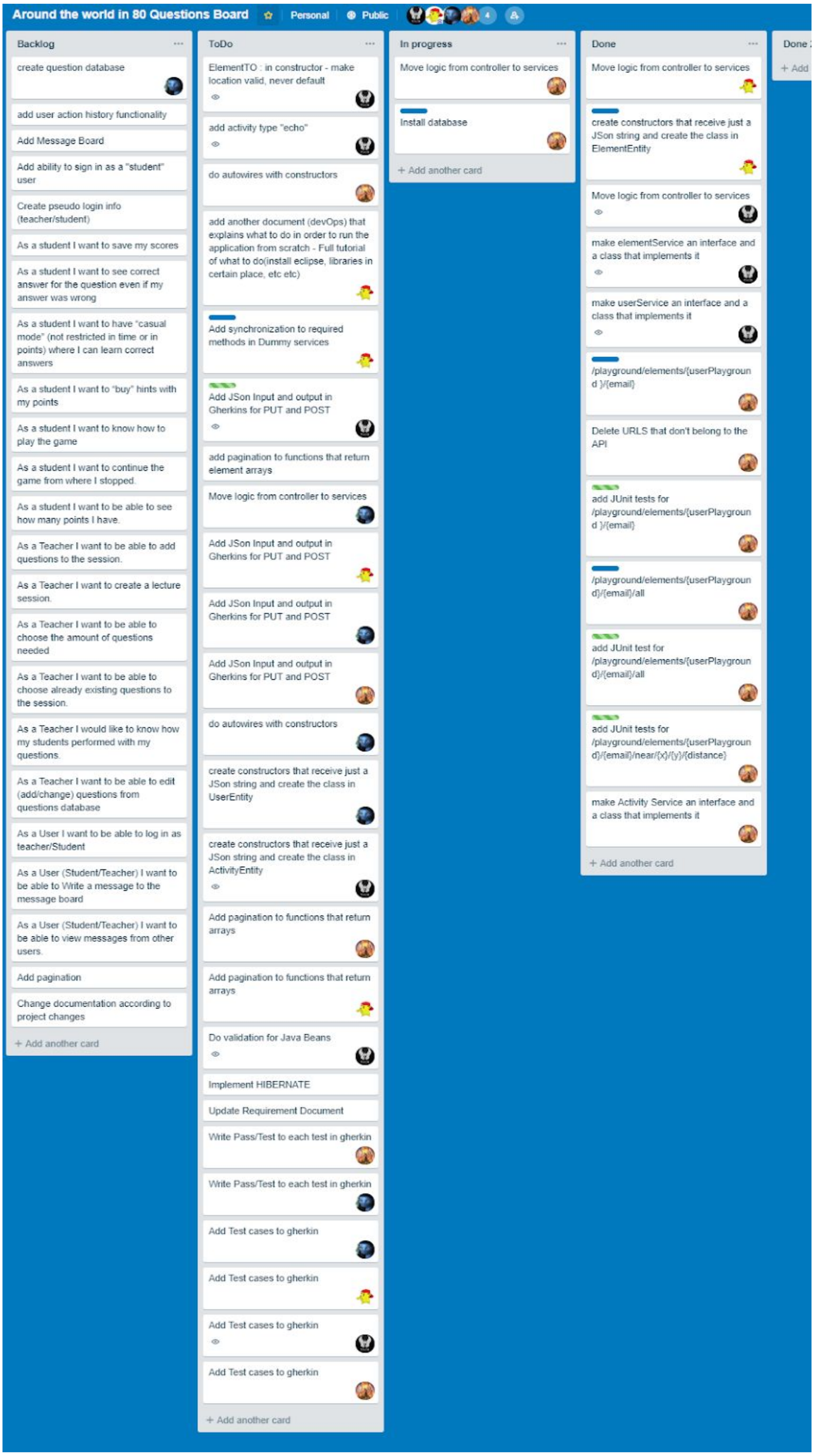
# Sprint Report #4

(25.11.18 - 16.12.18)

**Topics of the Sprint:** Move from stub services to JPA services

**Link to scrum board:** trello board

**Scrum board at the beginning of sprint (25.11.18):**



Around the world in 80 Questions Board

Personal · Public

**Backlog**

- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.
- As a Teacher I want to be able to edit (add/change) questions from questions database
- As a User I want to be able to log in as teacher/Student
- As a User (Student/Teacher) I want to be able to Write a message to the message board
- As a User (Student/Teacher) I want to be able to view messages from other users.
- Add pagination
- Change documentation according to project changes

+ Add another card

**ToDo**

- ElementTO : in constructor - make location valid, never default
- add activity type "echo"
- do autowires with constructors
- add another document (devOps) that explains what to do in order to run the application from scratch - Full tutorial of what to do(install eclipse, libraries in certain place, etc etc)
- Add synchronization to required methods in Dummy services
- Add JSon Input and output in Gherkins for PUT and POST
- add pagination to functions that return element arrays
- Move logic from controller to services
- Add JSon Input and output in Gherkins for PUT and POST
- Add JSon Input and output in Gherkins for PUT and POST
- Add JSon Input and output in Gherkins for PUT and POST
- do autowires with constructors
- create constructors that receive just a JSon string and create the class in UserEntity
- create constructors that receive just a JSon string and create the class in ActivityEntity
- Add pagination to functions that return arrays
- Add pagination to functions that return arrays
- Do validation for Java Beans
- Implement HIBERNATE
- Update Requirement Document
- Write Pass/Test to each test in gherkin
- Write Pass/Test to each test in gherkin
- Add Test cases to gherkin
- Add Test cases to gherkin
- Add Test cases to gherkin
- Add Test cases to gherkin

+ Add another card

**In progress**

- Move logic from controller to services
- Install database

+ Add another card

**Done**

- Move logic from controller to services
- create constructors that receive just a JSon string and create the class in ElementEntity
- Move logic from controller to services
- make elementService an interface and a class that implements it
- make userService an interface and a class that implements it
- /playground/elements/{userPlayground}/{email}
- Delete URLS that don't belong to the API
- add JUnit tests for /playground/elements/{userPlayground}/{email}
- /playground/elements/{userPlayground}/{email}/all
- add JUnit test for /playground/elements/{userPlayground}/{email}/all
- add JUnit tests for /playground/elements/{userPlayground}/{email}/near/{x}/{y}/{distance}
- make Activity Service an interface and a class that implements it

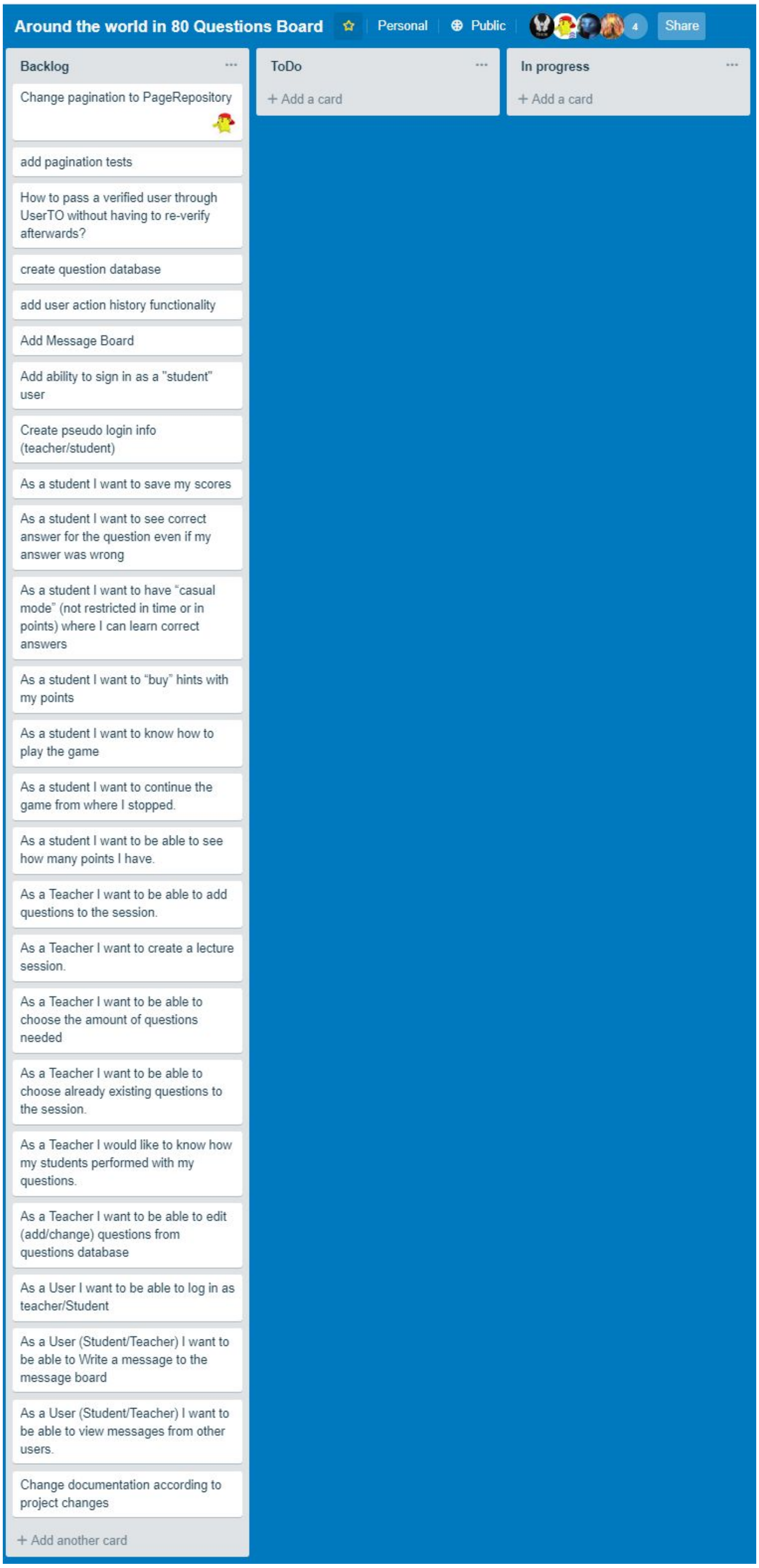
+ Add another card

**Done**

+ Add

## Scrum board at the end of sprint (16.12.18):

## Part 1:



Around the world in 80 Questions Board ☆ Personal Public 4 Share

**Backlog**

- Change pagination to PageRepository
- add pagination tests
- How to pass a verified user through UserTO without having to re-verify afterwards?
- create question database
- add user action history functionality
- Add Message Board
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.
- As a Teacher I want to be able to edit (add/change) questions from questions database
- As a User I want to be able to log in as teacher/Student
- As a User (Student/Teacher) I want to be able to Write a message to the message board
- As a User (Student/Teacher) I want to be able to view messages from other users.
- Change documentation according to project changes

+ Add another card

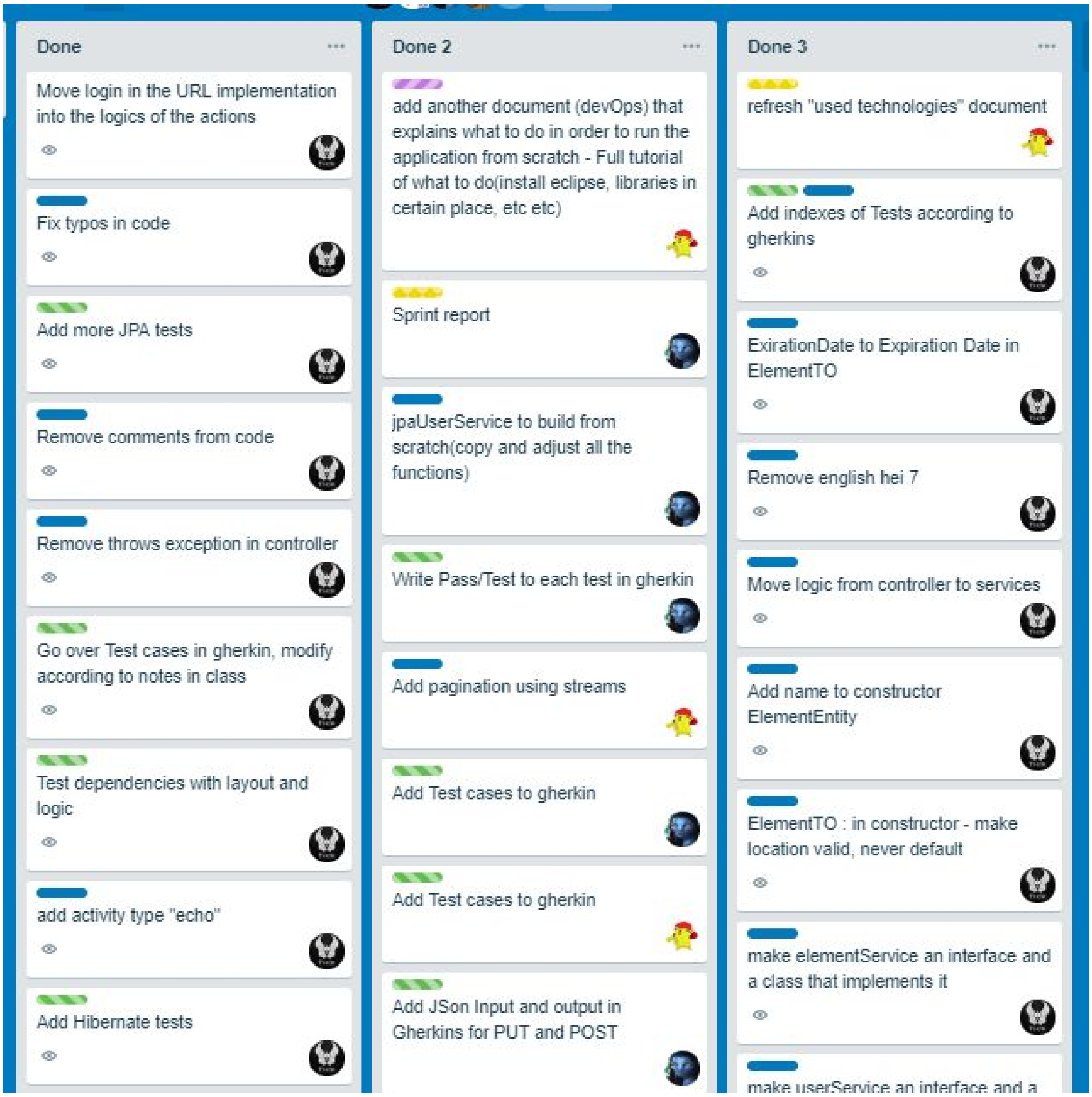
**ToDo**

+ Add a card

**In progress**

+ Add a card

**Part 2:**



Done

- Move login in the URL implementation into the logics of the actions
- Fix typos in code
- Add more JPA tests
- Remove comments from code
- Remove throws exception in controller
- Go over Test cases in gherkin, modify according to notes in class
- Test dependencies with layout and logic
- add activity type "echo"
- Add Hibernate tests

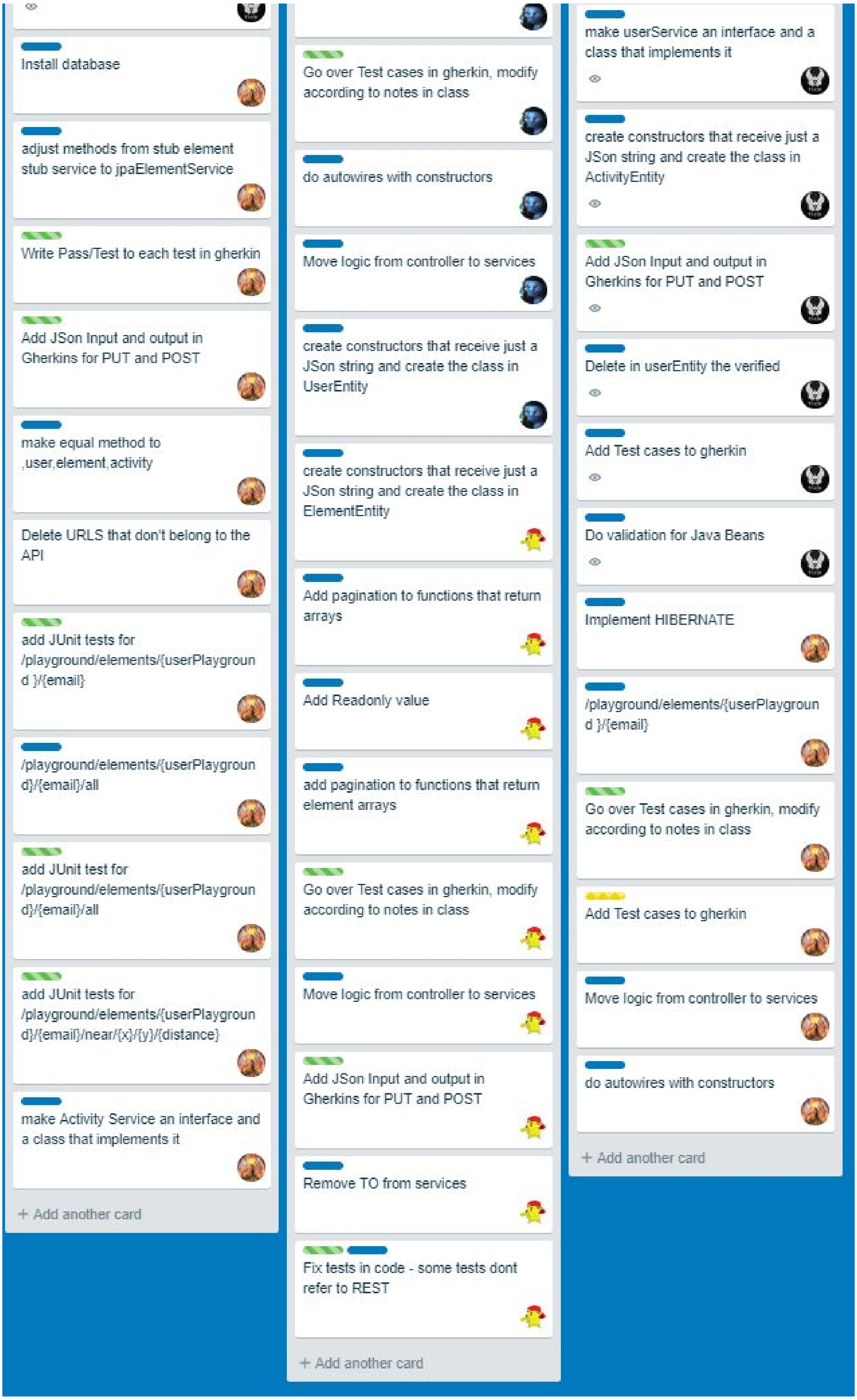
Done 2

- add another document (devOps) that explains what to do in order to run the application from scratch - Full tutorial of what to do(install eclipse, libraries in certain place, etc etc)
- Sprint report
- jpaUserService to build from scratch(copy and adjust all the functions)
- Write Pass/Test to each test in gherkin
- Add pagination using streams
- Add Test cases to gherkin
- Add Test cases to gherkin
- Add JSon Input and output in Gherkins for PUT and POST

Done 3

- refresh "used technologies" document
- Add indexes of Tests according to gherkins
- ExirationDate to Expiration Date in ElementTO
- Remove english hei 7
- Move logic from controller to services
- Add name to constructor ElementEntity
- ElementTO : in constructor - make location valid, never default
- make elementService an interface and a class that implements it
- make userService an interface and a

## Part 3:



Install database

adjust methods from stub element stub service to jpaElementService

Write Pass/Test to each test in gherkin

Add JSon Input and output in Gherkins for PUT and POST

make equal method to ,user,element,activity

Delete URLS that don't belong to the API

add JUnit tests for /playground/elements/{userPlayground }/{email}

/playground/elements/{userPlayground}/{email}/all

add JUnit test for /playground/elements/{userPlayground}/{email}/all

add JUnit tests for /playground/elements/{userPlayground}/{email}/near/{x}/{y}/{distance}

make Activity Service an interface and a class that implements it

+ Add another card

Go over Test cases in gherkin, modify according to notes in class

do autowires with constructors

Move logic from controller to services

create constructors that receive just a JSon string and create the class in UserEntity

create constructors that receive just a JSon string and create the class in ElementEntity

Add pagination to functions that return arrays

Add Readonly value

add pagination to functions that return element arrays

Go over Test cases in gherkin, modify according to notes in class

Move logic from controller to services

Add JSon Input and output in Gherkins for PUT and POST

Remove TO from services

Fix tests in code - some tests dont refer to REST

+ Add another card

make userService an interface and a class that implements it

create constructors that receive just a JSon string and create the class in ActivityEntity

Add JSon Input and output in Gherkins for PUT and POST

Delete in userEntity the verified

Add Test cases to gherkin

Do validation for Java Beans

Implement HIBERNATE

/playground/elements/{userPlayground }/{email}

Go over Test cases in gherkin, modify according to notes in class

Add Test cases to gherkin

Move logic from controller to services

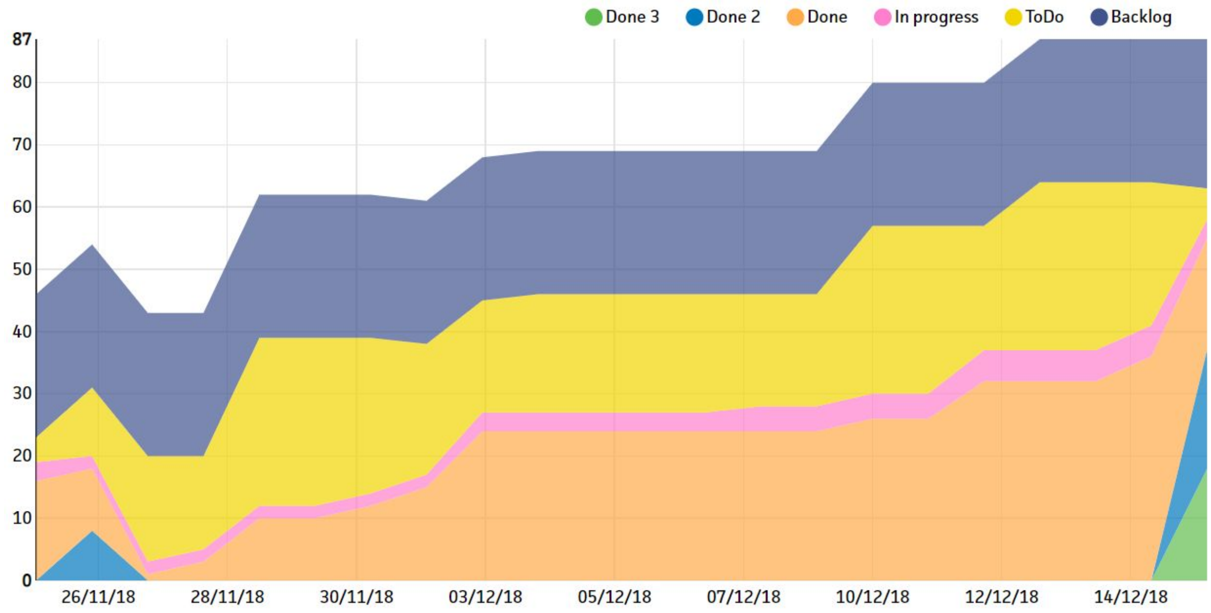
do autowires with constructors

+ Add another card

### Link to the Trello Board:

trello.com/b/6rYbPb41

### control flow diagram:



### Colors meaning in trello board:

QA

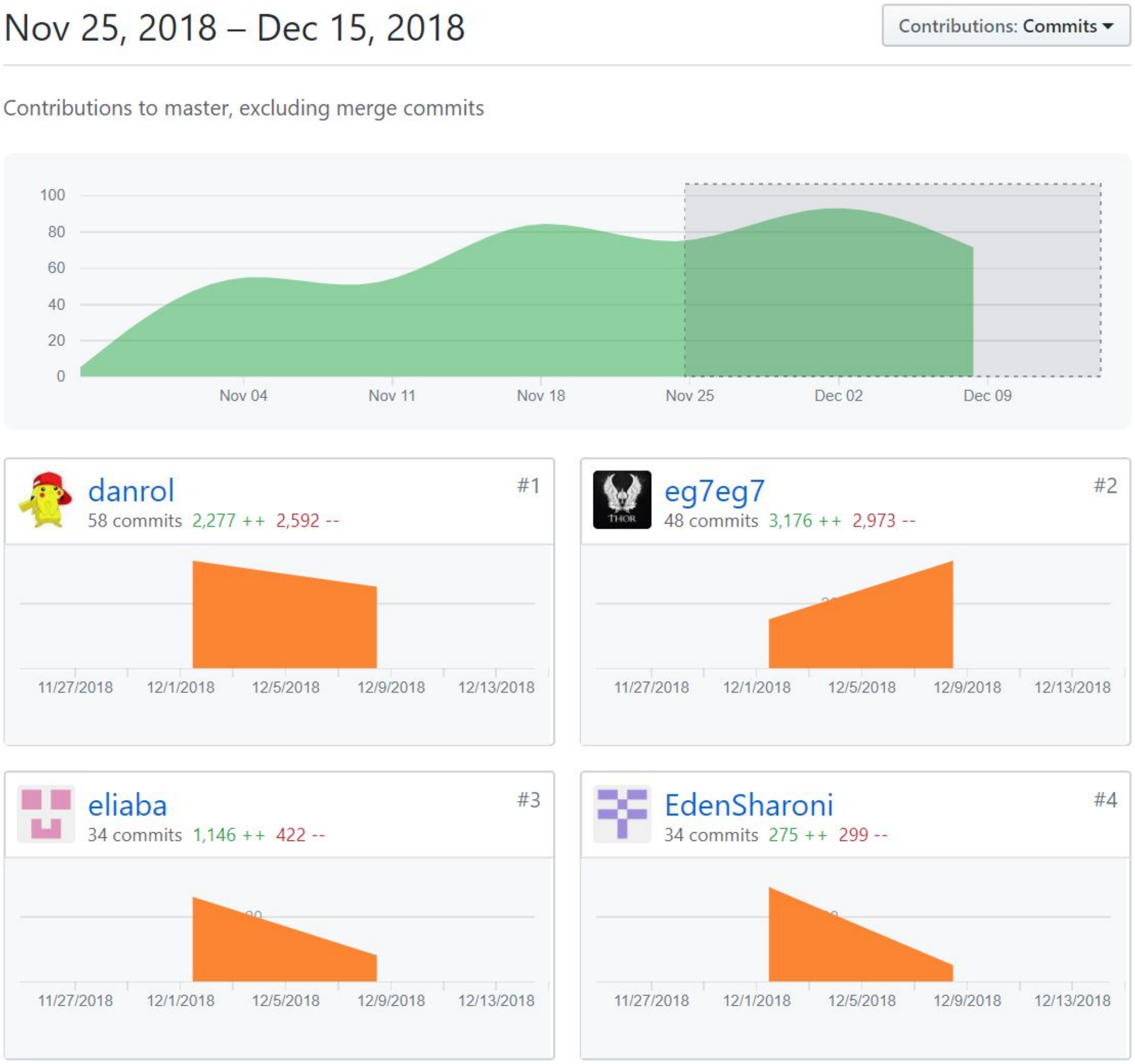
Programming

Requirements document, Paperwork

moodle, general

**Github Statistics:**

1. danrol - Daniel Rolnik
2. eg7eg7 - Eden Dupont
3. eliaba - Elia Ben Anat
4. EdenSharoni - Eden Sharoni



Nov 25, 2018 – Dec 15, 2018

Contributions: Commits ▾

Contributions to master, excluding merge commits

danrol #1
58 commits 2,277 ++ 2,592 --

eg7eg7 #2
48 commits 3,176 ++ 2,973 --

eliaba #3
34 commits 1,146 ++ 422 --

EdenSharoni #4
34 commits 275 ++ 299 --

**General summary of work:**

Four meetings were performed. During our first meeting we discussed:

- What needs to be done during sprint #4
- How to improve spring#3
- Spring Data JPA - reference documentation
- Data Access Layer
- JUnit tests were improved and implemented according to Eyal’s notes.
- Gherkin tests document are successfully checked

**What went well throughout the sprint?**

Team met with Eyal after sprint#3, he explained how to improve our workflow and cleared up some questions we had on mind.

Gherkins and their implementations were our top priority to finish.

At every meeting we discuss what will be done and how to improve the next meeting.

**What should be improved in team work?**

The work wasn’t shared equally by every member of the team.

Additionally, not all members on the team been in all of the meeting due to other projects in other courses.

**What problems did the team encounter through the sprint?**

Some tests did not pass and that led us to rewrite those tests that took a lot of time and slowed our workflow.

**What is expected for the next sprint?**

- Improve pagination implementation by switching pagination from streams to PagingAndSortingRepository.
- Define what is Activity in our project and implement according to our definition
- ID Generation for UserEntity and ElementEntity through H2- HIBERNATE
- Get targets for the sprint #5 from Eyal

## List of  used technologies:

1. Spring.io
2. RESTful api
3. jdk 1.8
4. JUnit
5. Eclipse IDE
6. Github repository
7. Hibernate
8. JPA

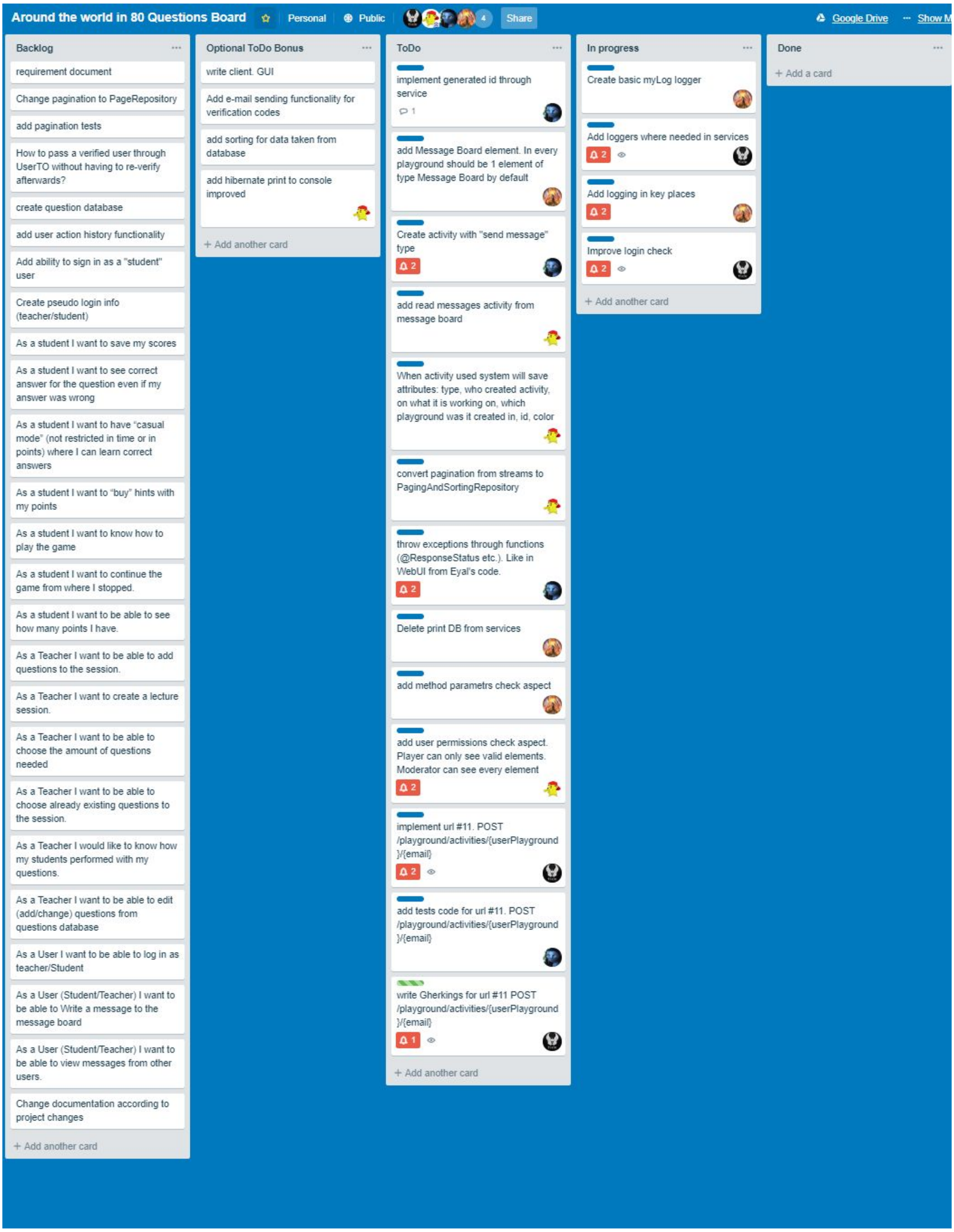
# Sprint Report #5

(16.12.18 - 30.12.18)

**Topic of the sprint:** Implement Aspect Oriented Programming in project

**Link to scrum board:** trello board

**Scrum board at the beginning of sprint (16.12.18):**



Around the world in 80 Questions Board ☆ Personal Public 4 Share Google Drive ··· Show M

**Backlog**

- requirement document
- Change pagination to PageRepository
- add pagination tests
- How to pass a verified user through UserTO without having to re-verify afterwards?
- create question database
- add user action history functionality
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to create a lecture session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I would like to know how my students performed with my questions.
- As a Teacher I want to be able to edit (add/change) questions from questions database
- As a User I want to be able to log in as teacher/Student
- As a User (Student/Teacher) I want to be able to Write a message to the message board
- As a User (Student/Teacher) I want to be able to view messages from other users.
- Change documentation according to project changes

+ Add another card

**Optional ToDo Bonus**

- write client. GUI
- Add e-mail sending functionality for verification codes
- add sorting for data taken from database
- add hibernate print to console improved

+ Add another card

**ToDo**

- implement generated id through service — 1
- add Message Board element. In every playground should be 1 element of type Message Board by default
- Create activity with "send message" type — 2
- add read messages activity from message board
- When activity used system will save attributes: type, who created activity, on what it is working on, which playground was it created in, id, color
- convert pagination from streams to PagingAndSortingRepository
- throw exceptions through functions (@ResponseStatus etc.). Like in WebUI from Eyal's code. — 2
- Delete print DB from services
- add method parametrs check aspect
- add user permissions check aspect. Player can only see valid elements. Moderator can see every element — 2
- implement url #11. POST /playground/activities/{userPlayground}/{email} — 2
- add tests code for url #11. POST /playground/activities/{userPlayground}/{email}
- write Gherkings for url #11 POST /playground/activities/{userPlayground}/{email} — 1

+ Add another card

**In progress**

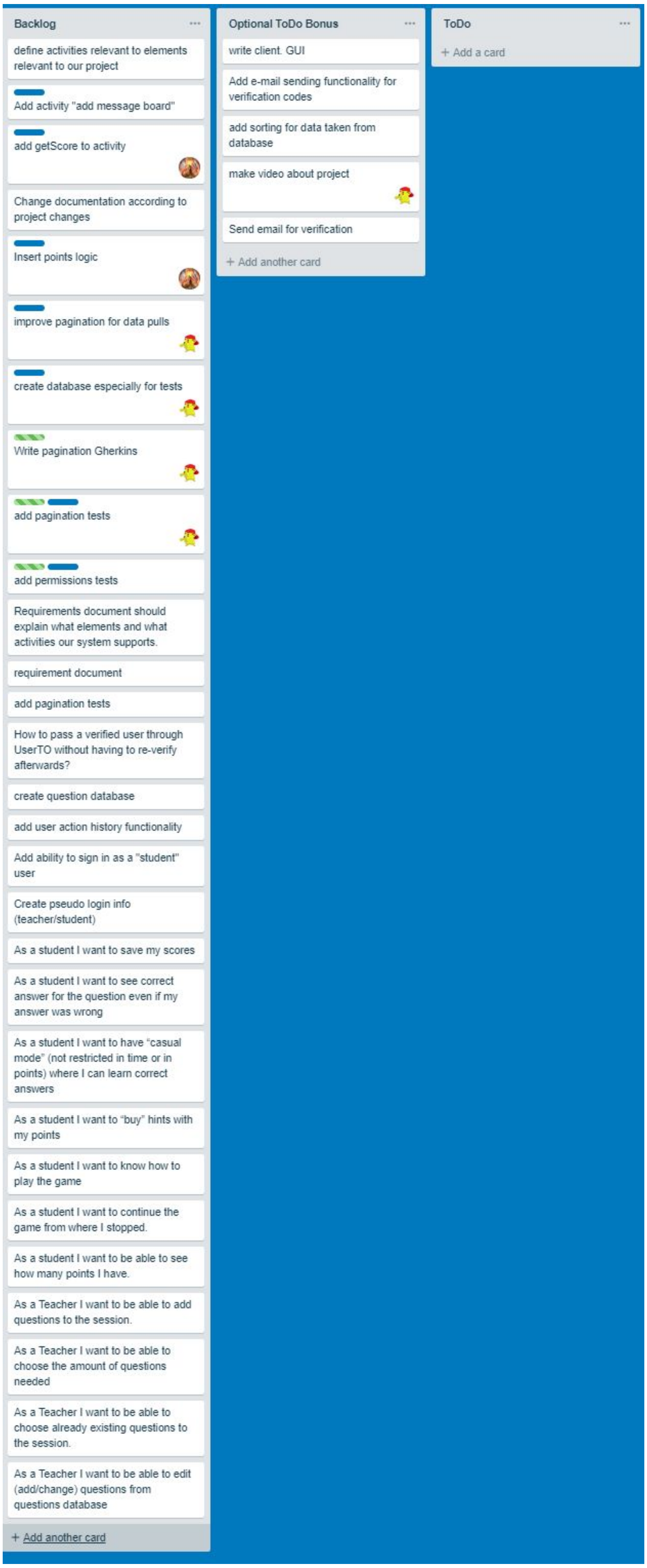
- Create basic myLog logger
- Add loggers where needed in services — 2
- Add logging in key places — 2
- Improve login check — 2

+ Add another card

**Done**

+ Add a card

## Scrum board at the end of sprint (30.12.18):



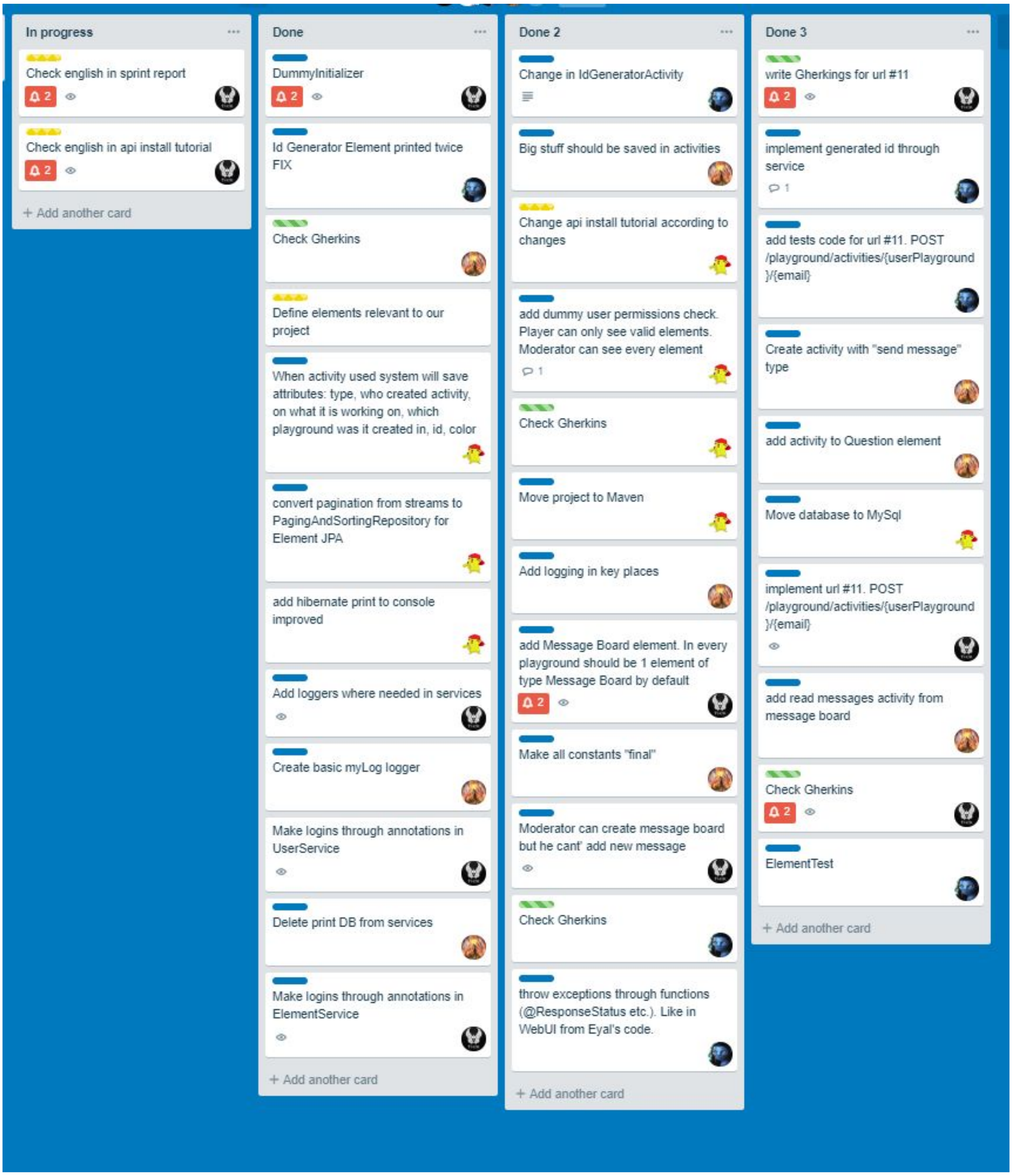
Backlog

- define activities relevant to elements relevant to our project
- Add activity "add message board"
- add getScore to activity
- Change documentation according to project changes
- Insert points logic
- improve pagination for data pulls
- create database especially for tests
- Write pagination Gherkins
- add pagination tests
- add permissions tests
- Requirements document should explain what elements and what activities our system supports.
- requirement document
- add pagination tests
- How to pass a verified user through UserTO without having to re-verify afterwards?
- create question database
- add user action history functionality
- Add ability to sign in as a "student" user
- Create pseudo login info (teacher/student)
- As a student I want to save my scores
- As a student I want to see correct answer for the question even if my answer was wrong
- As a student I want to have "casual mode" (not restricted in time or in points) where I can learn correct answers
- As a student I want to "buy" hints with my points
- As a student I want to know how to play the game
- As a student I want to continue the game from where I stopped.
- As a student I want to be able to see how many points I have.
- As a Teacher I want to be able to add questions to the session.
- As a Teacher I want to be able to choose the amount of questions needed
- As a Teacher I want to be able to choose already existing questions to the session.
- As a Teacher I want to be able to edit (add/change) questions from questions database

Optional ToDo Bonus

- write client. GUI
- Add e-mail sending functionality for verification codes
- add sorting for data taken from database
- make video about project
- Send email for verification

ToDo



**In progress**

- Check english in sprint report — 2
- Check english in api install tutorial — 2

**Done**

- DummyInitializer — 2
- Id Generator Element printed twice FIX
- Check Gherkins
- Define elements relevant to our project
- When activity used system will save attributes: type, who created activity, on what it is working on, which playground was it created in, id, color
- convert pagination from streams to PagingAndSortingRepository for Element JPA
- add hibernate print to console improved
- Add loggers where needed in services
- Create basic myLog logger
- Make logins through annotations in UserService
- Delete print DB from services
- Make logins through annotations in ElementService

**Done 2**

- Change in IdGeneratorActivity
- Big stuff should be saved in activities
- Change api install tutorial according to changes
- add dummy user permissions check. Player can only see valid elements. Moderator can see every element — 1
- Check Gherkins
- Move project to Maven
- Add logging in key places
- add Message Board element. In every playground should be 1 element of type Message Board by default — 2
- Make all constants "final"
- Moderator can create message board but he cant' add new message
- Check Gherkins
- throw exceptions through functions (@ResponseStatus etc.). Like in WebUI from Eyal's code.

**Done 3**

- write Gherkings for url #11 — 2
- implement generated id through service — 1
- add tests code for url #11. POST /playground/activities/{userPlayground }/{email}
- Create activity with "send message" type
- add activity to Question element
- Move database to MySql
- implement url #11. POST /playground/activities/{userPlayground }/{email}
- add read messages activity from message board
- Check Gherkins — 2
- ElementTest

**List of Students:**

1) Daniil Rolnik . ID: 334018009.

   Roles in the team: Product Manager, Product Owner, DevOps.

   Avatar:

   Github account: danrol

2) Elia Ben Anat . ID: 308048388.

   Role in the team: Database Administrator.

   Avatar:

   Github account: eliaba

3) Eden Dupont . ID: 204808596.

   Role in the team: QA Engineer.

   Avatar:

   Github account: eg7eg7

4) Eden Sharoni . ID: 315371906

   Role in the team: Scrum Master.

   Avatar:

   Github account: EdenSharoni

**Link to Github repository: https://github.com/danrol/ATW80Q**

**General summary of work:**

4 meetings were performed. During those meetings we discussed:

- What needs to be done during sprint #5
- How to implement AOP in our project and what aspects are relevant to our project
- What to add to the JpaActivityService to match our project
- Division of labor. Every team member received what goals they need to achieve during sprint #5
- methods were added to JpaActivityService and the logic of permissions is added to those methods
- Activity JUnit tests were added
- Gherkin tests document has been modified to fit the new requirements of the project
- pagination streams was added to the methods in  JpaActivityService
- Automated id generation added in services
- Integrated Maven into our project
- Integrated MySQL into the project

**What went well throughout the sprint?**

The dynamics of the group has improved by much , we are able to communicate and develop a common  line of thinking about the functionality of the project and the code structure.

The team has been working continuously throughout the sprint with no major break so every member is familiar with most of the code and the main logics of every section .

**What should be improved in team work?**

The management of time was lacking this sprint , we have managed to write the necessary functions to the JpaActivityService bud didnt have the time to write enough user stories to check the functionality  of the new methods.

**What  problems did the team encounter through the sprint?**

Most of the problem were with the logic of the login and permissions , how to include them in the program and what part of it should be the AOP and what other part of it in the logic of the services to manage .

We couldn’t find out why AutoGeneration.AUTO works vs Autogeneration.Identity in IdGeneratorActivity while it worked for other generators.

Permission check and pagination for some methods in jpa element service requires better understanding of how to write complex sql queries in ElementDAO.

There was a problem with switching to MySql database from H2 database. The problem was solved by switching to Maven.

**Why did we not complete all planned work?**

It is close to the end of the semester and other courses begin to realize that they must give us our last home assignments, and as a result, team members have a lot of other occupations, and less time to invest in the project.

Some things weren’t discussed enough between team members which lead to bad implementation.

**What is expected for the next sprint?**

- Activities implementation should be improved
- We plan to add more tests for activities
- fix non working tests for dummy services
- Add pagination tests
- Build client
- Decide how we want to present our project
- Synchronize requirements document with our project
- Add highscore logic
- Understanding how to write complex sql queries in ElementDAO and improve permissions check and pagination implementation

## List of used technologies:

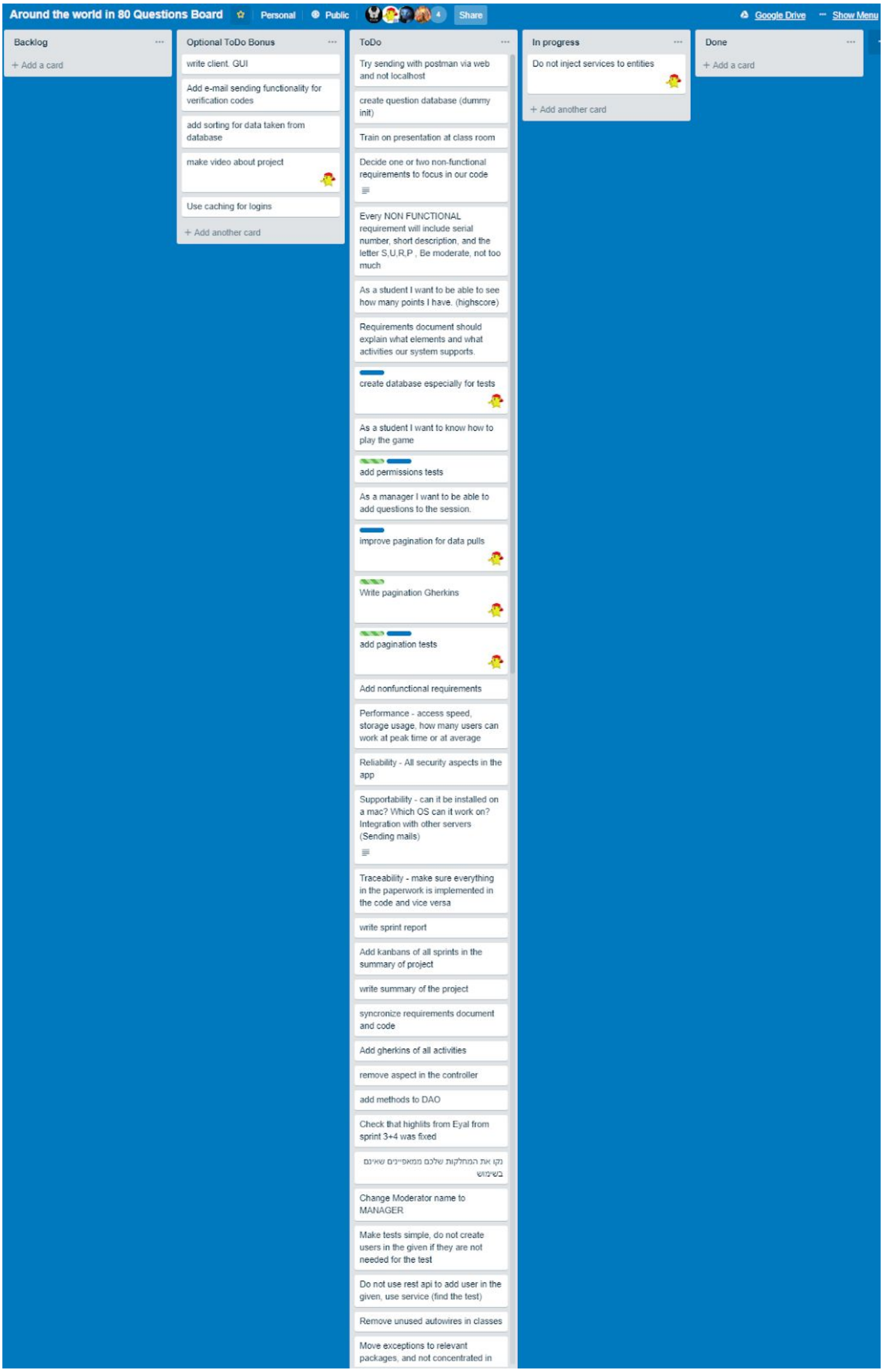
1) Spring.io
2) RESTful api
3) jdk 1.8
4) JUnit
5) Eclipse IDE
6) Github repository
7) Hibernate
8) JPA
9) Maven
10) MySQL
11) AspectJ
12) H2

# Sprint Report #6

(30.12.18 - 11.01.19)

**Topic of the sprint: prepare project and documents to final presentation**

**Scrum Board at the beginning of the sprint (30.12.18):**



Around the world in 80 Questions Board

Backlog

+ Add a card

Optional ToDo Bonus

- write client. GUI
- Add e-mail sending functionality for verification codes
- add sorting for data taken from database
- make video about project
- Use caching for logins

+ Add another card

ToDo

- Try sending with postman via web and not localhost
- create question database (dummy init)
- Train on presentation at class room
- Decide one or two non-functional requirements to focus in our code
- Every NON FUNCTIONAL requirement will include serial number, short description, and the letter S,U,R,P , Be moderate, not too much
- As a student I want to be able to see how many points I have. (highscore)
- Requirements document should explain what elements and what activities our system supports.
- create database especially for tests
- As a student I want to know how to play the game
- add permissions tests
- As a manager I want to be able to add questions to the session.
- improve pagination for data pulls
- Write pagination Gherkins
- add pagination tests
- Add nonfunctional requirements
- Performance - access speed, storage usage, how many users can work at peak time or at average
- Reliability - All security aspects in the app
- Supportability - can it be installed on a mac? Which OS can it work on? Integration with other servers (Sending mails)
- Traceability - make sure everything in the paperwork is implemented in the code and vice versa
- write sprint report
- Add kanbans of all sprints in the summary of project
- write summary of the project
- syncronize requirements document and code
- Add gherkins of all activities
- remove aspect in the controller
- add methods to DAO
- Check that highlits from Eyal from sprint 3+4 was fixed
- נקו את המחלקות שלכם ממאפיינים שאינם בשימוש
- Change Moderator name to MANAGER
- Make tests simple, do not create users in the given if they are not needed for the test
- Do not use rest api to add user in the given, use service (find the test)
- Remove unused autowires in classes
- Move exceptions to relevant packages, and not concentrated in

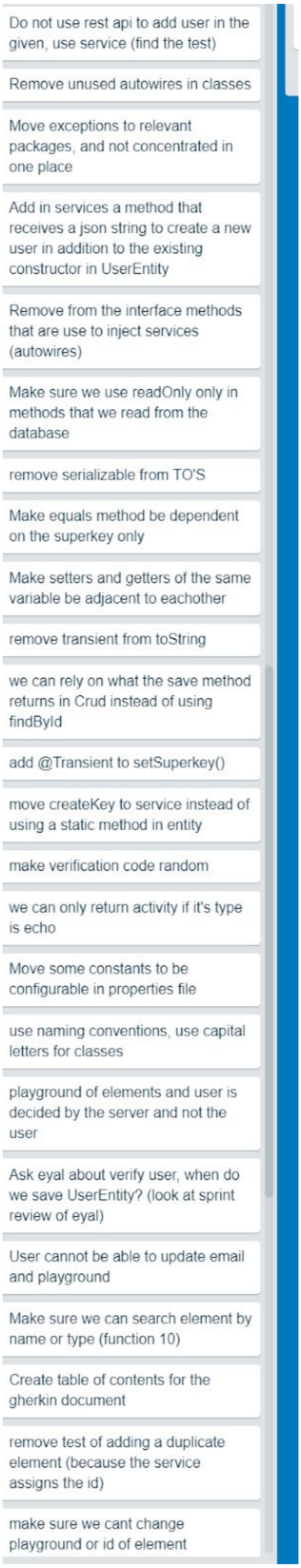
In progress

- Do not inject services to entities

+ Add another card

Done

+ Add a card



Do not use rest api to add user in the given, use service (find the test)

Remove unused autowires in classes

Move exceptions to relevant packages, and not concentrated in one place

Add in services a method that receives a json string to create a new user in addition to the existing constructor in UserEntity

Remove from the interface methods that are use to inject services (autowires)

Make sure we use readOnly only in methods that we read from the database

remove serializable from TO'S

Make equals method be dependent on the superkey only

Make setters and getters of the same variable be adjacent to eachother

remove transient from toString

we can rely on what the save method returns in Crud instead of using findById

add @Transient to setSuperkey()

move createKey to service instead of using a static method in entity

make verification code random

we can only return activity if it's type is echo

Move some constants to be configurable in properties file

use naming conventions, use capital letters for classes

playground of elements and user is decided by the server and not the user

Ask eyal about verify user, when do we save UserEntity? (look at sprint review of eyal)

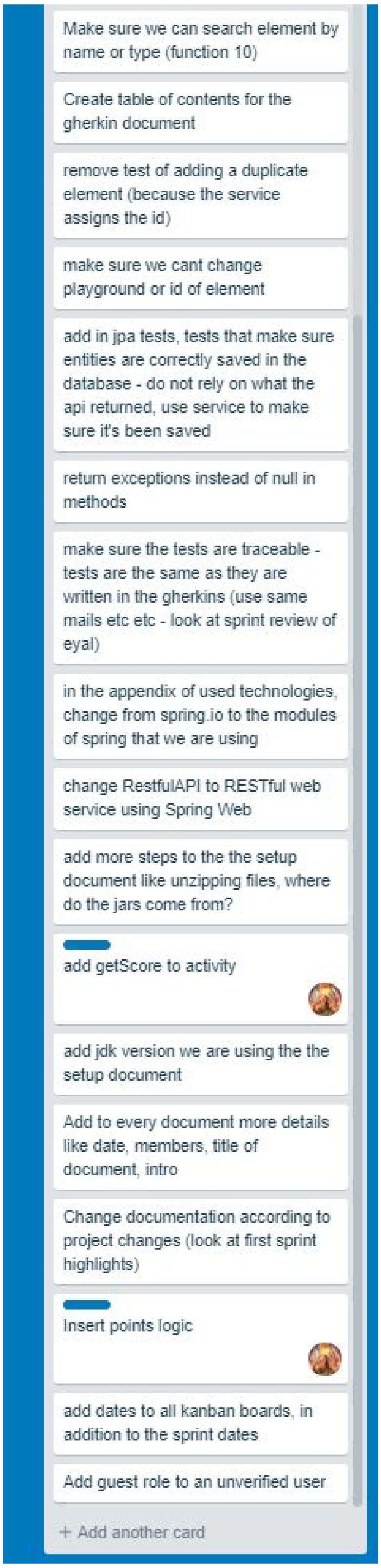
User cannot be able to update email and playground

Make sure we can search element by name or type (function 10)

Create table of contents for the gherkin document

remove test of adding a duplicate element (because the service assigns the id)

make sure we cant change playground or id of element



Make sure we can search element by name or type (function 10)

Create table of contents for the gherkin document

remove test of adding a duplicate element (because the service assigns the id)

make sure we cant change playground or id of element

add in jpa tests, tests that make sure entities are correctly saved in the database - do not rely on what the api returned, use service to make sure it's been saved

return exceptions instead of null in methods

make sure the tests are traceable - tests are the same as they are written in the gherkins (use same mails etc etc - look at sprint review of eyal)

in the appendix of used technologies, change from spring.io to the modules of spring that we are using

change RestfulAPI to RESTful web service using Spring Web

add more steps to the the setup document like unzipping files, where do the jars come from?

add getScore to activity

add jdk version we are using the the setup document

Add to every document more details like date, members, title of document, intro

Change documentation according to project changes (look at first sprint highlights)

Insert points logic

add dates to all kanban boards, in addition to the sprint dates

Add guest role to an unverified user

**Colors meaning in trello board:**

QA

Programming

Requirements document, Paperwork

moodle, general

**Link to scrum board:** trello board

**Scrum board at the end of the sprint (13.01.19):**

# TBD

**List of Students:**

1) Daniil Rolnik . ID: 334018009.

   Roles in the team: Product Manager, Product Owner, DevOps.

   Avatar:

   Github account: danrol

2) Elia Ben Anat . ID: 308048388.

   Role in the team: Database Administrator.

   Avatar:

   Github account: eliaba

3) Eden Dupont . ID: 204808596.

   Role in the team: QA Engineer.

   Avatar:

   Github account: eg7eg7

4) Eden Sharoni . ID: 315371906

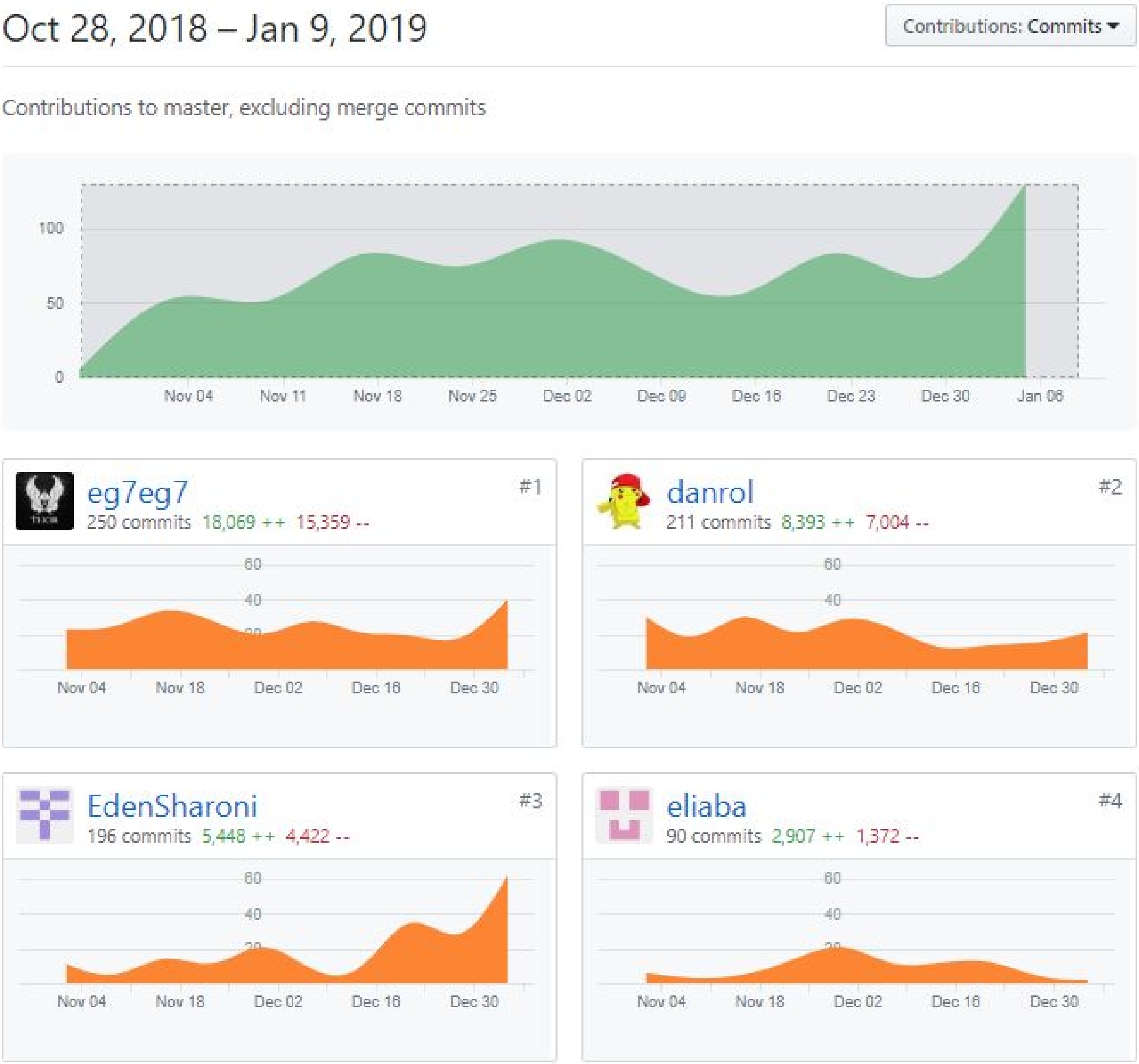
   Role in the team: Scrum Master.

Avatar:

Github account: EdenSharoni

**Link to Github repository: https://github.com/danrol/ATW80Q**

**Github Statistics at the end of Sprint #6:**



Oct 28, 2018 – Jan 9, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits

eg7eg7 #1
250 commits 18,069 ++ 15,359 --

danrol #2
211 commits 8,393 ++ 7,004 --

EdenSharoni #3
196 commits 5,448 ++ 4,422 --

eliaba #4
90 commits 2,907 ++ 1,372 --

**General summary of work:**

**4 meetings were performed. During those meetings we discussed:**

- What need to be improved from sprint #5
- What needs to be done during sprint #6
- What functionality we want to implement before final presentation
- How we want our presentation to look like
- What optional functionality we want to implement

**Through this sprint we added:**

- Gui/client for the project
- Verification code now sent to email

**What went well throughout the sprint?**

The dynamics of the group are at great level. It felt like every team member want to give maximum effort to finish project in best way we can.

**What should be improved in team work?**

Team flow was great this time. Nothing need to be improved. Every team member tried to give maximum to achieve best result through presentation.

**What  problems did the team encounter through the sprint?**

At some moment we understood that implementation of the url "/playground/elements/{userPlayground}/{email}/search/{attributeName}/{value}" need to be changed and it was performed.

**Why did we not complete all planned work?**

It is close to the end of the semester and other courses begin to realize that they must give us our last home assignments, and as a result, team members have a lot of other occupations, and less time to invest in the project.

Some things weren’t discussed enough between team members which lead to bad implementation.

## List of  used technologies:

1. Spring v2.1.1
    a. Spring Boot
    b. Spring Web
    c. Spring Data
2. RESTful Web Service with Spring Web
3. jdk 1.8
4. JUnit
5. Eclipse IDE
6. Github repository
7. Hibernate
8. JPA
9. Maven
10. MySql
11. AspectJ
12. Java Swing
13. Boot starter mail
14. H2