General Summary of Integrative Software Engineering (10141)

What went well throughout the course and what things you will use in next courses?

We performed many meetings and through each meeting we finished a lot of stories from scrum board. At the beginning of every sprint we performed disclosure about what need to be done through spring.

What should be improved for next courses?

During some sprints we didn't properly understood sprint targets from Eyal for next courses additional checks for sprint's/course's targets should be performed. Synchronization and understanding of team member missions still may work better than it worked.

What was most enjoyable thing of this course

It was one of the hardest programming courses through all BSc but it was also course that we learned most and got a lot of tools from software engineering field. It was really great to use that many different technologies in the project and to see it works at the end with all this complex functionality. It was also funny to bypass 900 commits in github.

What would you do different in this course

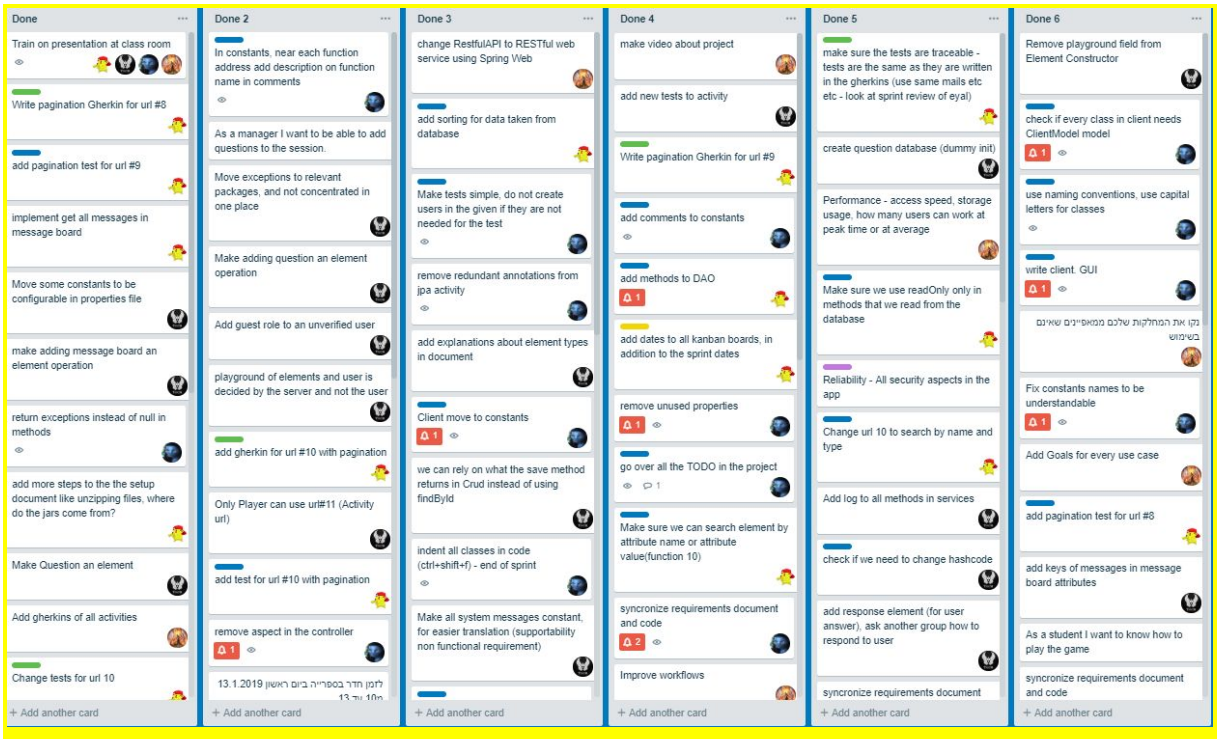
We think defining requirement document through first course was redundant. We never used it because at the beginning of every sprint we got new targets and started understood better what need to be done. At first sprint we still lacked understanding of what need to be done so requirement document was redundant.

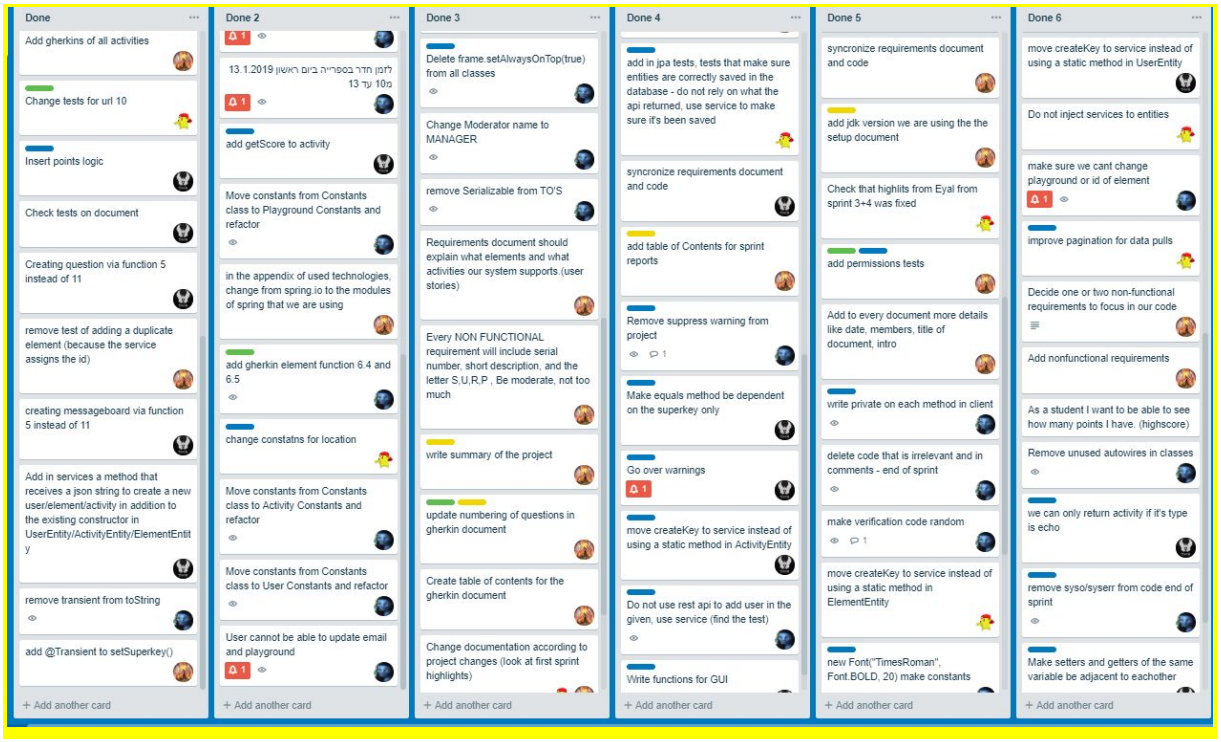
It would've save us time and headache if we started to work with constants earlier.

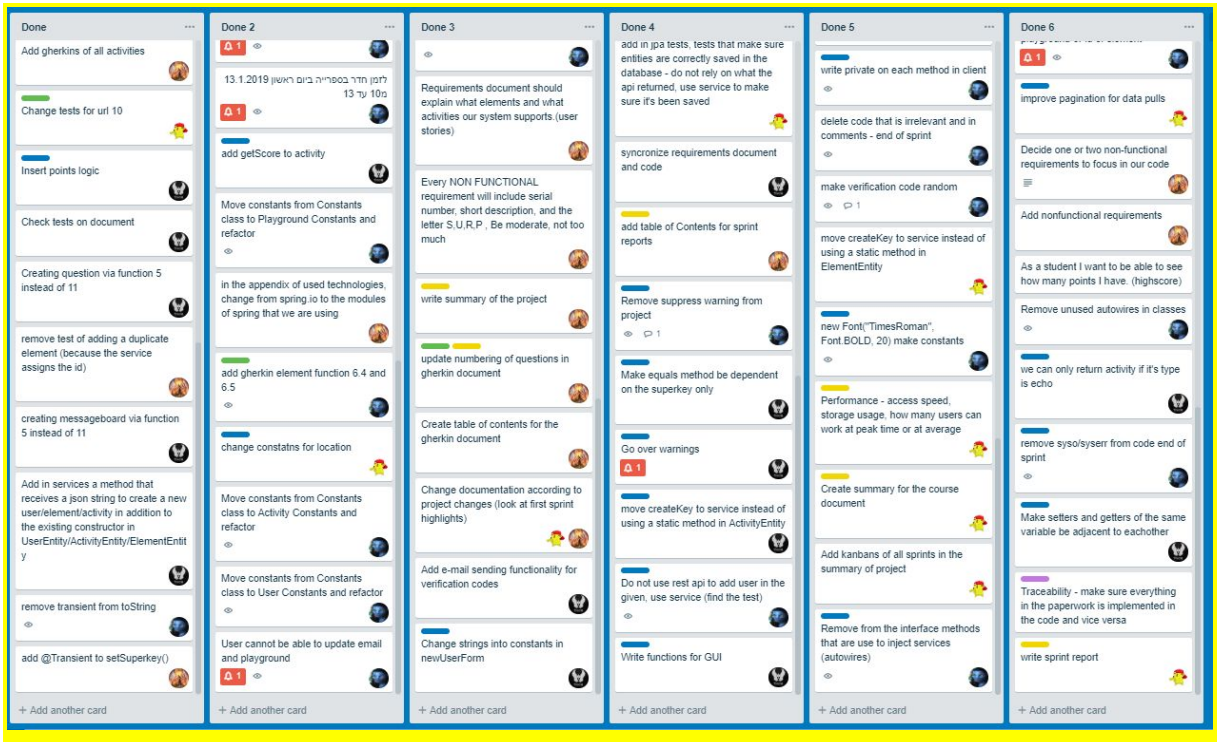
It may be better to show some code example of what we need to achieve at the end of the course.

We would like to see more comments with explanation in code. Sometimes it is hard to learn from code.

It would’ve been better to receive one last sprint remarks one week before we finish current sprint.






Done
Add gherkins of all activities
Change tests for url 10
Insert points logic
Check tests on document
Creating question via function 5 instead of 11
remove test of adding a duplicate element (because the service assigns the id)
creating messageboard via function 5 instead of 11
Add in services a method that receives a json string to create a new user/element/activity in addition to the existing constructor in UserEntity/ActivityEntity/ElementEntity
remove transient from toString
add @Transient to setSuperkey()
+ Add another card
Done 2
1
לזמן חדר בספרייה ביום ראשון 13.1.2019 מ10 עד 13
1
add getScore to activity
Move constants from Constants class to Playground Constants and refactor
in the appendix of used technologies, change from spring io to the modules of spring that we are using
add gherkin element function 6.4 and 6.5
change constatns for location
Move constants from Constants class to Activity Constants and refactor
Move constants from Constants class to User Constants and refactor
User cannot be able to update email and playground
1
+ Add another card
Done 3
Requirements document should explain what elements and what activities our system supports.(user stories)
Every NON FUNCTIONAL requirement will include serial number, short description, and the letter S,U,R,P , Be moderate, not too much
write summary of the project
update numbering of questions in gherkin document
Create table of contents for the gherkin document
Change documentation according to project changes (look at first sprint highlights)
Add e-mail sending functionality for verification codes
Change strings into constants in newUserForm
+ Add another card
Done 4
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
syncronize requirements document and code
add table of Contents for sprint reports
Remove suppress warning from project
1
Make equals method be dependent on the superkey only
Go over warnings
1
move createKey to service instead of using a static method in ActivityEntity
Do not use rest api to add user in the given, use service (find the test)
Write functions for GUI
+ Add another card
Done 5
write private on each method in client
delete code that is irrelevant and in comments - end of sprint
make verification code random
1
move createKey to service instead of using a static method in ElementEntity
new Font("TimesRoman", Font.BOLD, 20) make constants
Performance - access speed, storage usage, how many users can work at peak time or at average
Create summary for the course document
Add kanbans of all sprints in the summary of project
Remove from the interface methods that are use to inject services (autowires)
+ Add another card
Done 6
1
improve pagination for data pulls
Decide one or two non-functional requirements to focus in our code
Add nonfunctional requirements
As a student I want to be able to see how many points I have. (highscore)
Remove unused autowires in classes
we can only return activity if it's type is echo
remove syso/syserr from code end of sprint
Make setters and getters of the same variable be adjacent to eachother
Traceability - make sure everything in the paperwork is implemented in the code and vice versa
write sprint report
+ Add another card

Gherkins