I. Introduction

Employees needed a tool to easily connect, collaborate, and do more together in an increasingly distributed company.

Many people may worked in different countries and timezones, complicating collaboration—especially hard if there wasn't a single source of information for everyone to go to.

Coordinating work across teams is chaotic—especially if you have to dig through emails or messaging apps to find information. Without a system for planning, organizing, and executing work, confusion and chaos can grow. That's where work management can help.

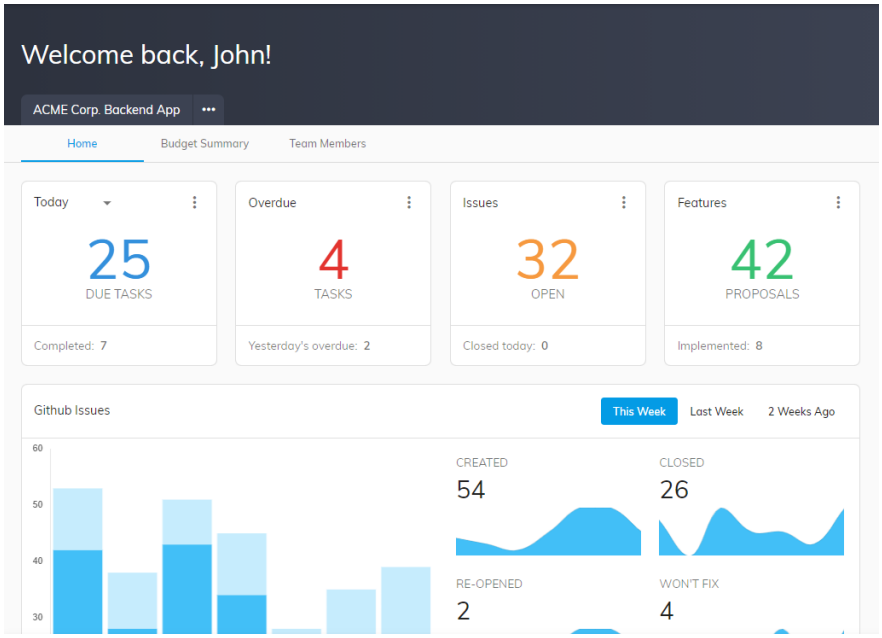
Our's goal was to ensure a smooth, organizanization-wide digital transformation for how employees all work together. A critical piece of this digital transformation was implementing a work management platform.

Work management is a systematic approach to orchestrating an organization's workflows—be it a project, an ongoing process, or routine tasks—to provide the clarity teams need in order to hit their goals faster. It's about coordinating people and work across all levels of an organization to ensure that everyone has the information they need to accomplish the work that matters most.

II. Project feature and function

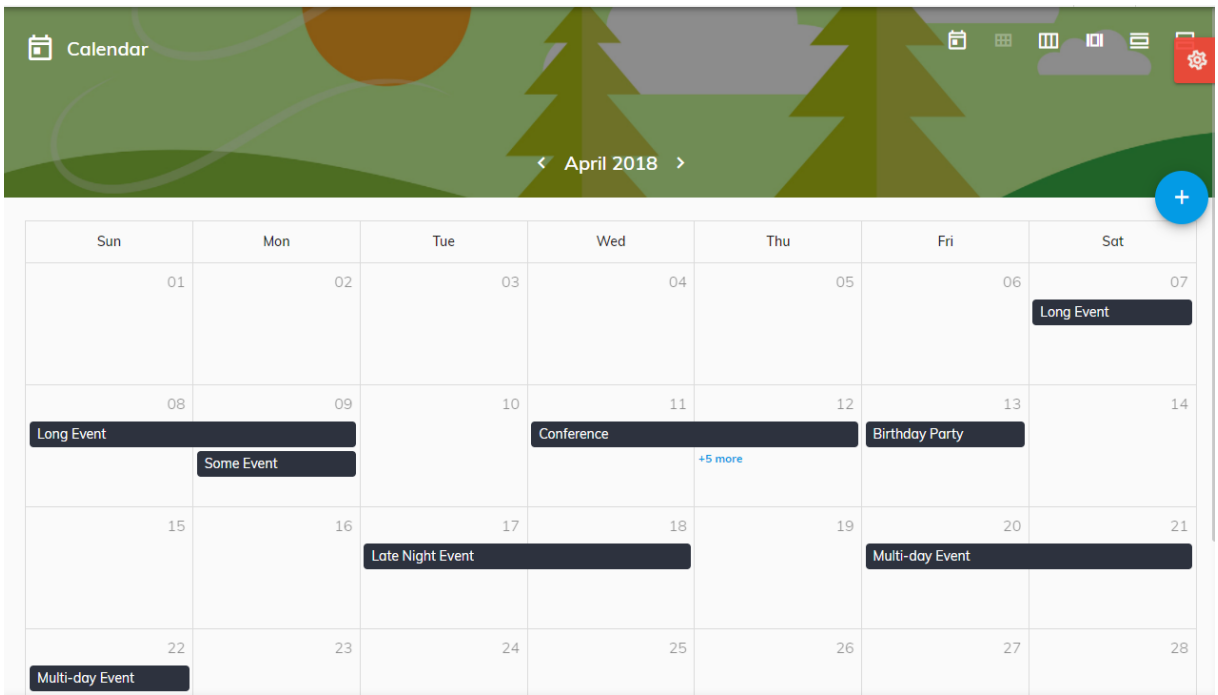
2.1. Dashboard page

View your team’s projects and keep track with statistic



2.2. Calendar

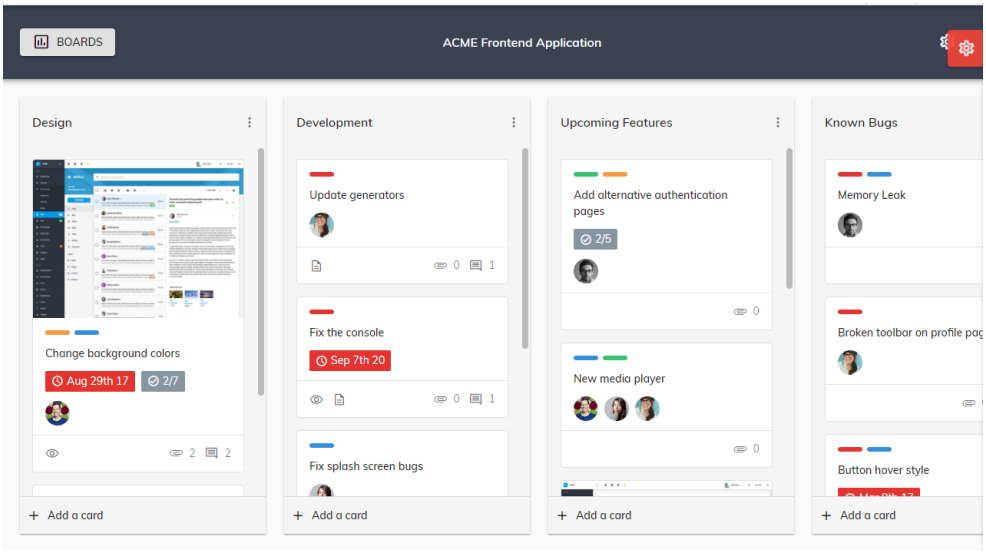
View your team's work on one shared calendar



2.3. Board

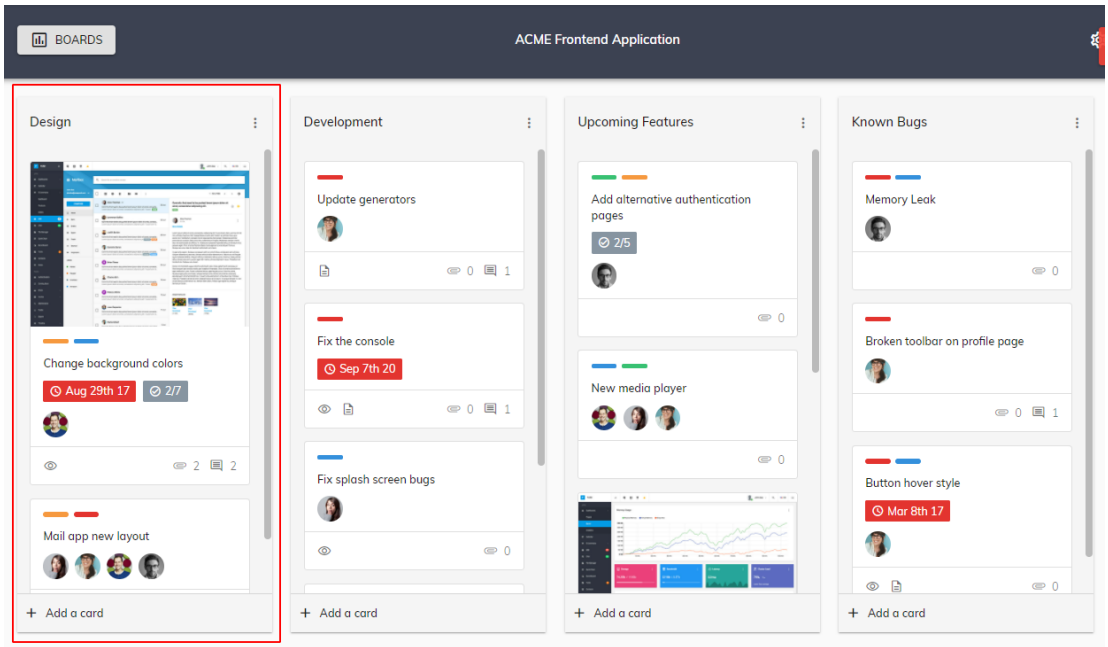
Organize your work into shared projects as lists or Kanban boards for your initiatives, meetings, and programs.

Give teams clarity and context on work in one place. Share key resources, status updates, and more.



2.4. List

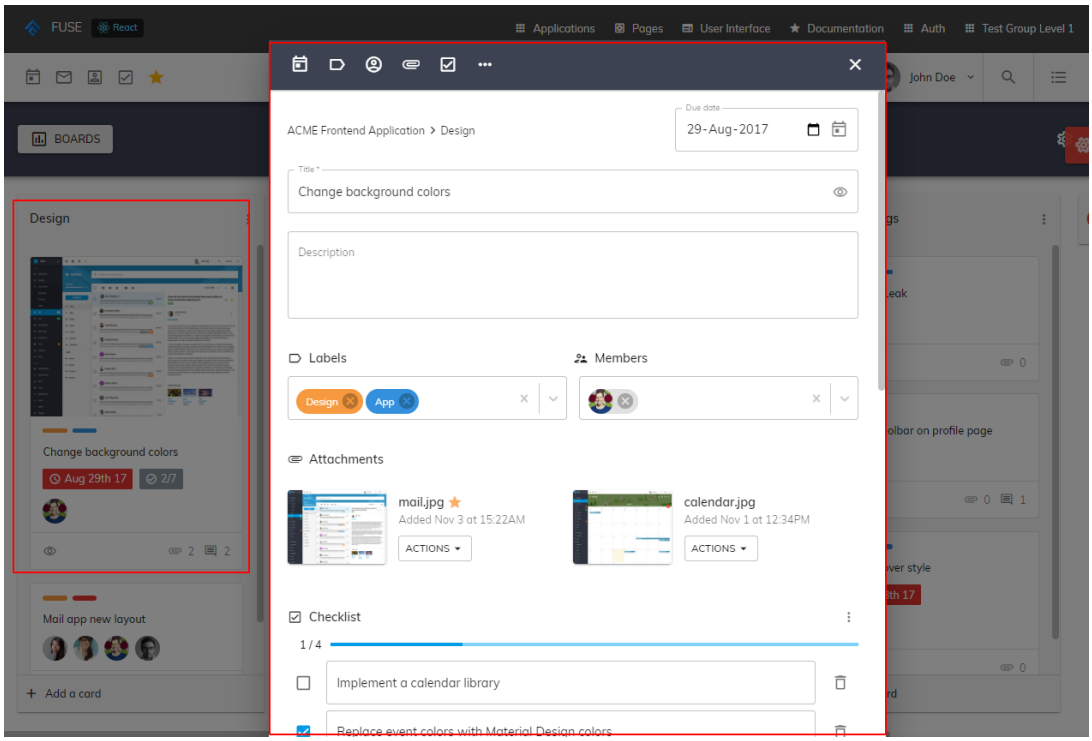
Group tasks into lists in any project to match workflows, break up types of work, and keep tasks organized.



2.5. Tasks

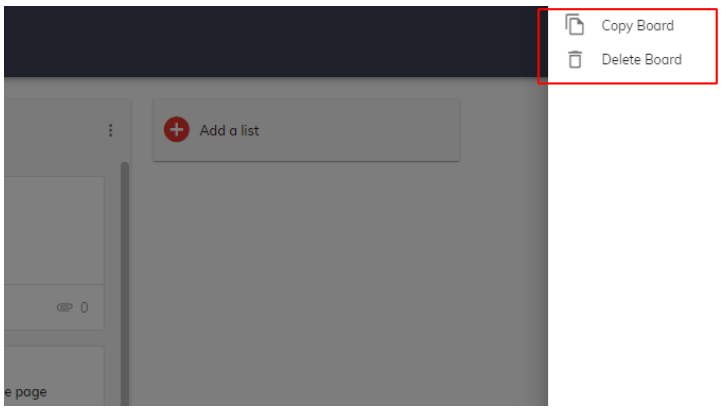
Break work into manageable pieces for you and your team. In this module you can:

- Break up a task into smaller parts
- Show additional steps to complete an overall task.
- Task assignees: Give tasks a clear owner, so everyone knows who's responsible.
- Add a comment or attach a file
- Add a due date or time
- Any many more…



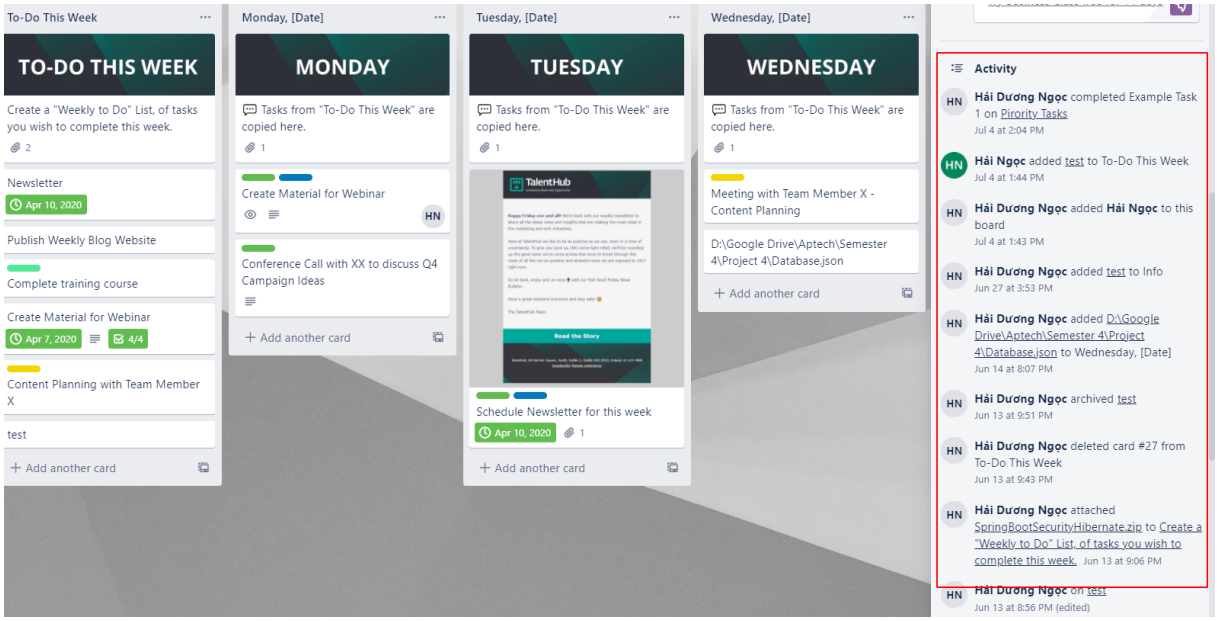
2.6. Copy board

Eliminate time spent recreating your team's common workflows.



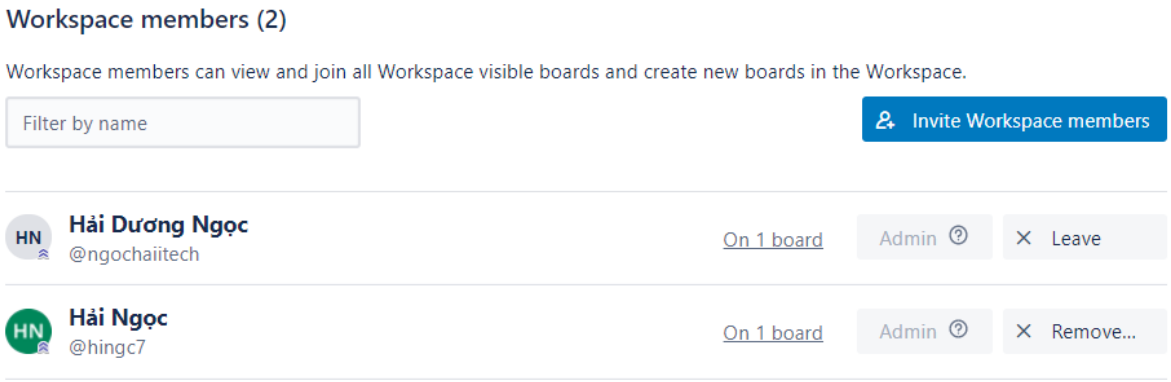
2.7. Board activity

Control all activity on project to keep an eye from any changes and updates

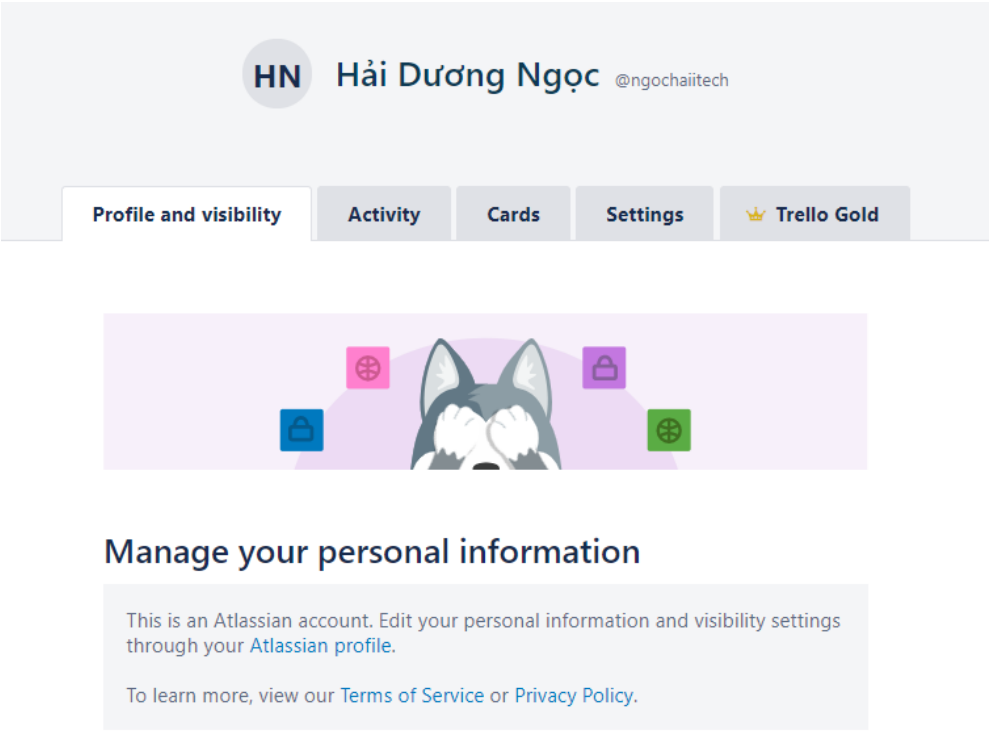


2.8. Members

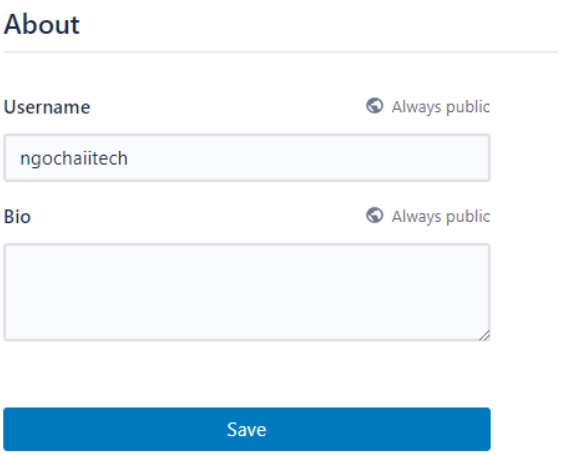
See your teams member or project member



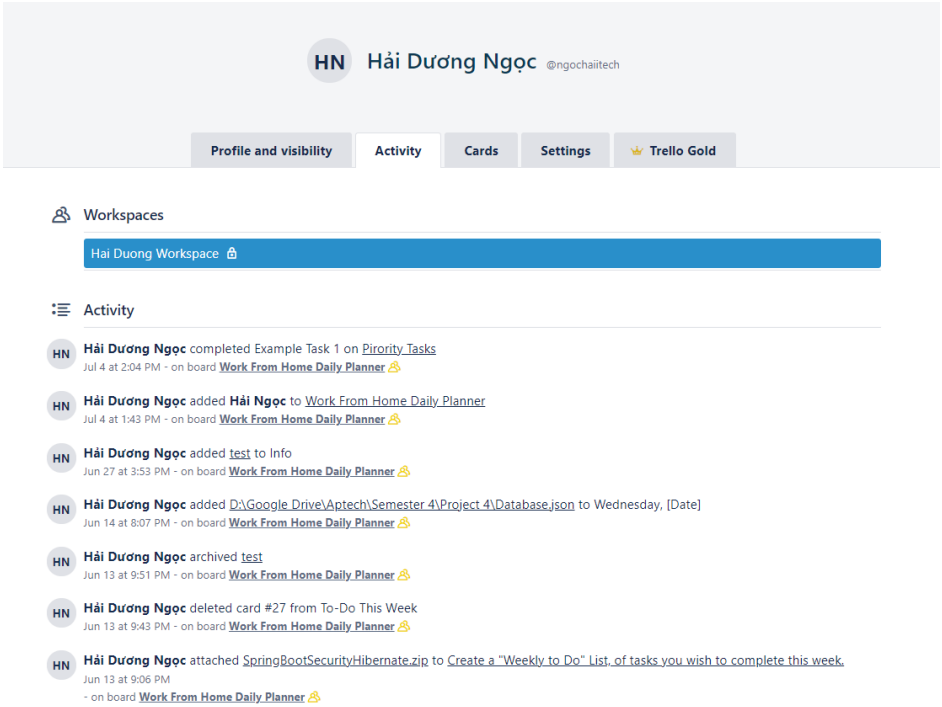
2.9. Personal page



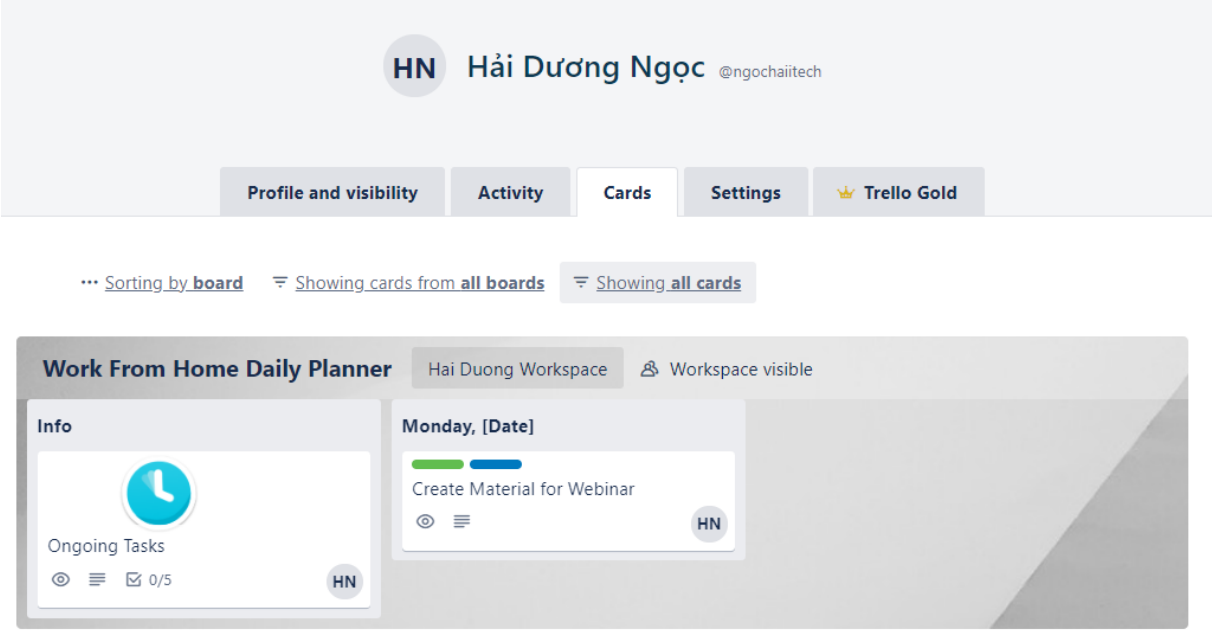
- Manage your personal information



- See your activities



- See which project you are in and cards you created/assigned to you



III. Database

3.1. Introduction

The database is built in mySQL but its has archetype like NoSQL, most of object and relational will be nested in a stringified JSON.

We choose is archetype to take some advantage of NoSQL

- **Flexibility**: this approach typically offer flexible diagrams that make development faster and more repeatable. The flexible data model makes NoSQL databases the ideal choice for unstructured or incompletely structured data.
- **Scalability**: databases are typically designed to scale using clusters of distributed hardware rather than scale up with expensive and powerful server additions. Some cloud service providers treat these activities non-publicly as a fully managed service.
- **High performance**: this approach are optimized for specific data models and access patterns which provide greater performance gains than trying to achieve the same level of functionality with a database relationship.
- **Extremely practical**: this databases provide extremely practical APIs and data types built specifically for each respective data model.

3.2. Table definition

3.2.1. User

User						
Name	Data Type	Primary Key	Identity	Not null	Default value	Description
id	**int**	X	X	X		Id of User
email	varchar(50)			X		Email of User
password	varchar(50)			X		Password of User
username	varchar(50)			X		Username of User
fullName	varchar(50)			X		Name of User
contact	text					Stringified JSON contains multiple user
info	text					Stringified JSON contains multiple data

User's detailed property in json format:

```
"user": {
        "id": "",
        "username": "",
        "password": "",
        "email": "",
        "fullName": "",
        "contact": ["userId1", "userId2"],
        "info": {
            "avatar": "",
            "phoneNumber": ""
        }
    }
```

3.2.1. Board

Board						
Name	Data Type	Primary Key	Identity	Not null	Default value	Description
id	**int**	X	X	X		Id of User
name	varchar(50)			X		Email of User
members	varchar(50)			X		Password of User
lists	varchar(50)			X		Username of User
info	text			X		Stringified JSON contains multiple data
activities	text			X		Stringified JSON contains multiple data
labels	text			X		Stringified JSON contains multiple data
createdAt	datetime			X		Date created
updatedAt	datetime			X		Date updated

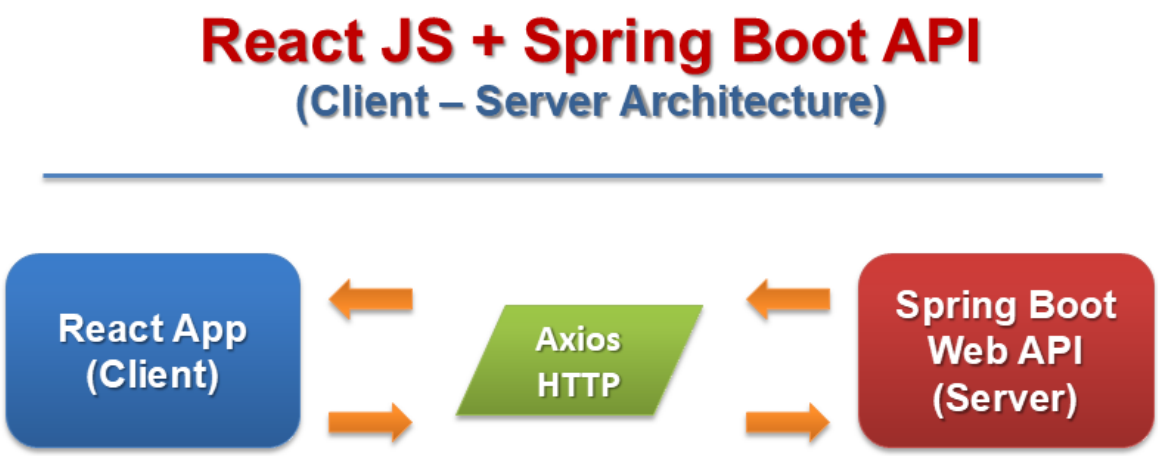
Board's detailed property in json format:

```
"board":{
        "id": "",
        "name": "",
        "members": [
            {
                "userId": "",
                "role": ""
            }
        ],
        "lists":[
            {
                "id": "",
                "name": "",
                "cards": [
                    {
                        "id": "",
                        "name": "",
                        "content": "",
                        "members":[],
                        "labels": [],
                        "checklist":[
                                {
                                "id": "",
                                "name": "",
                                "done": ""
                            }
                        ],
                        "dueDate": "",
                        "comment":[
                            {
                                "userId": "",
                                "content": "",
                                "date": ""
                            }
                        ],
                        "coverImage":""
                    }
                ]
            }
        ],
        "info": {
            "backgroundImage": "",
            "author": "authorId",
            "type": "personalBoard/template"

        },
        "activities": [
            {
                "userId": "",
                "type": [
                    "joined/removed from board",
```

```
                "create/detele list",
                "create/delete card",
                "comment",
                "add flags",
                "set dueDate",
                "add labels",
                "..."
            ],
            "cardId": "",
            "listId": "",
            "date": ""
        }
    ],
    "labels": [
        {
          "id": "",
          "name": "",
          "color": ""
        }
    ],
    "author": "authorId",
    "type": "personalBoard/template",
    "createdAt": "",
    "updatedAt": ""
}
```

IV. Technology



1. Presentation Tier:

Is the tier in which the users interact with application. In React presentation Tier contents Presentational and container components.

Technology: ReactJS, Redux, Axios,…

2. Business Logic Tier:

Is mainly working as the bridge between Data Tier and Presentation Tier. All the Data passes through the Business Tier before passing to the Presentation Tier.

Technology: Spring Boot Web API.

3. Data Access Tier:

Is basically the server which stores all the application's data .Data tier contents Database Tables, Database Views and other means of storing Application Data .
Technology: My SQL, Hibernate.

V. Site map

1. Dashboard
2. Features
3. Boards
   - 3.1. Template
   - 3.2. Your boards
4. Profile
   - 4.1. Your profile
   - 4.2. Activity
   - 4.3. Cards
   - 4.4. Contacts
5. About us