



Goals

Demo

Tools

Code

Findings

Limits

Design

Citations

HomeHarmony

Take the Stress Out of Sharing Space - Record, Track, and Celebrate Chore Fairness!

Team Members

Kaiying Yu, Qinyuan shen, Jianing Wang

Agenda

Goals

Code

Design

Findings

Demo

Limits

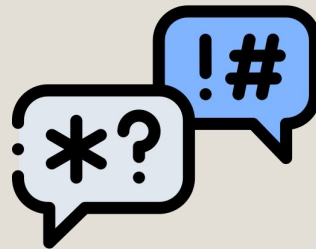
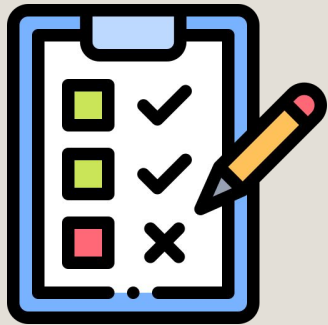
Tools

Citations



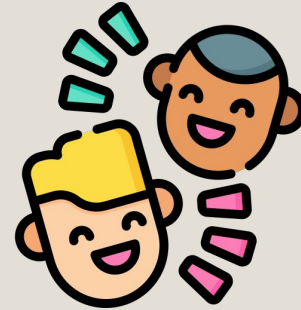
Pain points

- No clear records of who did what and when
- The status of chore is not clear
- More arguments than collaboration
- Chores are NO FUN but we ALL HAVE TO do it



Goals

- Maintain a clear record of who did what and when
- Make chore status crystal clear
- Encourage collaboration and healthy competition
- Make doing chores FUN for all users

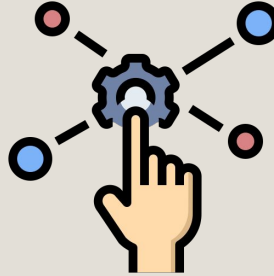


Record Keeping

Transparency

Gamification

Collaboration



Chore

User

Space

**Main
Classes**

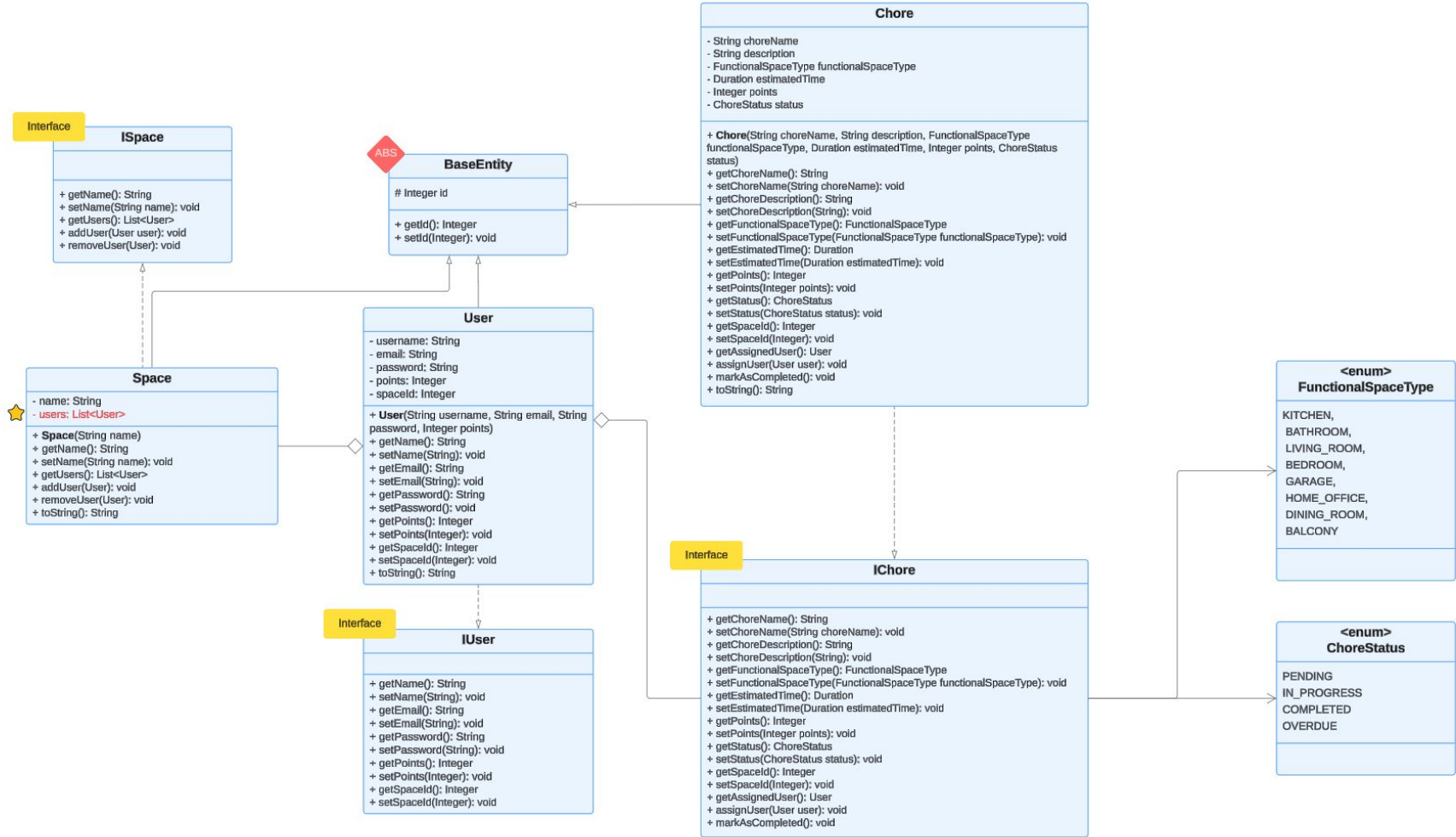
Shared space

Public chore status

Different points for chore

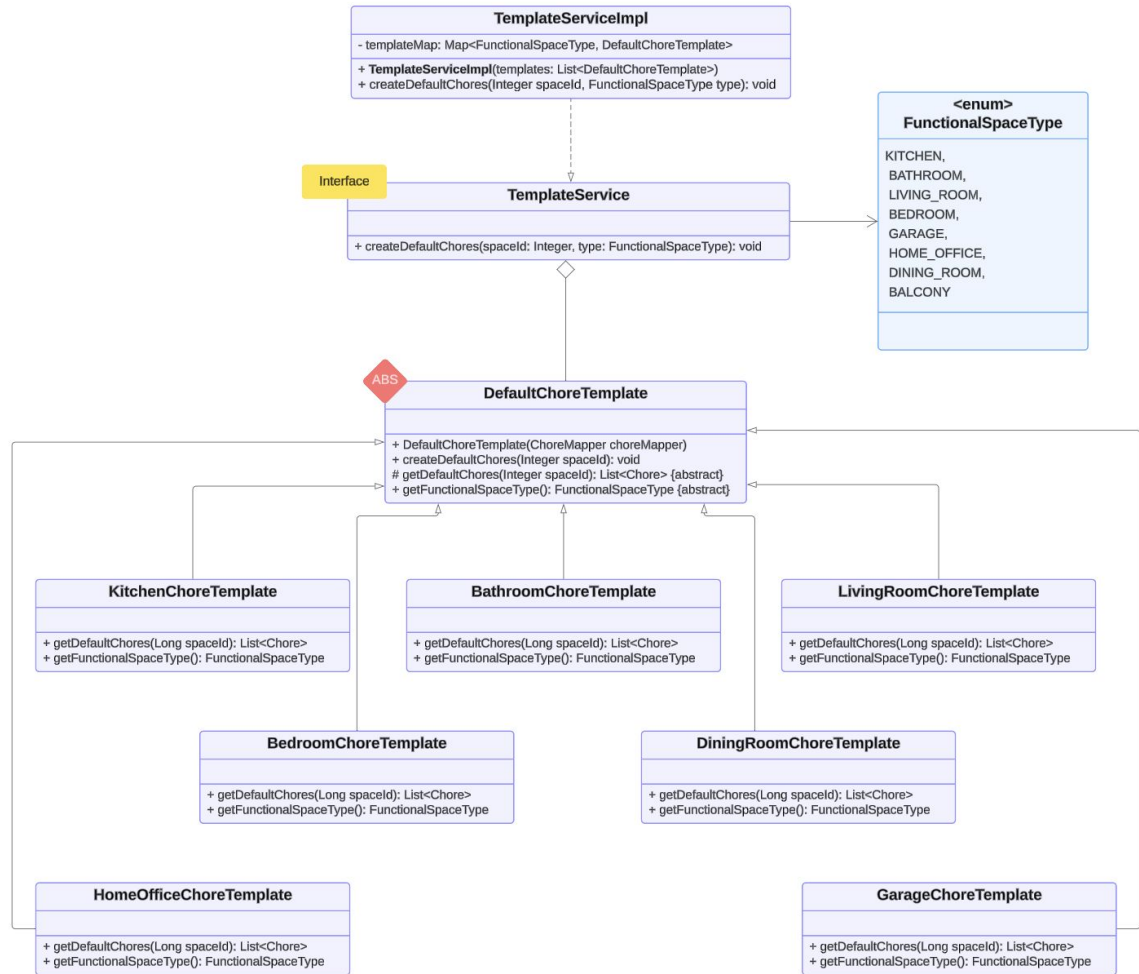
Public roommates' progress

...



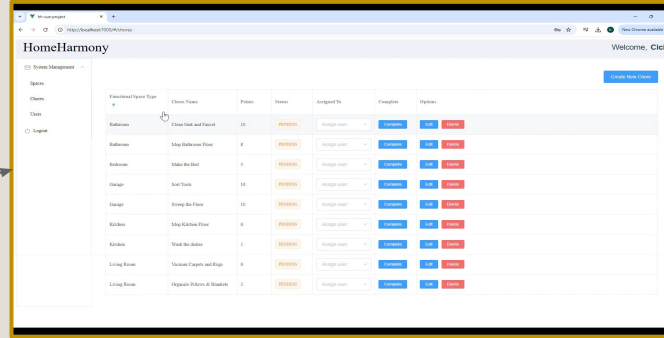
UML

TemplateService

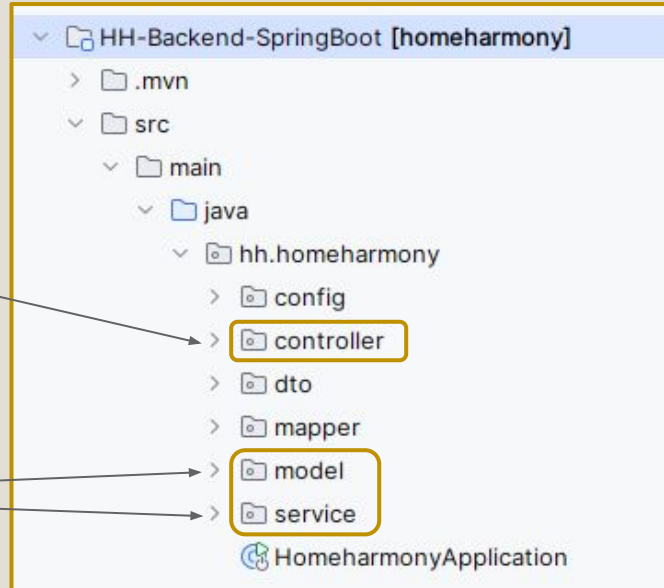


OOD Big Idea-MVC

View



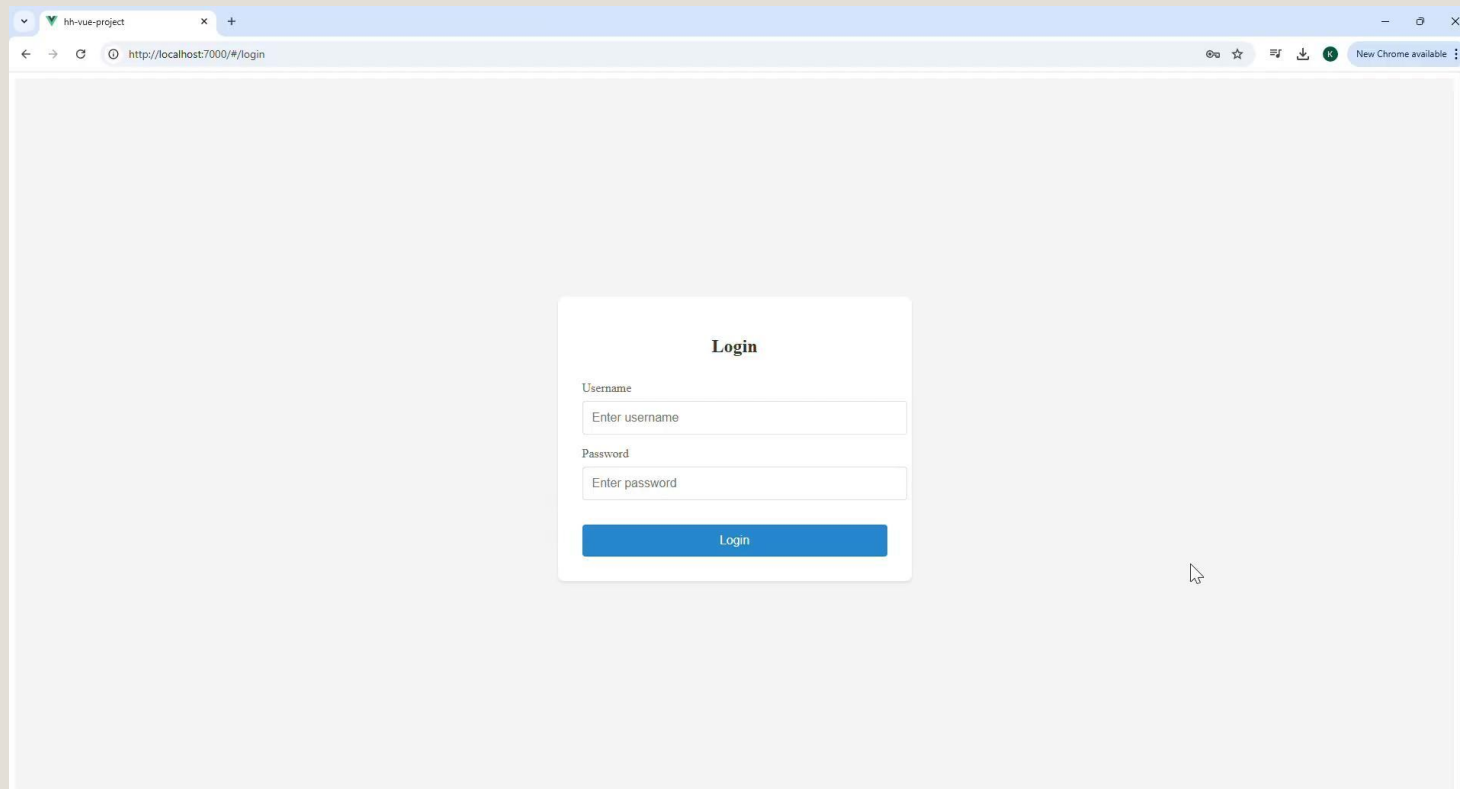
Controller



Model

Let's see the demo
on localhost

Or watch a recorded video



Let's recap

1. Create/Join a space
2. Create/Edit/Delete chores
3. Assign a user to a chore
4. Complete a chore
5. Points added to users'
6. Display who scores the highest

Tools

Our Tech Stack Trio



Code Demonstration

Bathroom	Clean Sink and Faucet	10	COMPLETED	Alice	Completed	Edit	Delete
Bathroom	Mop Bathroom Floor	8	COMPLETED	Bob	Completed	Edit	Delete
Bedroom	Make the Bed	3	COMPLETED	Cici	Completed	Edit	Delete

```
@Override 3 usages Kaiying-Yu
public void markAsCompleted() {
    if (this.status == ChoreStatus.IN_PROGRESS) {
        this.status = ChoreStatus.COMPLETED;
    }
}
```

Chore

```
@Override 4 usages Kaiying-Yu
public Chore completeChore(Integer choreId) throws IllegalArgumentException {
    Chore chore = choreMapper.selectById(choreId);
    if (chore == null) {
        throw new IllegalArgumentException("Chore not found");
    }
    chore.markAsCompleted();
    choreMapper.update(chore);
    return chore;
}
```

ChoreServiceImpl

Code Demonstration

Space Members

Alice 

28 Points

Assigned Chores

Chore Name	Functional Space	Points
Wash the dishes	KITCHEN	5
Clean Sink and Faucet	BATHROOM	10
Mop Bathroom Floor	BATHROOM	8
Sort Tools	GARAGE	10

```
@Override 4 usages Kaiying-Yu
public void addPoints(Integer userId, Integer points) throws IllegalArgumentException {
    if (userId == null) {
        throw new IllegalArgumentException("User ID cannot be null");
    }
    if (points == null || points < 0) {
        throw new IllegalArgumentException("Points must be a positive number");
    }

    User user = userMapper.findUserById(userId);
    if (user == null) {
        throw new IllegalArgumentException("User not found with ID: " + userId);
    }

    // Calculate and update points
    Integer currentPoints = user.getPoints() != null ? user.getPoints() : 0;
    user.setPoints(currentPoints + points);
    userMapper.updatePoints(user);
}
```

UserServiceImpl

Findings and Lesson Learned

Circular Reference/Bidirectional Relationship

Example: Space and User

- Data Inconsistency Risks
- Multiple Sources of Truth

Findings and Lesson Learned

Hard:

- Learning to use Java Spring Boot framework, VUE and mySQL integratively
- Merging front end and back end
- Database Design

Easy:

- Coming up with specific chores for each functional space
- Connecting to mySQL database using Spring Boot

Limitations and Future Extensions

Limitations:

1. Accessibility for mobile app vs website

Future Extensions:

1. Localhost vs deploy to the internet
2. Recurring chores (daily, weekly or monthly)

Citations

Elisabeth Freeman, Eric Freeman, Bert Bates, and Kathy Sierra. 2004. Head First Design Patterns. O' Reilly & Associates, Inc.

<https://docs.spring.io/spring-boot/tutorial/first-application/index.html>

<https://docs.spring.io/spring-boot/reference/web/servlet.html>

Q & A