News Site Project - Final Document Collection

Integrative Software Engineering | 2022b.elazar.fine | 22.05.22

Table of contents

News Site Project - Requirements Document

Integrative Software Engineering | 2022b.elazar.fine | updated 21.05.22

Maor Ofek, Chen Ifargan, Barak Moskovich, Elazar Fine, Denis Karabitski, Gal Karasnty

- 1. Introduction

  A news site system that gathers news articles from external sources and displays them to registered subscribers, saving their various preferences and actions.

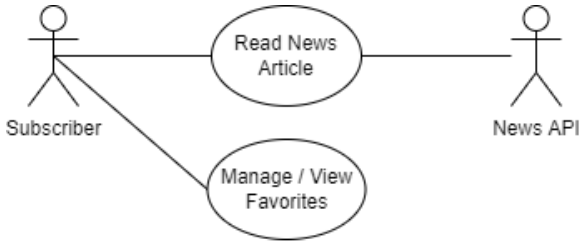
  - 1.1. Purpose of System:
    - First and foremost, the system will serve its main functionality - a news site, as described.
    - Secondly, by gathering relevant user data the system will have the future potential for a recommendation system in order to serve users with relevant articles, and have actionable user data for possible future analysis and / or targeted promotions.
  - 1.2 Scope of System:

    A full stack project consisting of a Java backend, a database and a React frontend with additional focus on user data gathering and less focus on data analysis and cybersecurity.

- 2. Actors and goals

Actor	**Goal(s)**
Subscribers (main actor)	Read news articles. Manage / View favorite articles.
News APIs	Serve news articles and their metadata.

- 3. Functional Requirements
    - 3.1 Use Case Diagram



    - 3.2 Use Case Details

      Use cases:

Use Case: Read News Article. Participating Actors: Subscriber, News API. Flow of Events: 1. The subscriber enters the system. 2. The system requests news articles from the News API. 3. The News API serves news articles according to the system's request. 4. The system displays the news articles to the subscriber. 5. The subscriber clicks an article to read. 6. The system saves the request details to the database. 7. The system redirects the subscriber to the article. Alternative Flow A: A.1. At step 3 there are no available articles matching the request. A.2. The system displays an error page/message to the subscriber.
Use Case: Manage / View Favorites. Participating Actors: Subscriber.

Flow of Events:
1. The subscriber enters the system.
2. The subscriber requests his favorites list.
3. The system saves the request details to the DB.
4. The system displays the favorites list.
5. The subscriber requests to remove an article from the list.
6. The system saves the request details to the DB.
7. The system removes the requested article from the list and refreshes the list.
8. The subscriber requests to view an article from the list.
9. The system displays the news article to the subscriber
.
Alternative Flow A:
A.1. At step 3 the subscriber has no favorites.
A.2. The system displays a message to the subscriber informing him that he has no favorites.

- 4. Non Functional Requirements

#	Requirement Description	Requirement Type
1	The system's user interface will be simple and easy to use.	Usability

News Site Project - Technologies Used

Integrative Software Engineering | 2022b.elazar.fine

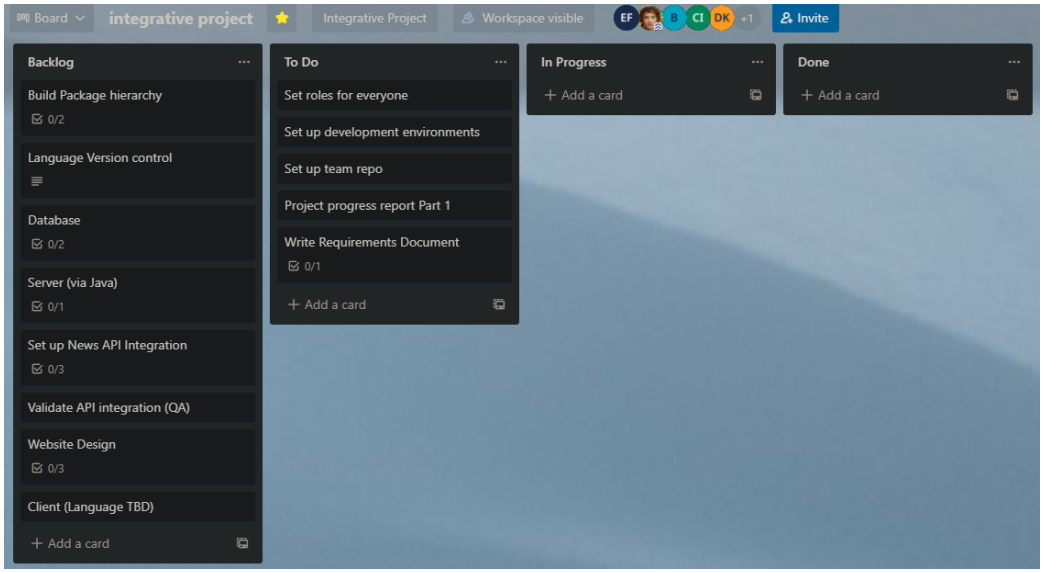
Database	MySQ, H2.
Server	Spring boot, Spring data, Hibernate, lombok, JUnit, Apache Commons Validator, Maven.
Client	React, bootstrap, redux, npm.
Development Tools	IntelliJ, VS Code, Postman, Draw.io, Bitbucket.
Communication and project management Tools	WhatsApp, Kanban board using Trello, discord, zoom, xoyondo.

News Site Project - Project General Summary

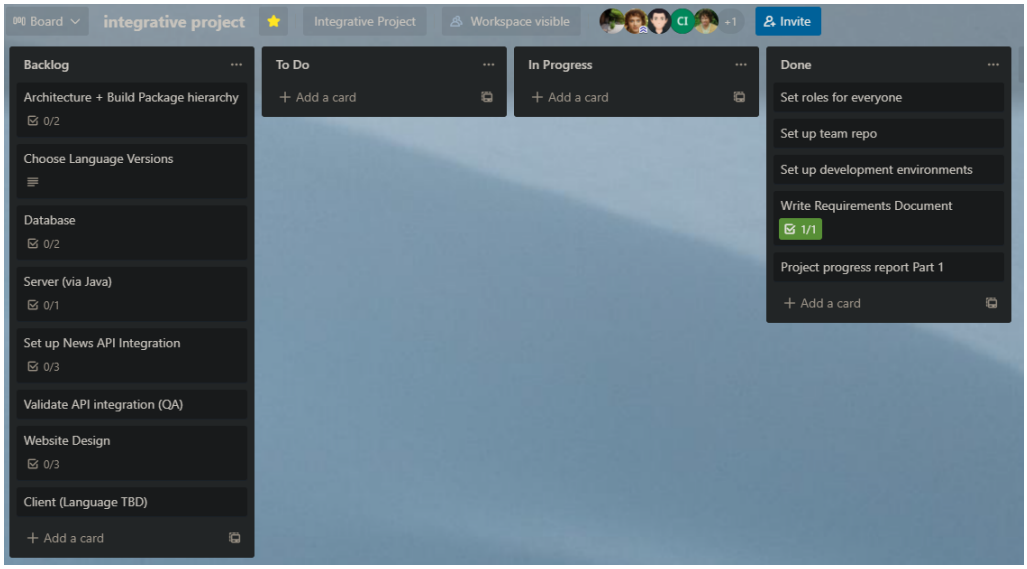
Integrative Software Engineering | 2022b.elazar.fine | 22.05.22

Project Kanban Boards:

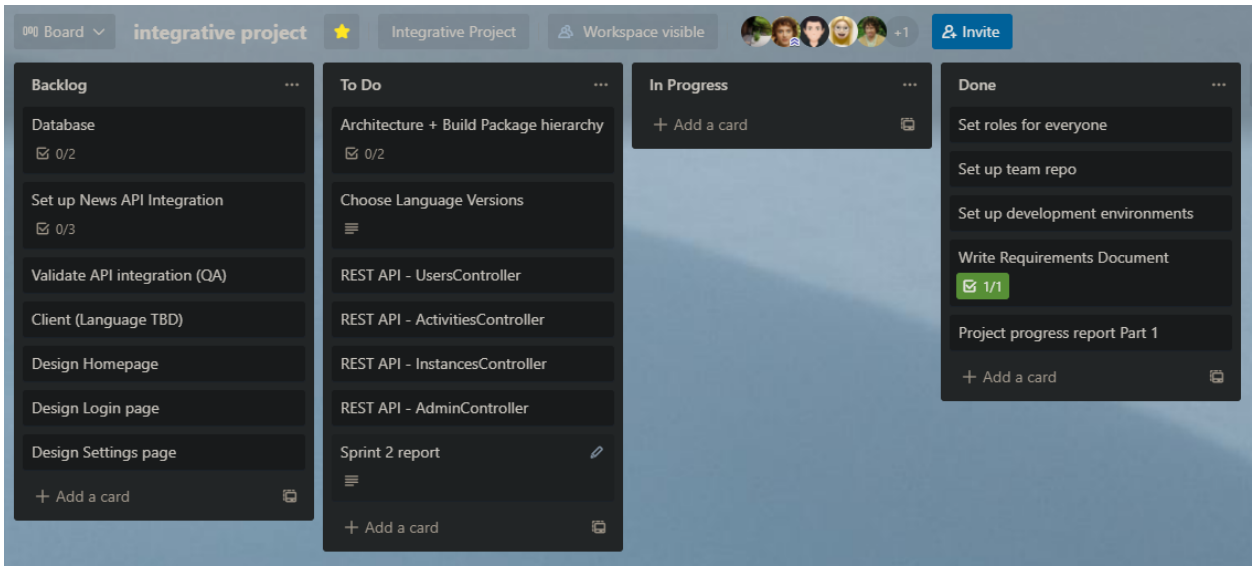
20.02.2022 - Project start, sprint 1.



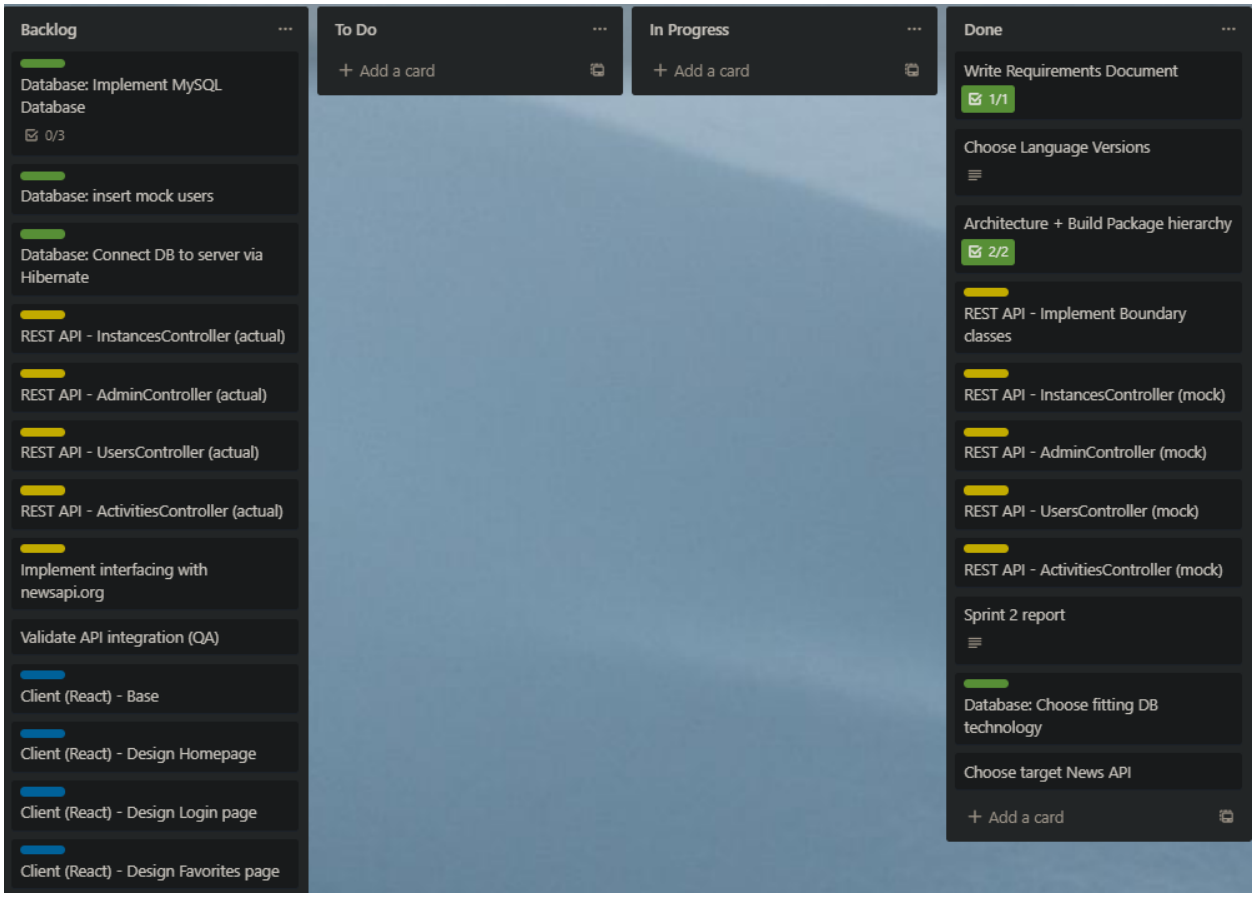
27.02.2022 - end sprint 1.



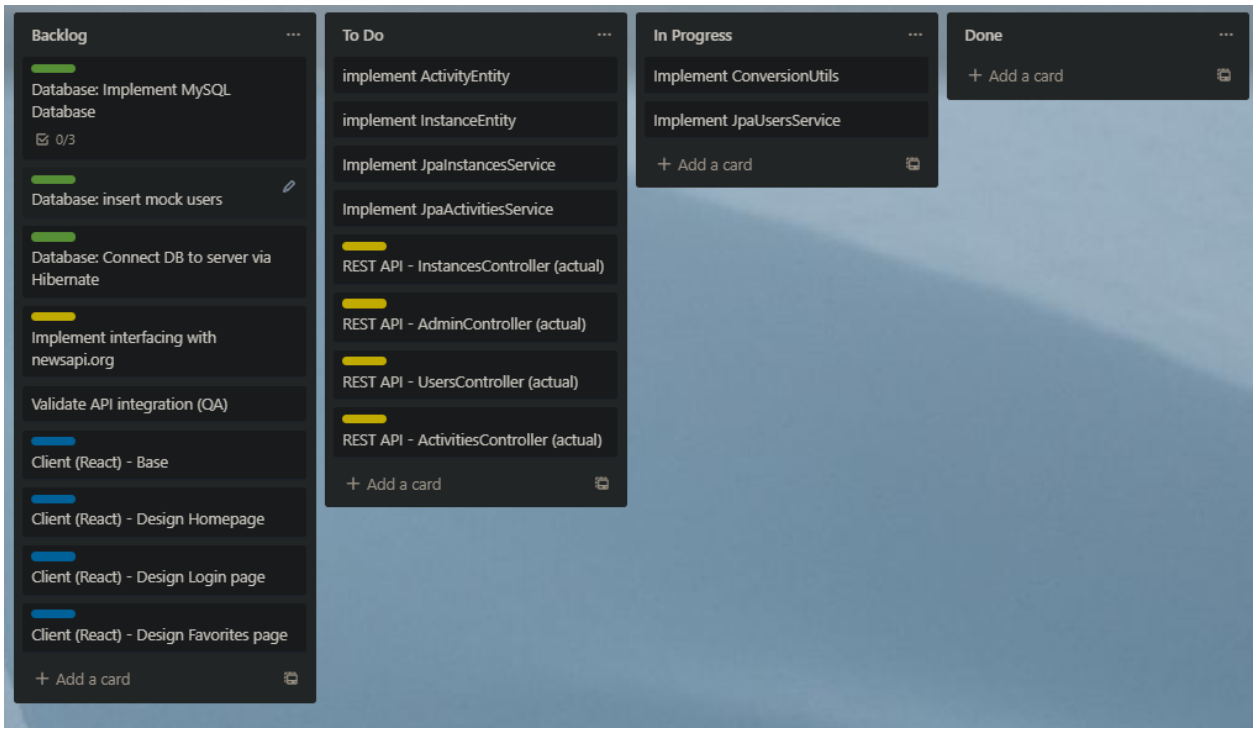
06.03.22 - sprint 2 beginning.



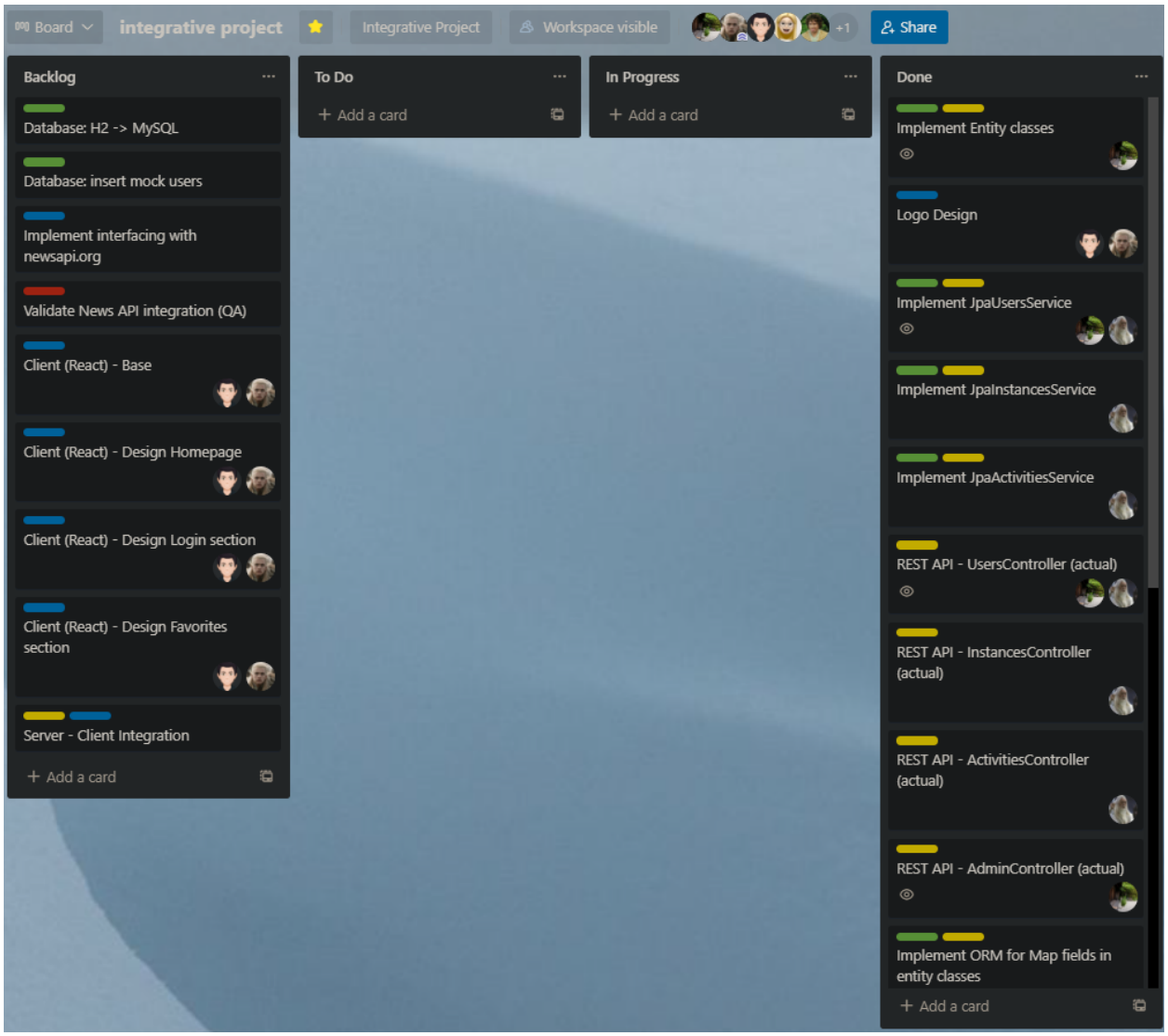
26.03.22 - sprint 2 end.



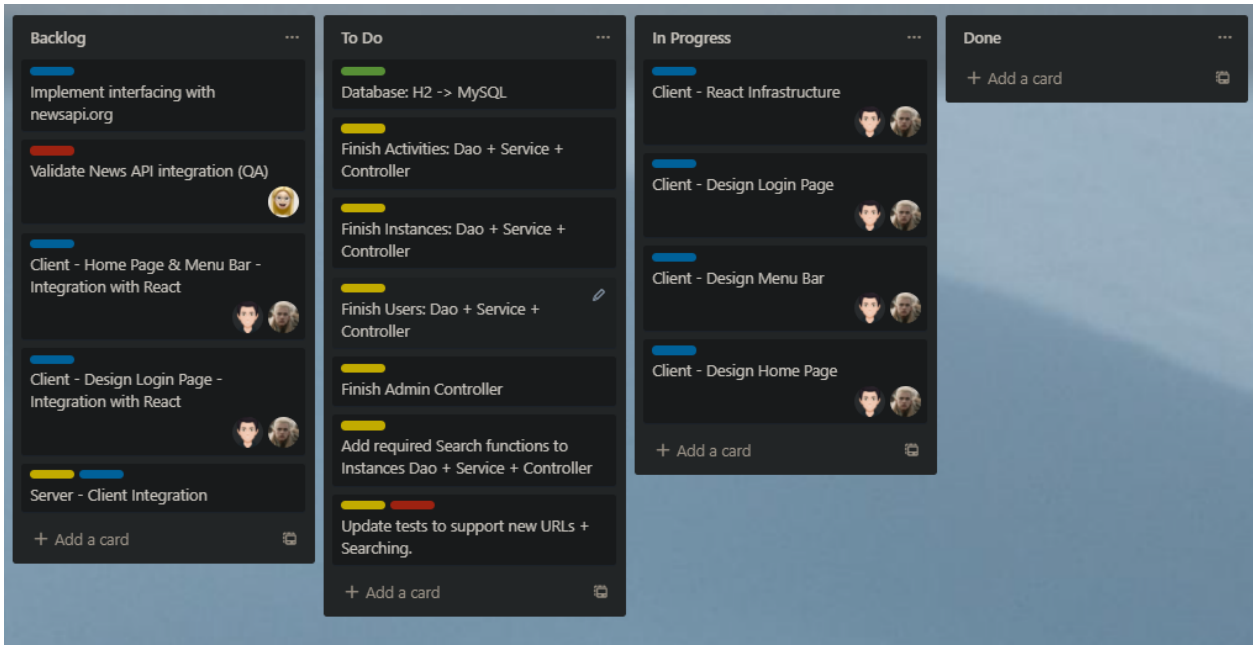
27.03.22 - spring 3 beginning.



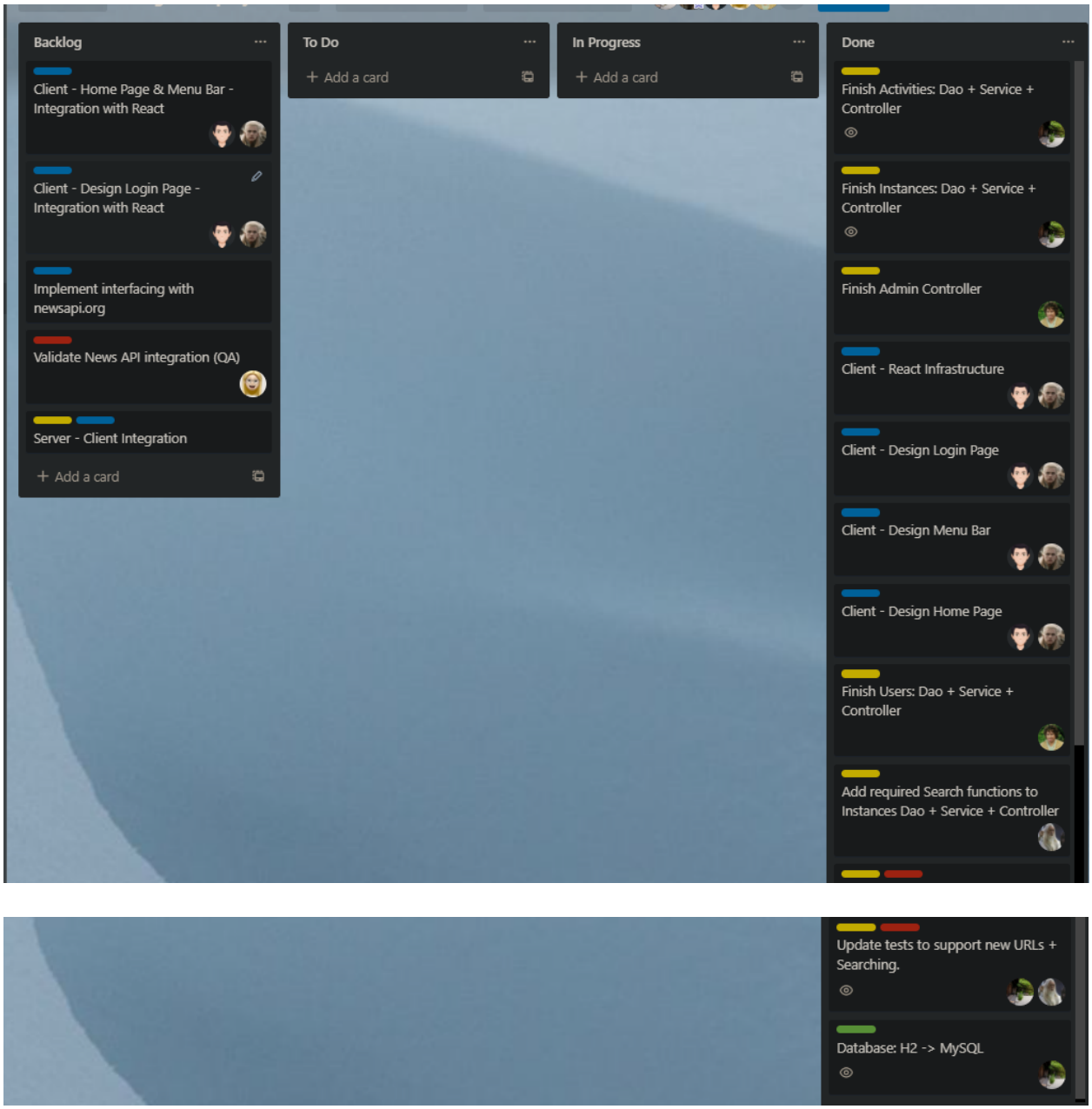
23.04.22 - sprint 3 end.



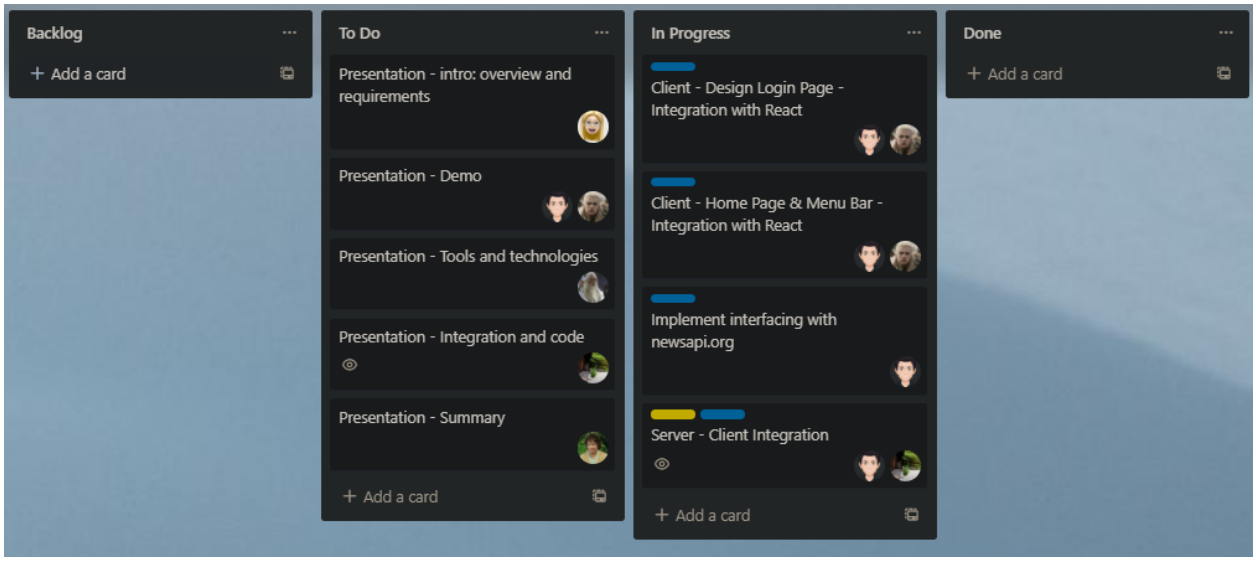
23.04.22 - sprint 4 beginning.



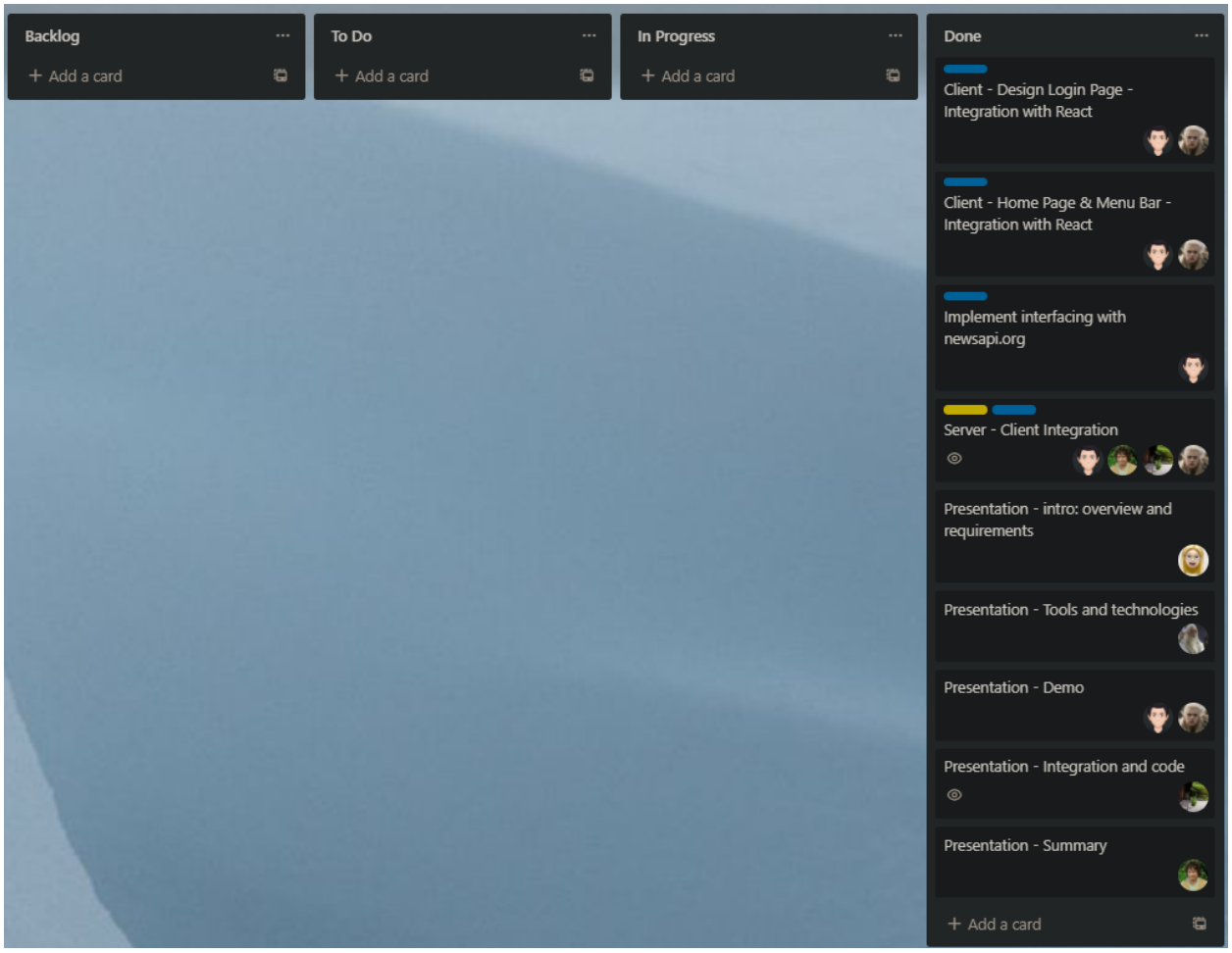
07.05.22 - sprint 4



08.05.22 - sprint 5 beginning.



22.05.22 - sprint 5 and project end.



General Summary

- **What worked well during the project that you would like to continue in future projects?**
    - Every team member was able to work somewhat independently and choose what to work on freely from the kanban board.
    - The combination of kanban board for tracking and managing tasks + WhatsApp and discord for extra communication worked very well throughout the project.
- **How can you improve your work in future projects?**
    - We sometimes started to work on the sprint's tasks (not talking about fixing the previous one) only halfway through the sprint, sometimes causing too concentrated work load during sprints' end.
    - We could have had more knowledge sharing during the project, for example, doing meetings where each member could explain what he did in order to accomplish tasks outside other members' capability / knowledge.
- **What did you most enjoy during the projects?**
    - The project workflow (sprints, weeklies, etc…) resembles industry workflows.
    - Learning new technologies.
    - Having freedom in certain areas to choose what and how to implement
    - Working on a project with friends.

- **If you had to start the project from zero again, what would you do differently?**
    - We would divide the sprints not evenly but according to the sprints' difficulty. (for example give sprints with more / harder tasks 3+ weeks instead of 2, and for easier ones 1 week).
    - Using story points in order to estimate how long each task will take during a sprint.
- **How did working remotely impact the project?**
    - Working remotely on software projects is what we like and what we are used to - in our studies and for some In our work as well.
    - Working remotely also fits how we divided the workflow - team members working independently on smaller tasks and sometimes doing pair programming via discord.

News Site Project - Sprint 5 Progress Report

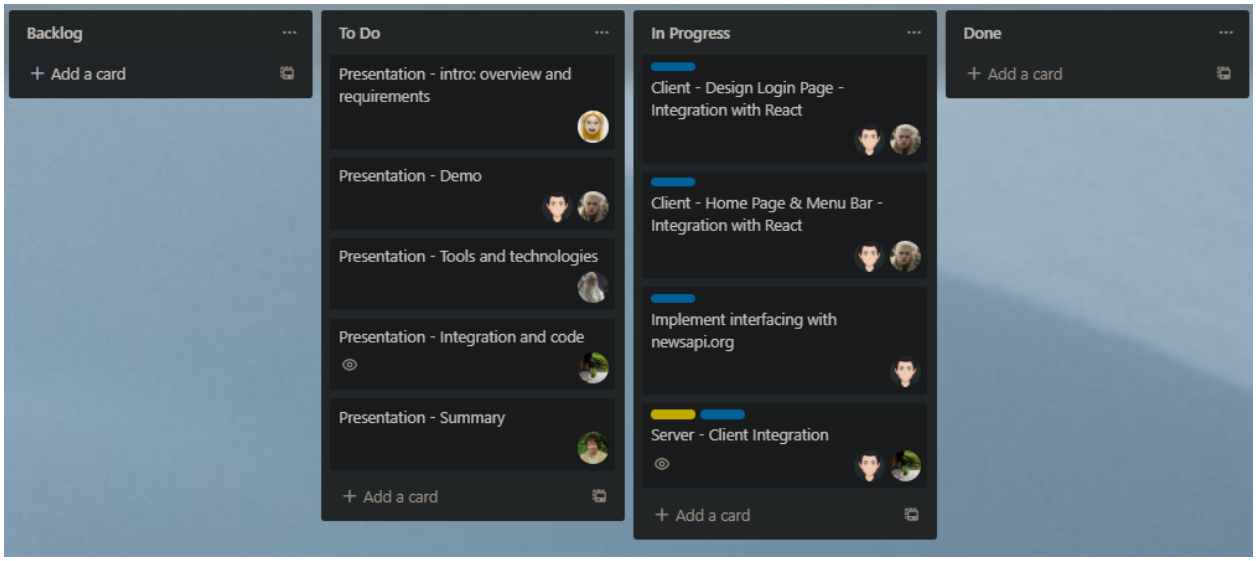
Integrative Software Engineering | 2022b.elazar.fine | 22.05.22

Students:

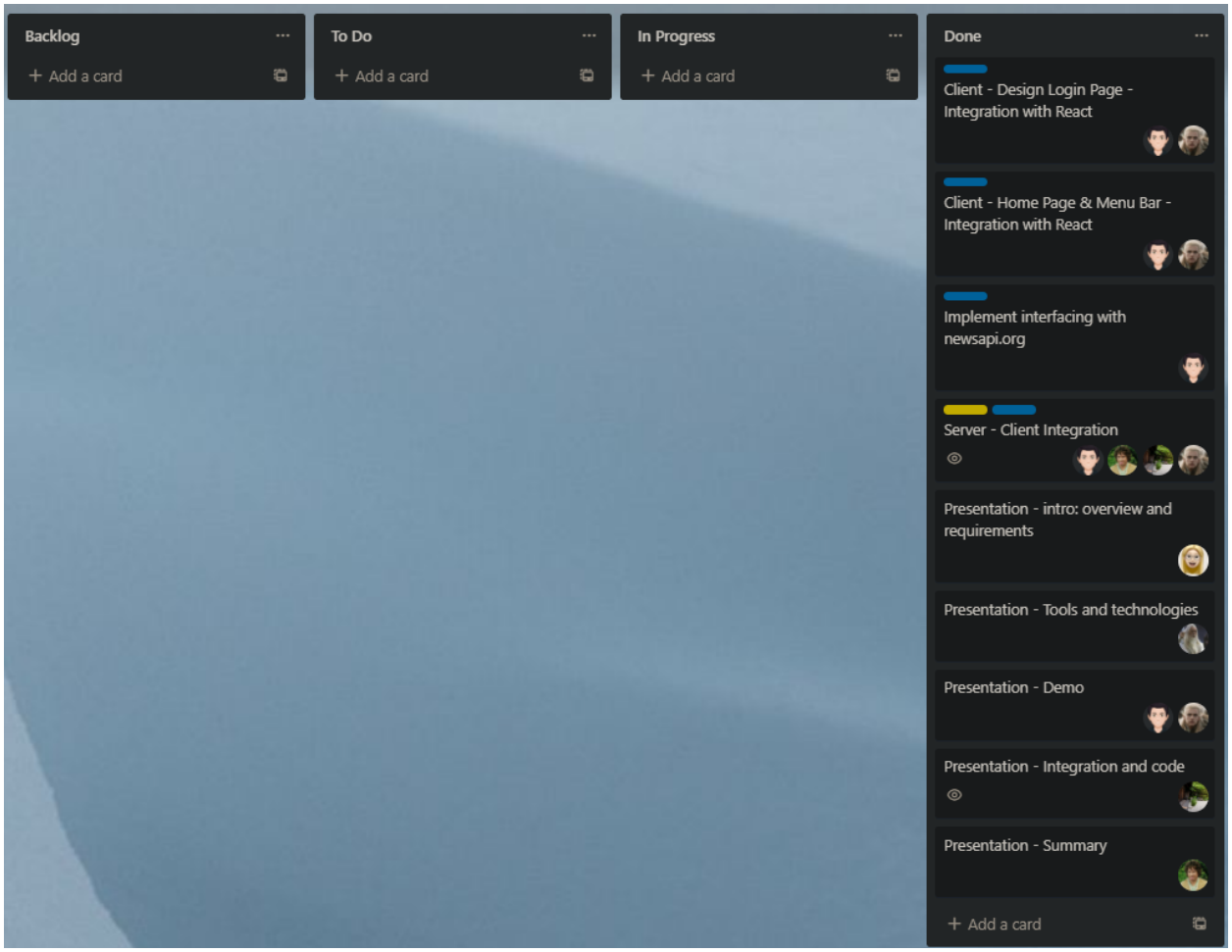
Name	Avatar	Role
Maor Ofek		DBA + Product Owner + Team
Chen Ifargan		Technical Writer + Team
Barak Moskovich		UI/UX + Team
Elazar Fine		Team Leader + Team
Denis Karabitski		DevOps + Scrum Master + Team
Gal Karasnty		QA + Team

Kanban Boards

Kanban board from sprint beginning (08.05.22):



Kanban board from sprint end (22.05.22):



General summary of work:

- **What went well for the team and should be continued in the next phases of work:**
    - Even with the semester's end and a lot of work on the final project and other courses' projects and assignments we managed to finish all the main tasks we set for ourselves.
    - The work methodology and tools we used in previous sprints works well, so we did not change them.
    - Team members who could contribute did so even when swamped with other tasks (like final project and other projects and assignments).
- **What should be improved in team work:**
    - We realized we could have done more knowledge sharing during the project, we had no time to fill this gap during this sprint considering this is not necessarily a huge priority for the end product's success, although still unfortunate.
- **What problems did the team encountered through this phase of work:**

    - Some functionality in the client side could not be 100% matched with the server's API, so we had to choose the lesser evil between the following:
        1. Either modify the server's API (forbidden)
        2. Or create extra requests for the server upon each request.
        3. Or create fictive instances and users.

      In the end we chose a combination of (2) and (3) that made the most sense to us, without creating too much extra burden on completing the integration on the client's side.

- There was an optional task we could have done if we really wanted to, but we decided not to because of time constraints and its very low priority: automated client QA
- We did not find a time for a meeting to prepare for the presentation such that all team members could be present, we ended up missing one member who later filled in with other members on his own.

- **<u>Why did we not complete all planned work:</u>**

  N/A