Project Requirements Document

Project: Country Club - Digital Twin

Team: 2021b.Daniel.Aizenband

Table of Contents:

1.Introduction:

The Vision

- Provide an easy solution for Country clubs who want to manage their facilities and manpower better.

1.1 Purpose of the system

- We envisioned a program that's purpose is to create a digital twin for a country club. Creating a digitalized replica of the place in real life dealing with the pandemic restrictions, while adding functionalities to support the daily routine of a country club - such as creating new users or updating their permissions allowing them to invoke different operations, managing inventory and most importantly allow members to sign up to classes, reserve a spot to an attraction.

1.2. Scope of System

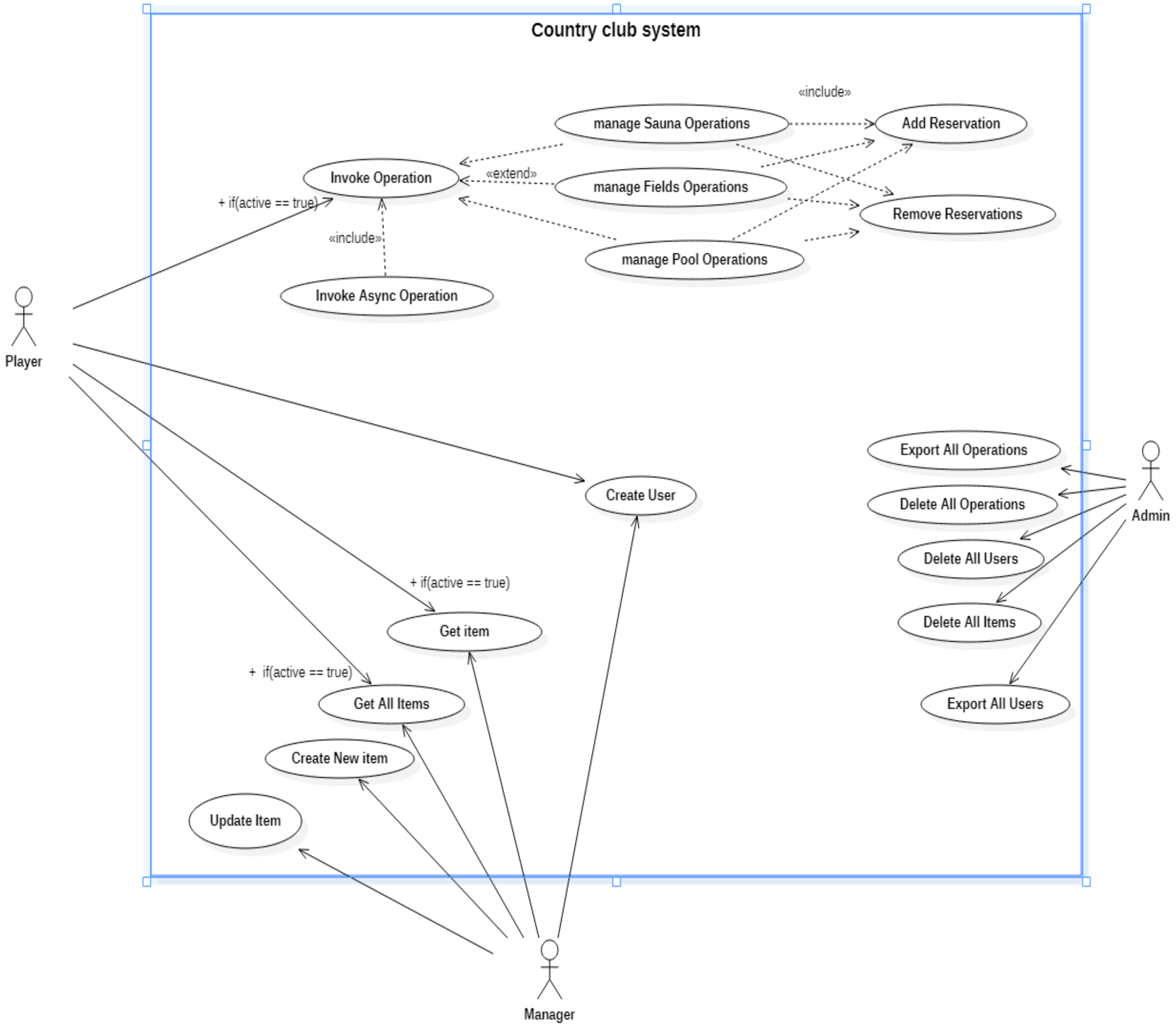
- The system will be a web API for the country club, allowing members to perform operations. The system will replace the interaction with the reception of the country club and allow the admin to keep better track of the facilities and members.
    - In scope:
        - Allow members to sign into classes
        - Manage sports fields reservations
        - Reserve spot at various facilities (such as sauna)
        - Manage members in and out of the pool
        - Provide facility information
        - Provide information about members that are in the club or stuff of the club
    - Out of scope:
        - Money management – membership payment and different types of membership
        - Members – keeping track of transactions between members and the country club
        - Time management – keeping reservations base on hours for sports fields, reserving a spot for a group of members and making sure that they have arrived.
        - Different actions – we wanted to add the "MANAGER" additional operations – controlling pool temp
        - Rent items that are unique to the attraction (such as towel to the sauna, tennis bet for the tennis court, etc.).

2. Actors and Goals:

- **Player** - Primary actor, uses the app to get notified about new activities and sign up.
  - Goals –
    - Successfully reserve a spot from the available attractions.
    - Successfully get a description of active items, attractions availability.
- **Manager** - Support actor, uses the app to update and manage facilities and items.
  - Goals –
    - Successfully create and update items.
- **Admin** - Support actor, uses the app to manage users and operations.
  - Goals –
    - Successfully get all operations and users.
    - Successfully delete all operations, items, and users.

3. Functional Requirements:

3.1 Use Case Diagram



3.2 Use Case Details

Use case	Description
Name	Create User
Goals	Add new user
Participating Actors	Player, Manager
Basic Workflow	New user is added
Alternative Workflow	Invalid user email, username or avatar - throw RuntimeException

Use case	Description
Name	Update User
Goals	Update existing user attributes
Participating Actors	Player, Manager
Basic Workflow	Existing user attributes are updated
Alternative Workflow	User does not exist - throw RuntimeException

Use case	**Description**
Name	Get Item
Goals	Find an Item by ID
Participating Actors	Player, Manager
Basic Workflow	Desired Item is returned
Alternative Workflow	Item does not exist - throw RuntimeException If invoked by Player but item not Active – throw RuntimeException

Use case	**Description**
Name	Delete All Users
Goals	Delete all the users in the database
Participating Actors	Admin
Basic Workflow	All users will be deleted from the database
Alternative Workflow	UserRole of invoker is not ADMIN or user does not exist - throw RuntimeException

Use case	**Description**
Name	Delete All Operations
Goals	Delete all the operations in the database
Participating Actors	Admin
Basic Workflow	All operations will be deleted from the database
Alternative Workflow	UserRole of invoker is not ADMIN or user does not exist - throw RuntimeException

Use case	**Description**
Name	Update Item
Goals	Update existing items attributes
Participating Actors	Manager
Basic Workflow	Existing item attributes are updated
Alternative Workflow	Item does not exist - throw RuntimeException

Use case	**Description**
Name	Create Item
Goals	Add new Item
Participating Actors	Manager
Basic Workflow	New hub is added
Alternative Workflow	UserRole of invoker is not MANAGER or user does not exist - throw RuntimeException

Use case	**Description**
Name	Delete All Items
Goals	Delete all the items in the database
Participating Actors	Admin
Basic Workflow	All hubs will be deleted from the database
Alternative Workflow	UserRole of invoker is not MANAGER or user does not exist - throw RuntimeException

Use case	Description
Name	Export All Operations
Goals	Receive all operations stored
Participating Actors	Admin
Basic Workflow	All operations in the database will return
Alternative Workflow	No operations in the database, UserRole of invoker is not ADMIN or user does not exist - throw RuntimeException

Use case	Description
Name	Get All Items
Goals	Receive all items stored
Participating Actors	Player, Manager
Basic Workflow	All items in the database will return (If Invoked by Player – only active items will return)
Alternative Workflow	No items in the database– throw RuntimeException Player invokes but there are no active items – throw RuntimeException

Use case	**Description**
Name	Manage Sauna Operations
Goals	Add/remove Users to/from sauna
Participating Actors	Player
Basic Workflow	User will be added/removed to/from item attributes and then the item will be updated
Alternative Workflow	Capacity has reached max number of users – No reservation is made No available reservations – No action taken

Use case	**Description**
Name	Manage Fields Operations
Goals	Add/remove a group of Users to/from various sports fields
Participating Actors	Player
Basic Workflow	User will be added/removed to/from item attributes and then the item will be updated
Alternative Workflow	Capacity has reached max number of users – No reservation is made No available reservations – No action taken

Use case	Description
Name	Manage Pool Operations
Goals	Add/remove Users to/from pool
Participating Actors	Player
Basic Workflow	User will be added/removed to/from item attributes and then the item will be updated
Alternative Workflow	Capacity has reached max number of users – No reservation is made No available reservations – No action taken

4. Non-Functional Requirements

Requirements Number	Requirements Description	Requirement Type
		U – Usability R – Reliability P – Performance S - Supportability
1	Ease of learning. The system must be easy to learn for both novices and users with experience from similar systems.	U – Usability
2	Defined edge conditions that might prevent failure.	R – Reliability
3	JUnit tests - Run upon each build to verify code integrity.	S - Supportability

5. Technologies:

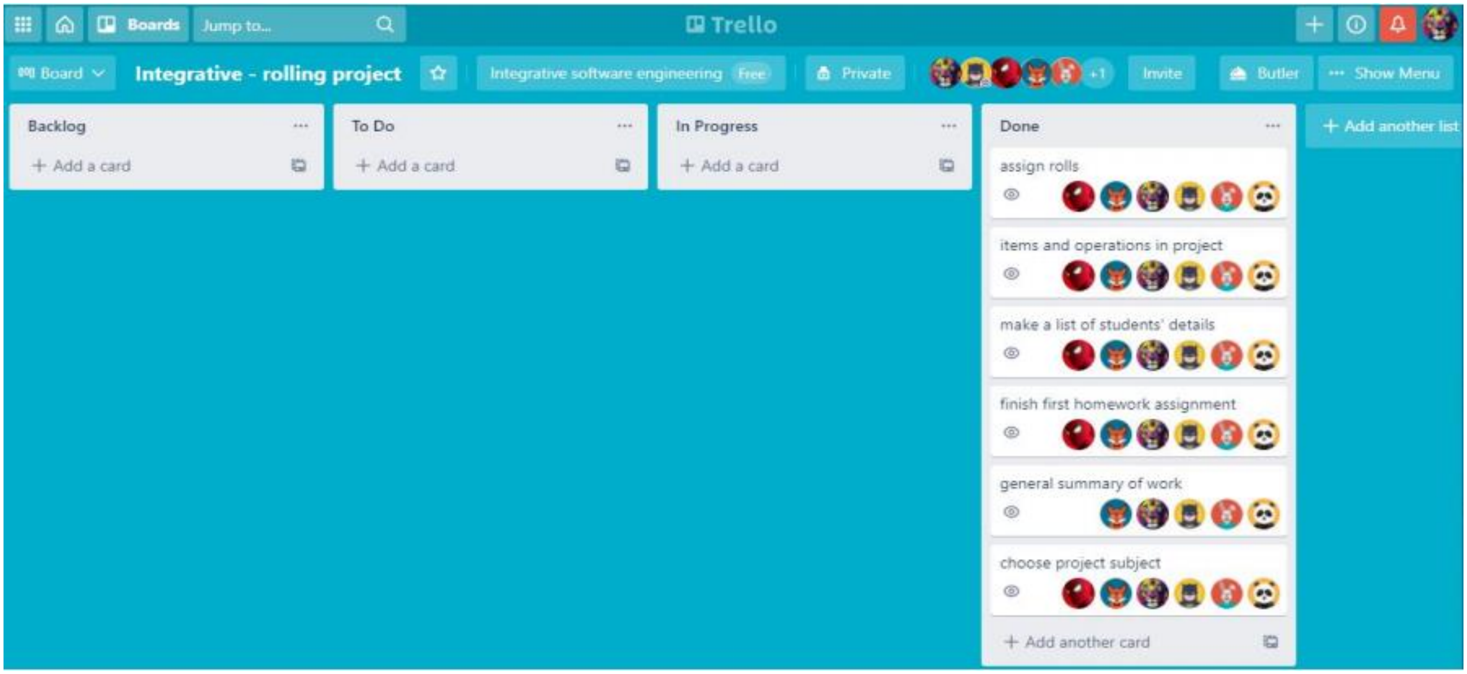
- Spring Web
- Spring Data JPA
- Spring Test
- Spring Boot
- MySQL Server - Version 8.0.20
- javax.persistence-API
- Spring-boot-maven-plugin
- Bitbucket for Code control
- Google Drive File sharing system
- Discord server for live communication
- ZOOM
- MOM - Messaging Oriented Middleware
- H2 DataBase
- Tomcat
- React
- Trello
- Postman

6.Genaral summary of integrative project – Country Club

6.1 Kanban boards throughout the projects:

Sprint 1:

Pre-screenshot of Kanban board - 09.03.21



Backlog

+ Add a card

To Do

+ Add a card

In Progress

+ Add a card

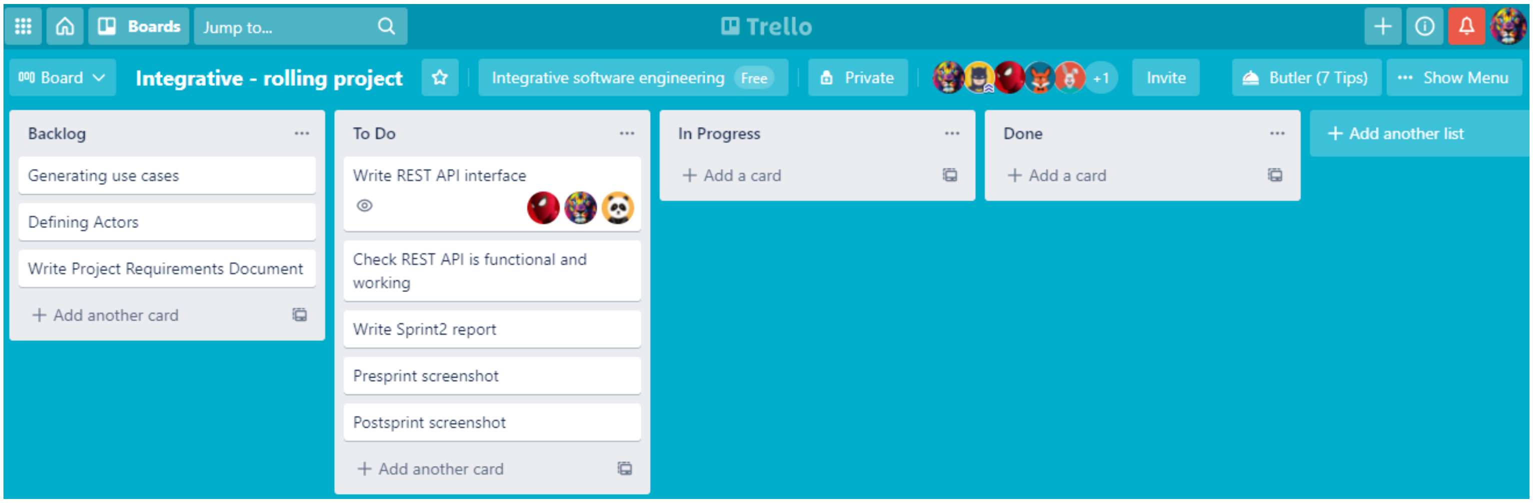
Done

- assign rolls
- items and operations in project
- make a list of students' details
- finish first homework assignment
- general summary of work
- choose project subject

+ Add another card

Sprint 2:

Pre-screenshot of Kanban board - 14.03.21



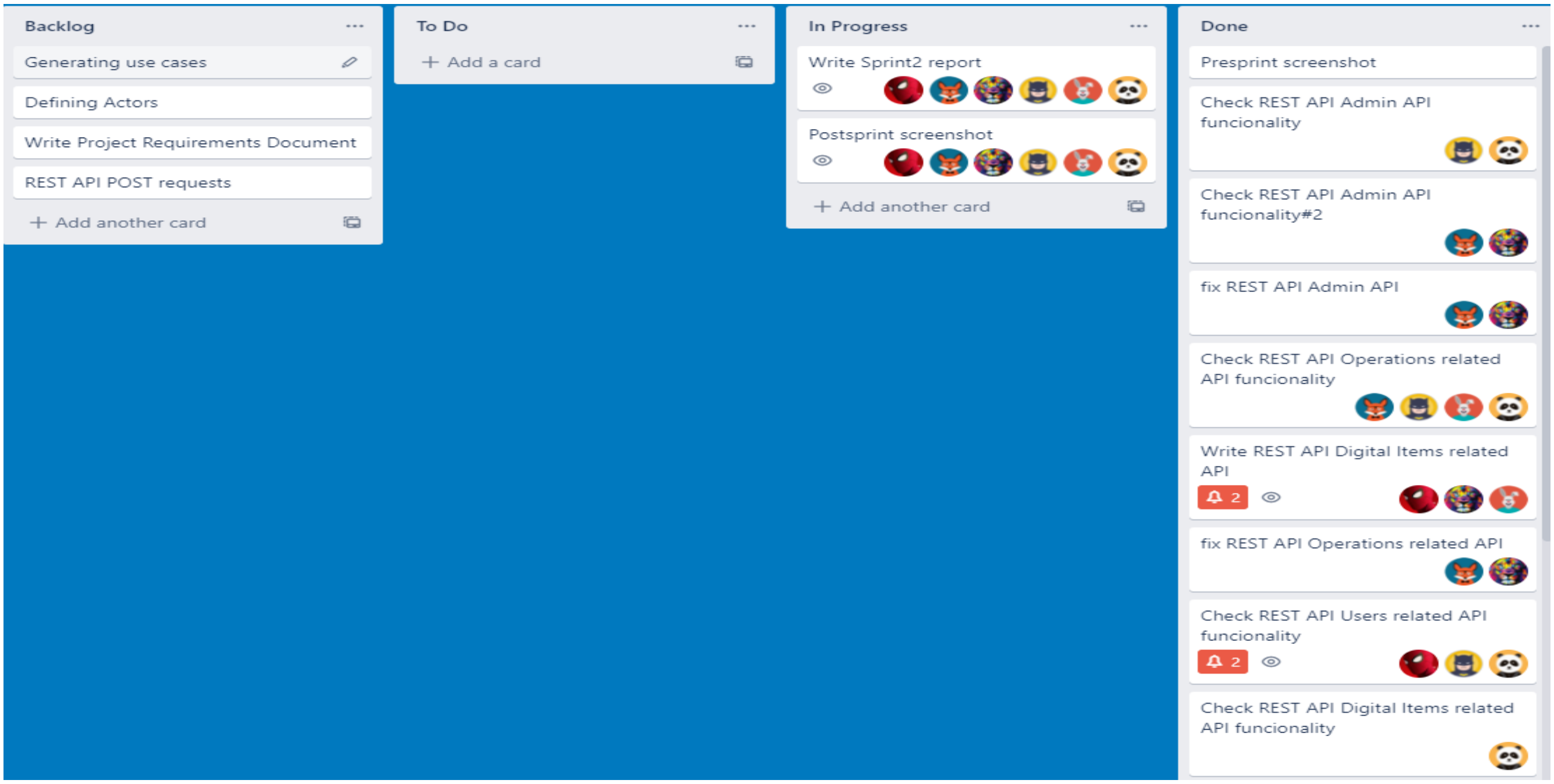
Backlog
- Generating use cases
- Defining Actors
- Write Project Requirements Document
- + Add another card

To Do
- Write REST API interface
- Check REST API is functional and working
- Write Sprint2 report
- Presprint screenshot
- Postsprint screenshot
- + Add another card

In Progress
- + Add a card

Done
- + Add a card

Post-screenshot of Kanban board - 22.03.21



Backlog
- Generating use cases
- Defining Actors
- Write Project Requirements Document
- REST API POST requests
- + Add another card

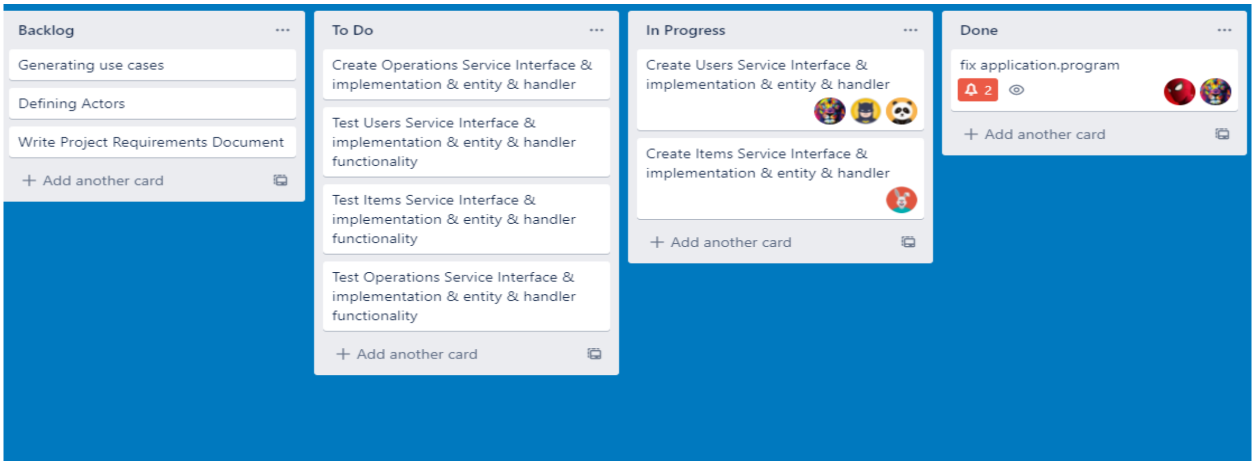
To Do
- + Add a card

In Progress
- Write Sprint2 report
- Postsprint screenshot
- + Add another card

Done
- Presprint screenshot
- Check REST API Admin API funcionality
- Check REST API Admin API funcionality#2
- fix REST API Admin API
- Check REST API Operations related API funcionality
- Write REST API Digital Items related API 2
- fix REST API Operations related API
- Check REST API Users related API funcionality 2
- Check REST API Digital Items related API funcionality

Sprint 3:

Pre-screenshot of Kanban board - 4.04.21



Backlog

- Generating use cases
- Defining Actors
- Write Project Requirements Document

To Do

- Create Operations Service Interface & implementation & entity & handler
- Test Users Service Interface & implementation & entity & handler functionality
- Test Items Service Interface & implementation & entity & handler functionality
- Test Operations Service Interface & implementation & entity & handler functionality

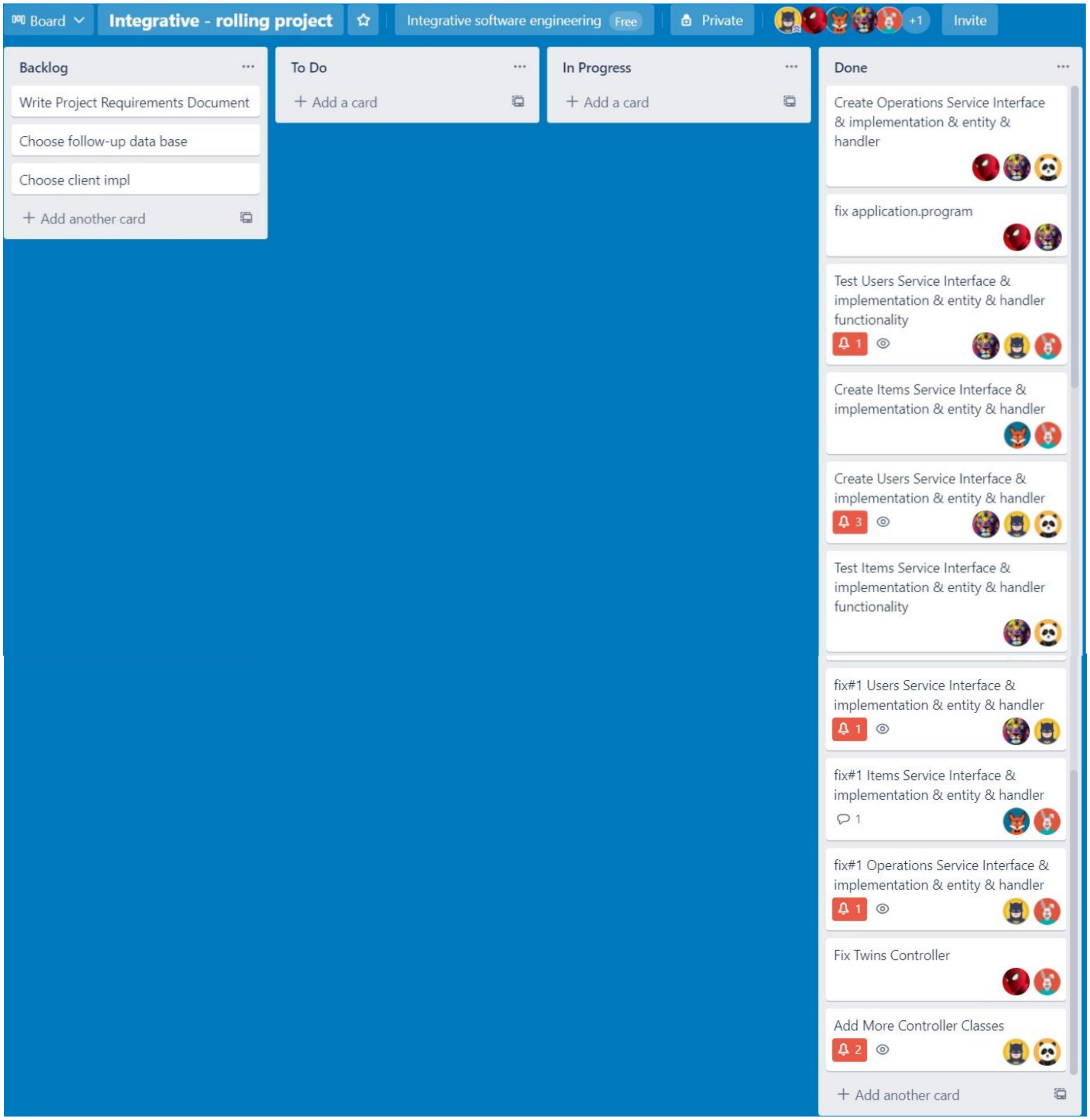
In Progress

- Create Users Service Interface & implementation & entity & handler
- Create Items Service Interface & implementation & entity & handler

Done

- fix application.program

Post-screenshot of Kanban board - 27.04.21


Board
Integrative - rolling project
Integrative software engineering Free
Private
+1
Invite

Backlog

- Write Project Requirements Document
- Choose follow-up data base
- Choose client impl

To Do

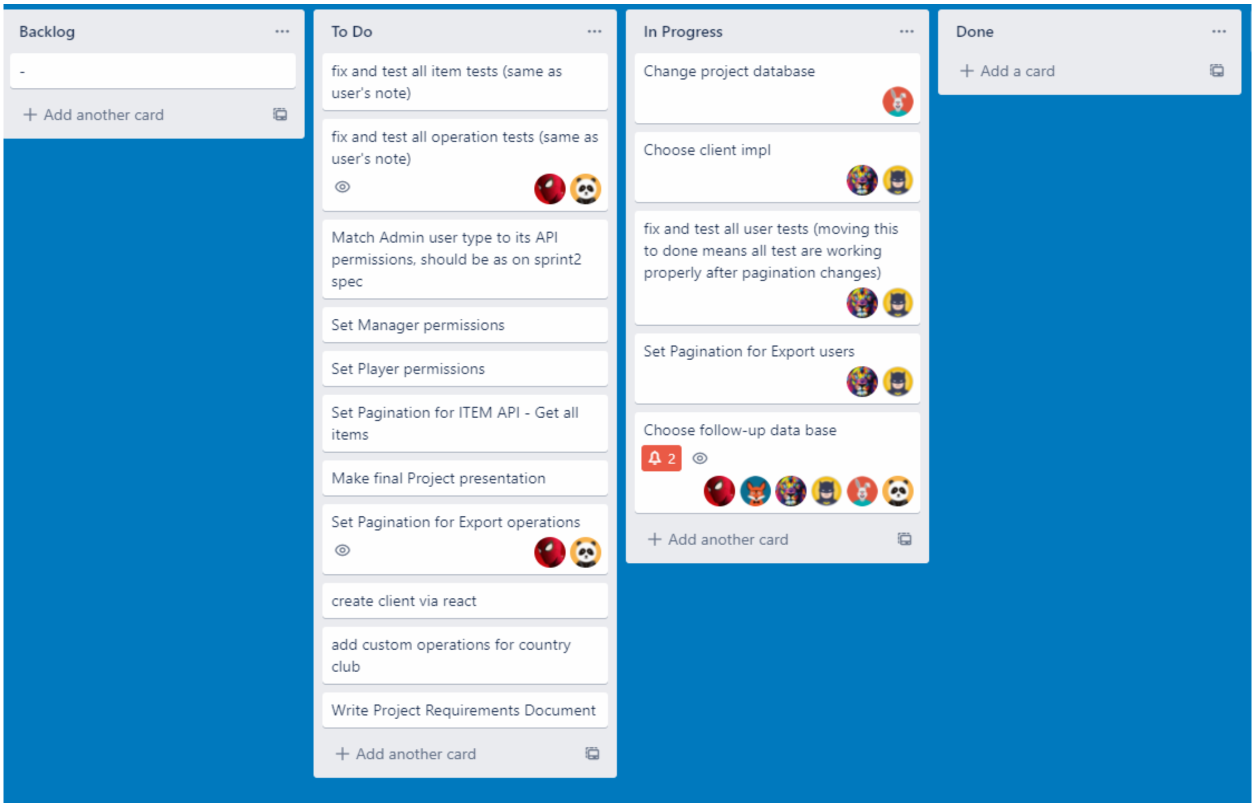
In Progress

Done

- Create Operations Service Interface & implementation & entity & handler
- fix application.program
- Test Users Service Interface & implementation & entity & handler functionality (1)
- Create Items Service Interface & implementation & entity & handler
- Create Users Service Interface & implementation & entity & handler (3)
- Test Items Service Interface & implementation & entity & handler functionality
- fix#1 Users Service Interface & implementation & entity & handler (1)
- fix#1 Items Service Interface & implementation & entity & handler (1)
- fix#1 Operations Service Interface & implementation & entity & handler (1)
- Fix Twins Controller
- Add More Controller Classes (2)

Sprint 4:

Pre-screenshot of Kanban board - 12.05.21



Backlog

- -

To Do

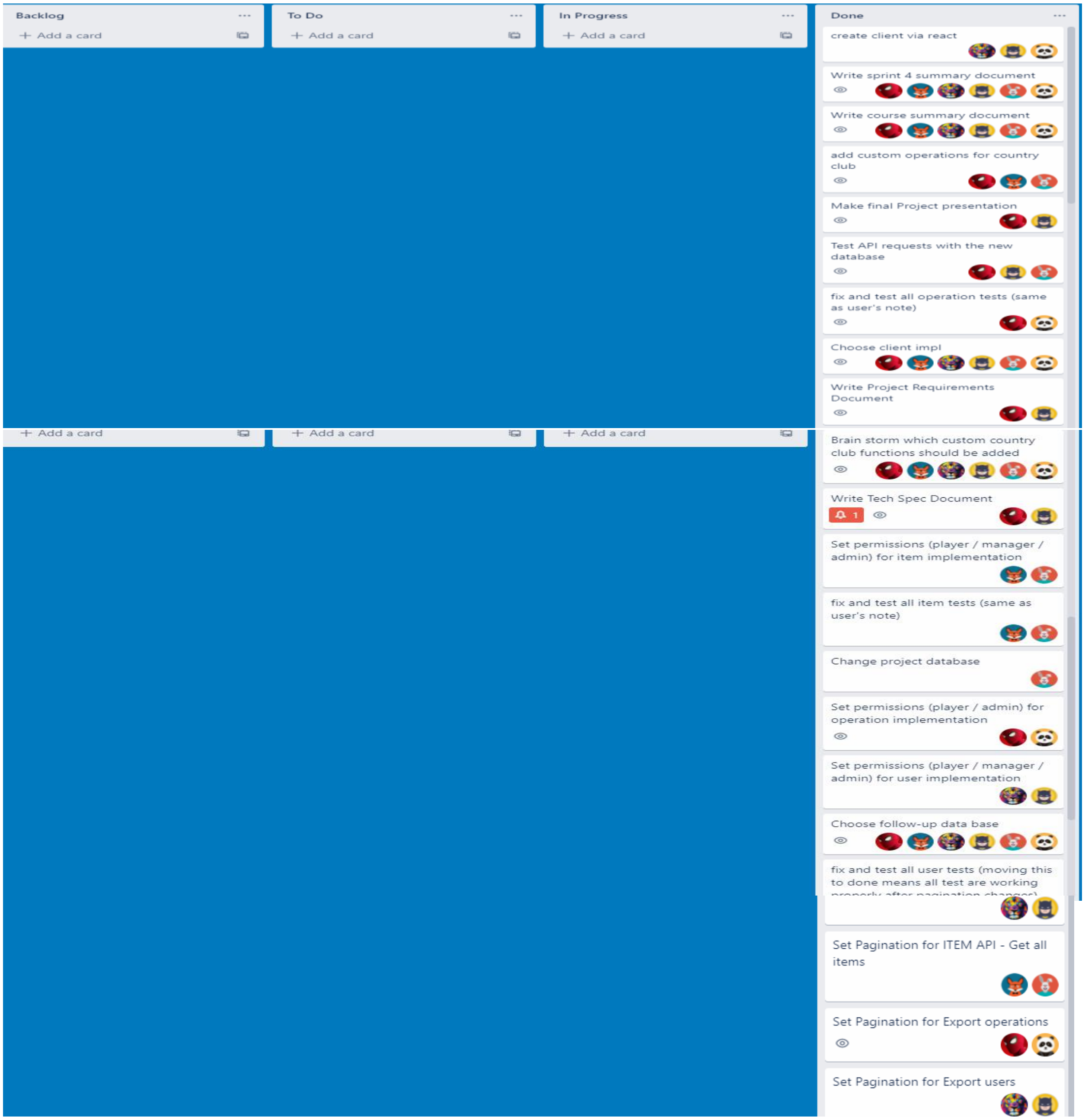
- fix and test all item tests (same as user's note)
- fix and test all operation tests (same as user's note)
- Match Admin user type to its API permissions, should be as on sprint2 spec
- Set Manager permissions
- Set Player permissions
- Set Pagination for ITEM API - Get all items
- Make final Project presentation
- Set Pagination for Export operations
- create client via react
- add custom operations for country club
- Write Project Requirements Document

In Progress

- Change project database
- Choose client impl
- fix and test all user tests (moving this to done means all test are working properly after pagination changes)
- Set Pagination for Export users
- Choose follow-up data base

Done

Pre-screenshot of Kanban board - 01.06.21



Backlog

+ Add a card

To Do

+ Add a card

In Progress

+ Add a card

Done

- create client via react
- Write sprint 4 summary document
- Write course summary document
- add custom operations for country club
- Make final Project presentation
- Test API requests with the new database
- fix and test all operation tests (same as user's note)
- Choose client impl
- Write Project Requirements Document

+ Add a card

+ Add a card

+ Add a card

- Brain storm which custom country club functions should be added
- Write Tech Spec Document 🔔 1
- Set permissions (player / manager / admin) for item implementation
- fix and test all item tests (same as user's note)
- Change project database
- Set permissions (player / admin) for operation implementation
- Set permissions (player / manager / admin) for user implementation
- Choose follow-up data base
- fix and test all user tests (moving this to done means all test are working properly after pagination changes)
- Set Pagination for ITEM API - Get all items
- Set Pagination for Export operations
- Set Pagination for Export users

6.2. Summary of work:

- **What went well for the team during the semester and should be kept for future projects:**
    - We communicated well during all the project
    - We always helped each other
    - We felt comfortable to ask for help from each other
    - We divide the work after getting feedback, which improved our time management greatly.

- **What will you improve for other projects you will work in:**
    - Prioritize dividing the work to small groups with specific workload.
    - Get familiarized with the working environments we will use throughout the projects.

- **What did you enjoy the most during this project?**
    - This is the first time we faced creating an entire system from scratch – backend and frontend, which was interesting and taught us the integration between server and client.

- **What would you do differently if you started the project today?**
    - We would start working on a client at an earlier phase to secure us more time to improve our system functionalities.

- **How did working from home affect your teamwork?**
    - We still managed to work well from home while using Zoom and Discord.
    - It was easier to work as smaller groups and from a far.
    - Using Trello helped us manage the assignments better and giving opportunities to all group members.

7.Integrative software engineering rolling project Final Sprint

Project: Country Club - Digital Twin

Due date: 02.06.21

Roles:

Daniel Aizenband 205455520 – Team leader, Technical Writer, Team

Nadv Rud 312172687 - DevOPS, Team

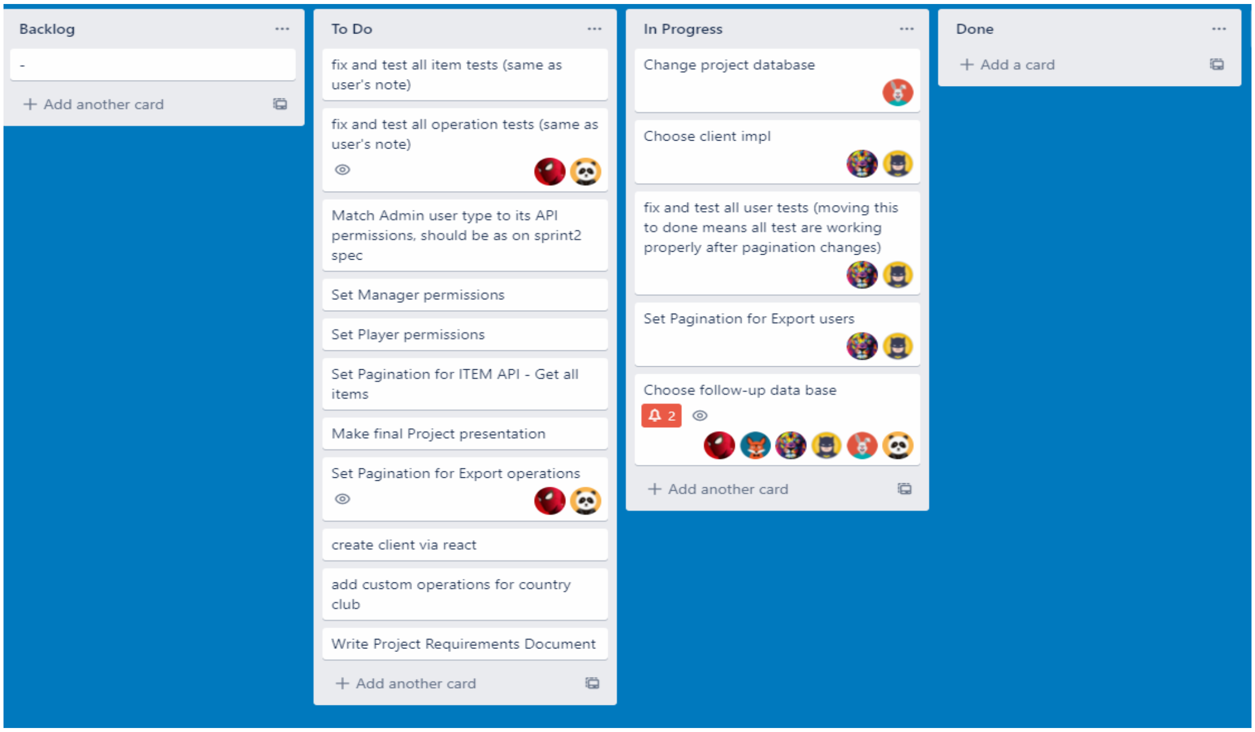
Lior Perlman 203311808 – Scrum Master, DBA, Team

Imry Ashur 203958228 – QA Engineer, Team

Noam Marko 205684921 – UIX, Team

Lior Toledano 204326532 - Product Owner, Team

Pre-screenshot of Kanban board - 12.05.21



Backlog

- -

To Do

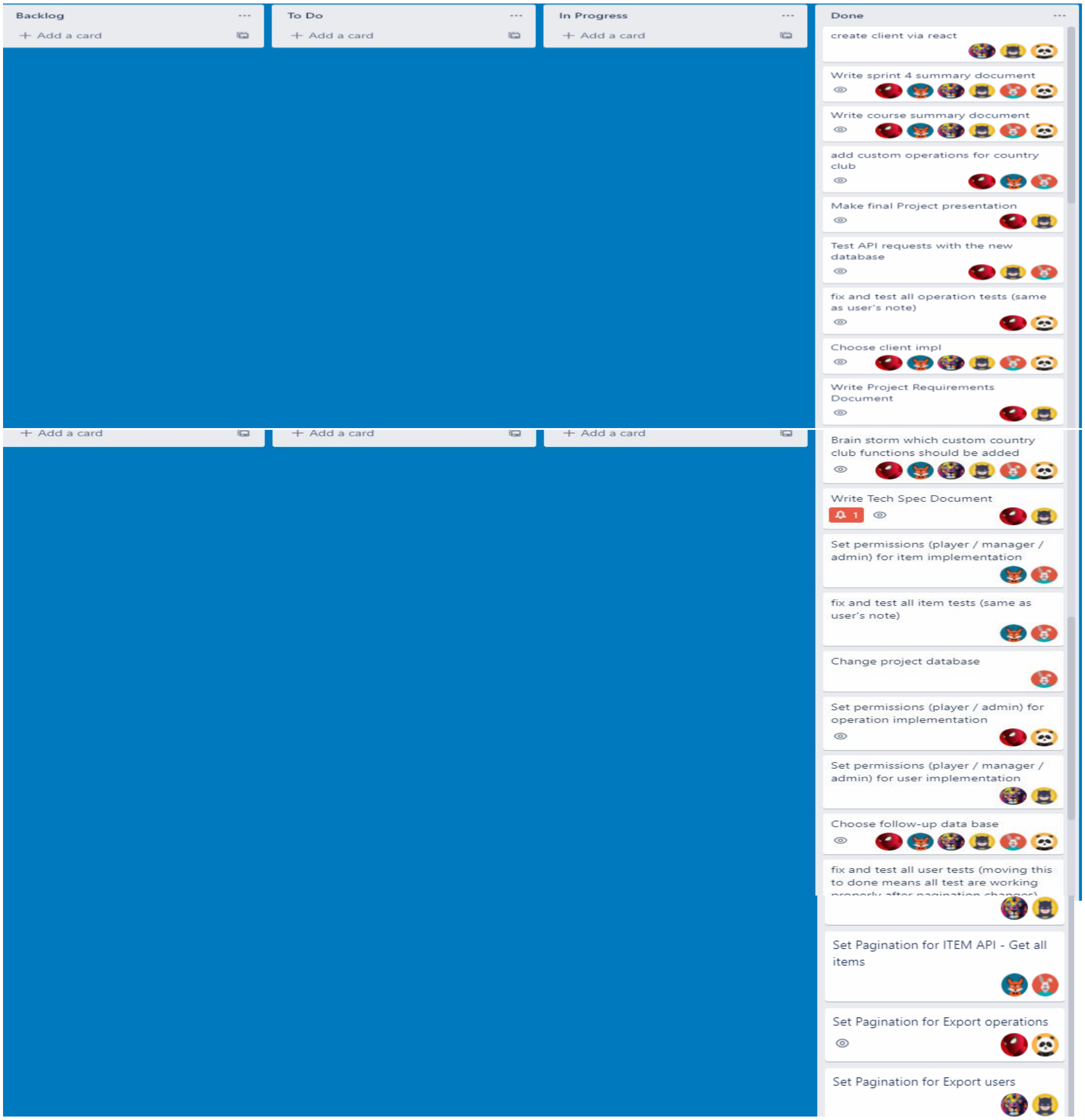
- fix and test all item tests (same as user's note)
- fix and test all operation tests (same as user's note)
- Match Admin user type to its API permissions, should be as on sprint2 spec
- Set Manager permissions
- Set Player permissions
- Set Pagination for ITEM API - Get all items
- Make final Project presentation
- Set Pagination for Export operations
- create client via react
- add custom operations for country club
- Write Project Requirements Document

In Progress

- Change project database
- Choose client impl
- fix and test all user tests (moving this to done means all test are working properly after pagination changes)
- Set Pagination for Export users
- Choose follow-up data base

Done

Pre-screenshot of Kanban board - 01.06.21



- create client via react
- Write sprint 4 summary document
- Write course summary document
- add custom operations for country club
- Make final Project presentation
- Test API requests with the new database
- fix and test all operation tests (same as user's note)
- Choose client impl
- Write Project Requirements Document
- Brain storm which custom country club functions should be added
- Write Tech Spec Document
- Set permissions (player / manager / admin) for item implementation
- fix and test all item tests (same as user's note)
- Change project database
- Set permissions (player / admin) for operation implementation
- Set permissions (player / manager / admin) for user implementation
- Choose follow-up data base
- fix and test all user tests (moving this to done means all test are working properly after pagination changes)
- Set Pagination for ITEM API - Get all items
- Set Pagination for Export operations
- Set Pagination for Export users

General summary of work:

- What went well for the team and should be continued on the next phases of work:
    - Was easy to schedule a zoom work meeting and everyone arrived on time.
    - Divided the work between 3 teams of 2, which allowed us to work simultaneously.
    - We tried to collaborate with other teams to understand how to improve our work.
    - We switched groups to create a client, which allowed team members to experience new parts of the system.

- What should be improved in our teamwork:
    - We should document more during each team's work - especially when working on the same piece of code.

- What problems did the team encounter through this phase of work:
    - We started creating a client and most of us did not have the knowledge to build it
        - Solution: team up Noam Marko and Lior Toledano to Lior Perlman to consult and help him.

- Why did we not complete all planned work:
    - We planned to add more unique functionalities to our system but decided to tone it down to work on the client.
        - We decided that we wanted to able to presents the final sprint with a client rather than complicate the system.