

# MY FITNESS PAL

Fitness Management  
System



# OUR TEAM

## CHIENYU CHEN

chen.chieny@northeastern.edu

## YIHAN XU

xu.yihan@northeastern.edu

## ZHENYU WANG

wang.zhenyu1@northeastern.edu

## ZILIANG ZHAO

zhao.zili@northeastern.edu

## JINGZHENG SONG

song.jingz@northeastern.edu



# OUTLINE

01

## PRODUCT

INTRO / FEATURES / DEMO

02

## TECHNIQUE

FRONT / BACK / DB

03

## METHODOLOGY

COMM / DOCS / TOOLS

04

## Q&A

Feel free to ask us anything!

01

# PRODUCT

Take control of your life from this moment.

# INTRODUCTION

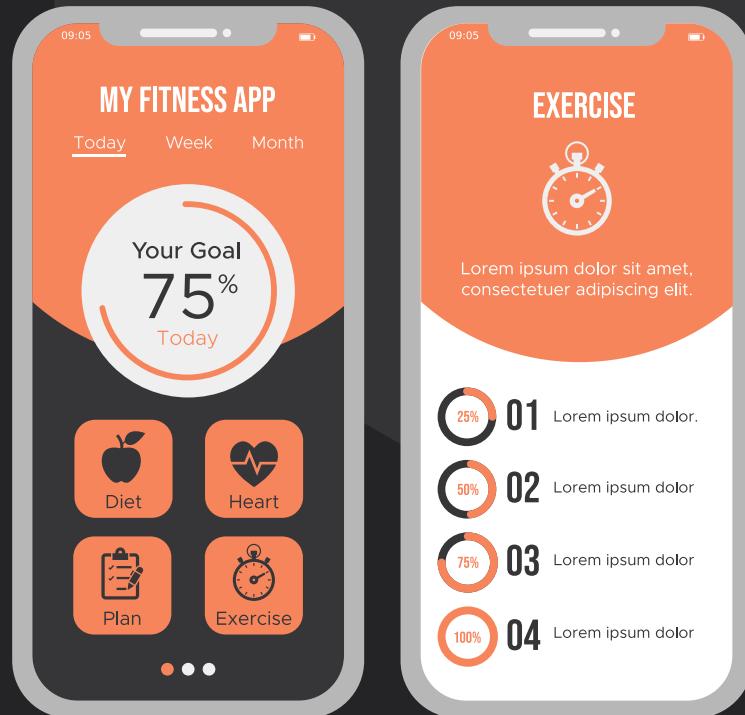
- A fitness management system that could provide foods and sports information for members.
- My Fitness Pal builds on **Vue.js** front-end framework. Servers are based on **Java SpringBoot** and **MySQL**.



# PRODUCT FEATURES

## THE APP DESIGN

- User Management
- Sports Information
- Calories Consumption Table
- Recipe Search
- To be continued...



# DEMO



02

## TECHNIQUE

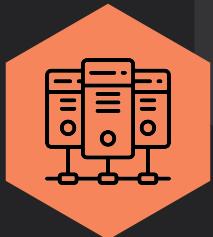
Full-stack Web Development

# STACK



## VUE

The lightweight progressive front-end JavaScript framework.



## SPRING BOOT

Spring Boot offers a faster way to build Spring applications.



## MYSQL

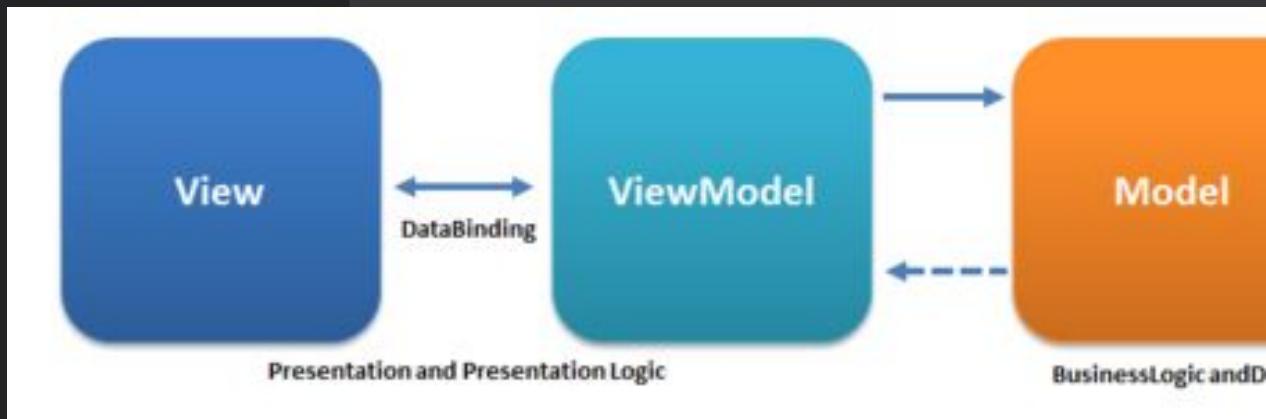
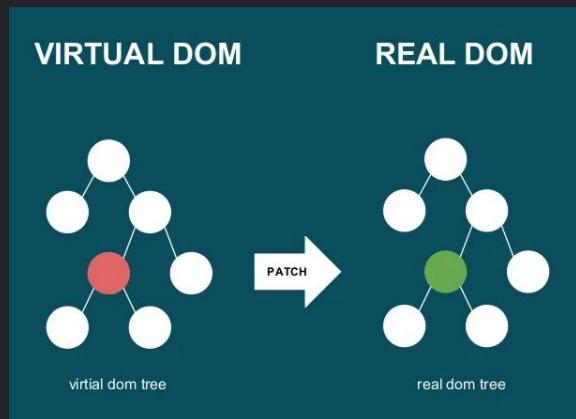
Fast, versatile and popular RDBMS.

# What is VUE.js?



Vue.js is an open-source lightweight model–view–viewmodel front-end JavaScript framework for building user interfaces and single-page applications.

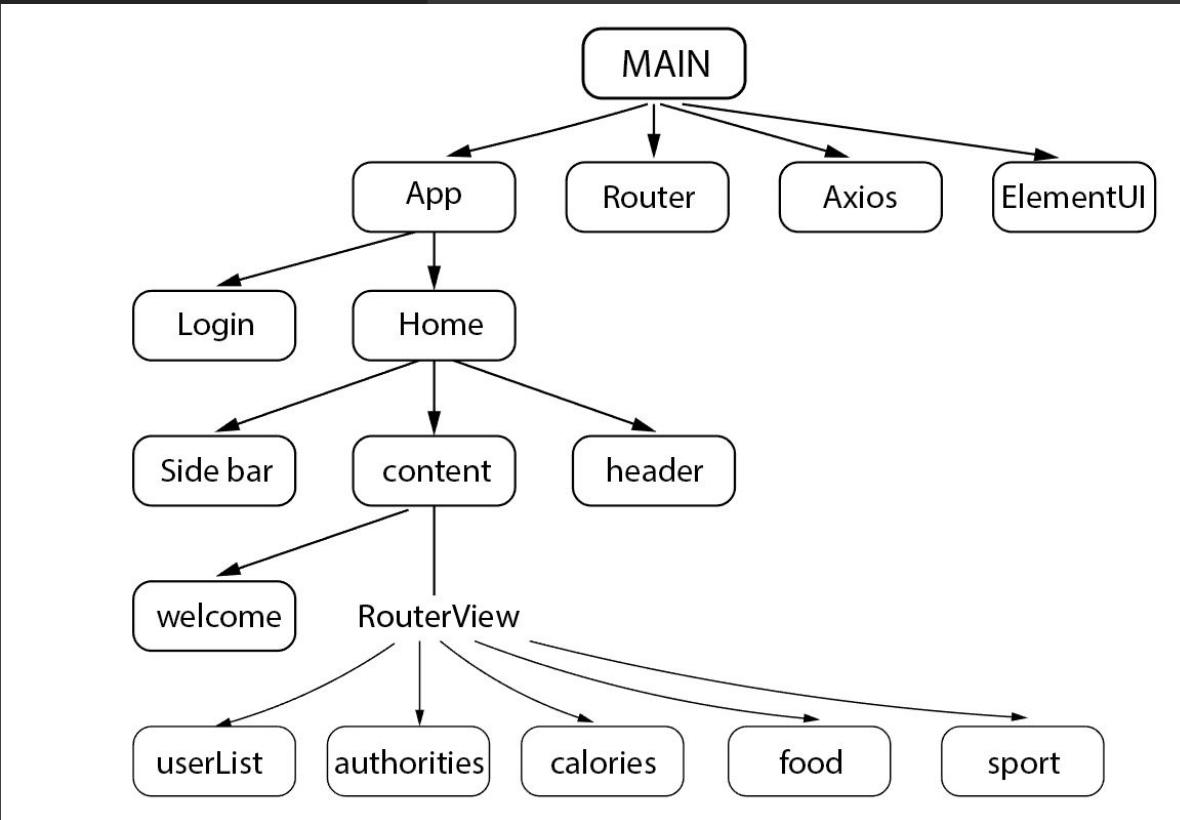
# Benefit of using VUE.js



1. It's lightweight (20 kb), very easy to install, perfect for small-scale projects.
2. Code less and achieve more.
3. Uses MVVM (model–view–viewmodel) which allows two-way communication.
4. Reusable components.
5. Virtual DOM, more efficient.

# Project Front-end Structure

```
> public
< src
  > assets
  < components
    < admin
      ✓ Goods.vue
      ✓ Rights.vue
      ✓ Sport.vue
      ✓ UserList.vue
      ✓ Calories.vue
      ✓ Food.vue
      ✓ HelloWorld.vue
      ✓ Home.vue
      ✓ Introduction.vue
      ✓ Login.vue
      ✓ Welcome.vue
    < plugins
      ! element.js
    < router
      ! index.js
  > store
    ✓ App.vue
    ! main.js
  ! .browserslistrc
  ! .gitignore
  ! babel.config.js
  ! package.json
  ! package-lock.json
```



# Project Details -- Main.js

1. Based on Vue CLI skeleton, import Element UI, Vue Router and axios in the app.
2. Define global styles in assets folder and use them.
3. Set default request baseURL globally.

```
import Vue from 'vue'
import App from './App.vue'
import router from './router'
import store from './store'
import './plugins/element.js'
import './assets/css/global.css'
import './assets/font/iconfont.css'
import axios from 'axios';

Vue.prototype.$http = axios
axios.defaults.baseURL = "https://still-stream-50614.herokuapp.com"

Vue.config.productionTip = false

new Vue({
  router,
  store,
  render: h => h(App)
}).$mount('#app')
```

# Project Details -- router

1. Register all routes in `VueRouter`.
2. Authentication middleware.

```
router.beforeEach((to, from, next) => {
  if (to.path == '/login') return next();
  const userFlag = window.sessionStorage.getItem("user");
  if (!userFlag) return next("/login");
  next();
})
```

```
const routes = [
  {
    path: "/",
    redirect: "/login"
  },
  {
    path: "/login",
    component: Login
  },
  {
    path: "/home",
    component: Home,
    redirect: "/Welcome",
    children: [
      {
        path: "/Welcome",
        component: Welcome,
      },
      {
        path: "/user",
        component: UserList,
      },
      {
        path: "/rights",
        component: Rights,
      },
    ]
  }
]
```

# Project Details -- Login.vue

1. Get the form via ref so that we can use its methods such as resetFields and validate.
2. V-model bind the form and its input field.
3. Add input validation schema.
4. Handle onclick event.
5. Send login request to backend.

```
login() {
  this.$refs.loginFormRef.validate(async valid => {
    if (!valid) return;
    const { data:res } = await this.$http.post("login", this.loginForm);
    console.log(res);
    if(res.flag == 'ok') {
      window.sessionStorage.setItem("user", res.user);
      this.$message.success("Login success");
      this.$router.push({path: "/home"});
      console.log(res.user);
    } else {
      this.$message.error("Login failed")
    }
});
```

```
<el-form ref="loginFormRef" :rules="loginRules" :model="loginForm" class="login_form" label-width="100px">
  <el-form-item prop="username" >
    <el-input v-model="loginForm.username" prefix-icon="iconfont icondenglou"></el-input>
  </el-form-item>
  <el-form-item prop="password" >
    <el-input v-model="loginForm.password" prefix-icon="iconfont iconmima" type="password"></el-input>
  </el-form-item>
  <el-form-item class="btns">
    <el-button type="primary" @click="login" >Submit</el-button>
    <el-button type="info" @click="resetLoginForm">Reset</el-button>
  </el-form-item>
</el-form>
```

# Project Details -- UserList

Filter and display members in the system:

1. Combine el-row, el-column and el-input component into a reusable search box component.
2. Under the search box component, display a table with el-table and el-table column.
3. Handle events, add dialog and modals.
4. Implement and add reusable pagination component at the bottom.

```
<el-card>
  <el-row :gutter="25" type="flex" justify="center" align="middle">
    <el-col :span="10">
      <el-input
        placeholder="Please input content to search for"
        v-model="queryInfo.query"
        clearable
        @clear="getUserList"
      >
        <el-button slot="append" icon="el-icon-search" @click="getUserList"></el-button>
      </el-input>
    </el-col>
    <el-col :span="4">
      <el-button type="primary" @click="toggleAddDialogVisible">Add User</el-button>
    </el-col>
  </el-row>
```

# Project Details -- UserList

Under the search box component, display a table with el-table and el-table column.

```
async deleteUser(id) {
  const confirmResult = await
  this.$confirm('This action will delete user permanently, click to continue.', 'Reminder', {
    confirmButtonText: 'Confirm',
    cancelButtonText: 'Cancel',
    type: 'warning'
  }).catch(err => err);
  if (confirmResult != 'confirm') {
    return this.$message.info("Deletion cancelled");
  }
  const { data:res } = await this.$http.delete("deleteUser?id=" + id);
  if (res != "success") {
    return this.$message.error("Failed to delete user!");
  }
  this.$message.success("Success!");
  this.getUserList();
},
async showEditDialog(id) {
  const { data:res } = await this.$http.get("getUpdate?id=" + id);
  this.editForm = res;
  this.editDialogVisible = true;
},
```

```
<el-table-column label="Action">
  <template slot-scope="scope">
    <el-button
      type="primary"
      icon="el-icon-edit"
      size="mini"
      @click="showEditDialog(scope.row.id)"></el-button>
    <el-button
      type="danger"
      icon="el-icon-delete"
      size="mini"
      @click="deleteUser(scope.row.id)"></el-button>
  </template>
</el-table-column>
```

# Project Details -- UserList

Handle events, add dialog and modals.

```
<el-dialog
    title="Add user"
    :visible.sync="addDialogVisible"
    width="50%"
    @close="addDialogClosed"
>
    <el-form :model="addForm" :rules="addFormRules" ref="addFormRef" label-width="100px">
        <el-form-item label="Username" prop="username">
            <el-input v-model="addForm.username"></el-input>
        </el-form-item>
        <el-form-item label="Password" prop="password">
            <el-input v-model="addForm.password"></el-input>
        </el-form-item>
        <el-form-item label="Email" prop="email">
            <el-input v-model="addForm.email"></el-input>
        </el-form-item>
    </el-form>
    <span slot="footer" class="dialog-footer">
        <el-button @click="addDialogVisible = false">Cancel</el-button>
        <el-button type="primary" @click="addUser">Confirm</el-button>
    </span>
</el-dialog>
```

# Project Details -- UserList

Under the search box component, display a table with el-table and el-table column.

```
handleSizeChange(newSize) {
  this.queryInfo.pageSize = newSize;
  this.getUserList();
},
handleCurrentChange(newPage) {
  this.queryInfo.pageNum = newPage;
  this.getUserList();
},
```

```
<div>
  <el-pagination
    @size-change="handleSizeChange"
    @current-change="handleCurrentChange"
    :current-page="queryInfo.pageNum"
    :page-sizes="[1, 2, 5, 100]"
    :page-size="queryInfo.pageSize"
    layout="total, sizes, prev, pager, next,
    :total='total'"
  >
  </el-pagination>
</div>
```

# Project Details -- Food.vue

- Fetch recipes and their ingredients from Edamam API.
- Store result in state and render list with v-for.

```
async getRecipes() {  
  console.log(this.query, this.APP_ID, this.APP_KEY);  
  const {data:res} = await this.$http.get(`https://api.edamam.com/search  
?q=${this.query}&app_id=${this.APP_ID}&app_key=${this.APP_KEY}`);  
  console.log(res);  
  this.recipes = res.hits;  
},  
  . . .
```

```
<div class="recipes" v-for="recipe in recipes" :key="recipe">  
  <div class="recipe">  
    <h1>{{ recipe.recipe.label }}</h1>  
    <ol>  
      <li v-for="ingredient in recipe.recipe.ingredients" :key="ingredient">  
        {{ingredient.text}}  
      </li>  
    </ol>  
    <p>Calories: {{recipe.recipe.calories}}</p>  
      
  </div>
```

# Challenges

## 1. New technical stack, learning curve not smooth

Solution:

Watch tutorials.

Don't be afraid to try.

## 2. Time difference

Solution:

Stick to communication plan and development methodology.

Communicate and debug in time.

## 3. Conflicts between front-end and back-end, and how to debug

Solution:

Use postman.

Check API docs.

Use console.log().

Communicate with teammates.

# Challenges

The screenshot shows a Postman request to `https://still-stream-50614.herokuapp.com/login`. The method is GET, and the Body tab is selected, containing the following JSON payload:

```
1 {  
2   "id":1,  
3   "username":"admin",  
4   "password":123456,  
5   "email":123@qq.com",  
6   "role":"Super Admin",  
7   "state":1  
8 }
```

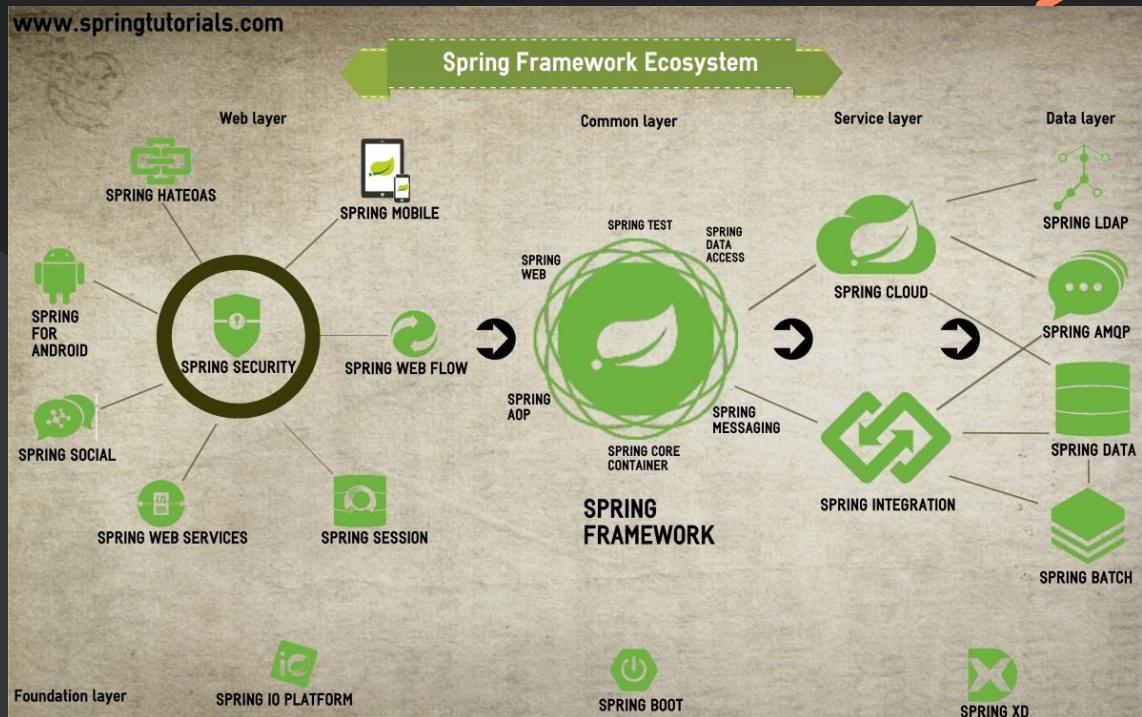
The response status is 200 OK, with a response body containing:

```
1 {"flag":"ok","user":{"email":"123@qq.com","id":1,"password":123456,"role":"Super Admin","state":true,"username":"admin"}}
```

Use postman to test backend API.

# Spring EcoSystem

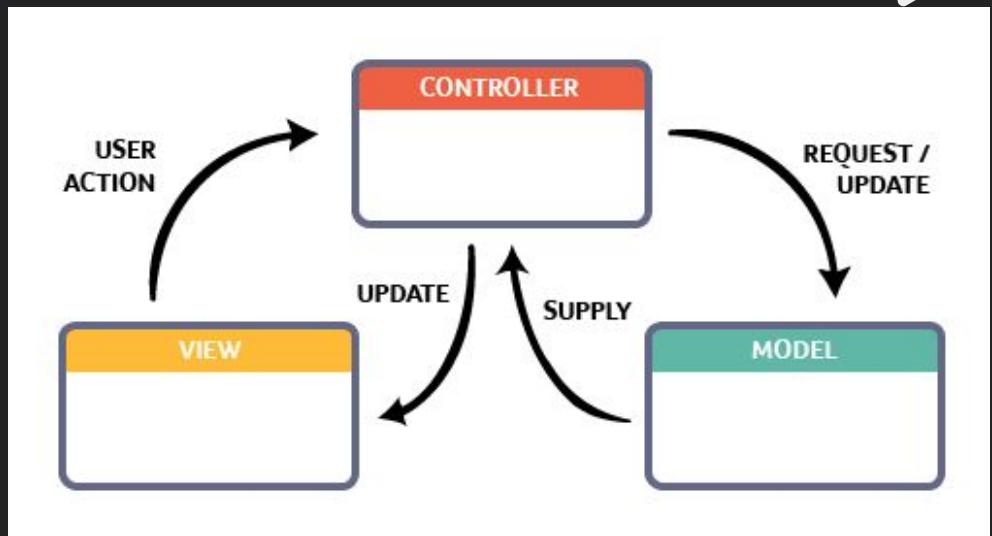
- Inversion of Control
- Dependency Injection
- Spring Boot?



# Spring and MVC

## Features

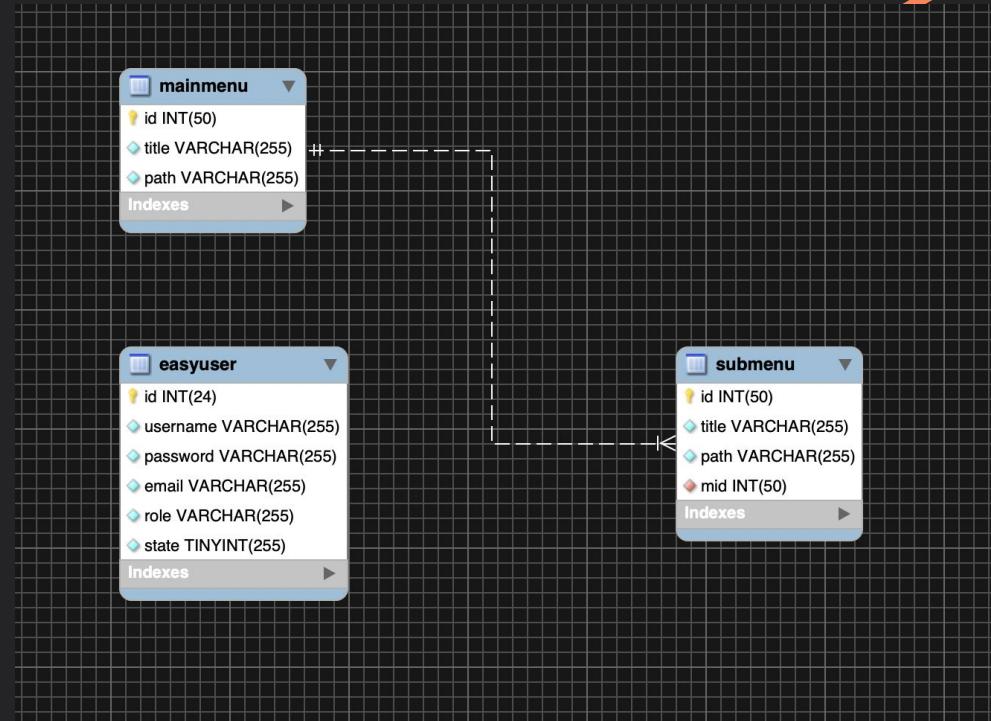
- Flexible, loosely coupled
- Model encapsulate data
- View using Model to render
- Controller processes requests



# MySQL RDBMS

## Features

- Open-source, popular
- Web development
- One of Largest Community
- ClearDB Add-on



# 03

## METHODOLOGY

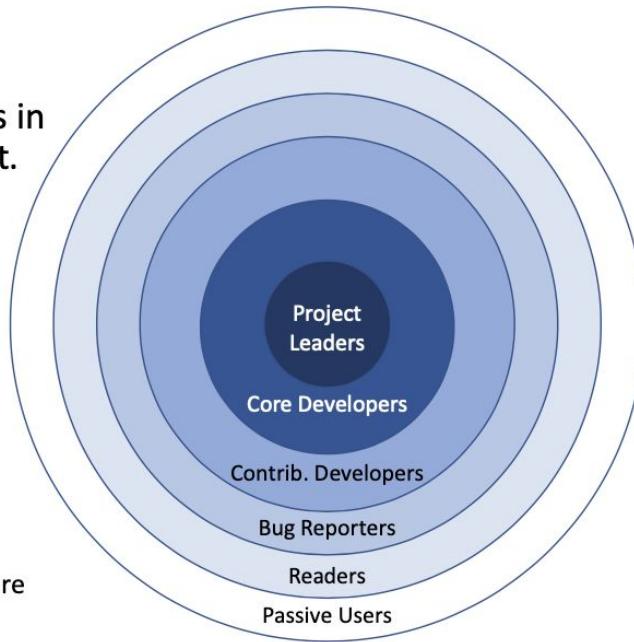
Software Process Methodology  
Communication Plan  
Documents & Tools  
Feedback

# Software Process Methodology

## Open-Source Model

### Open-Source Roles

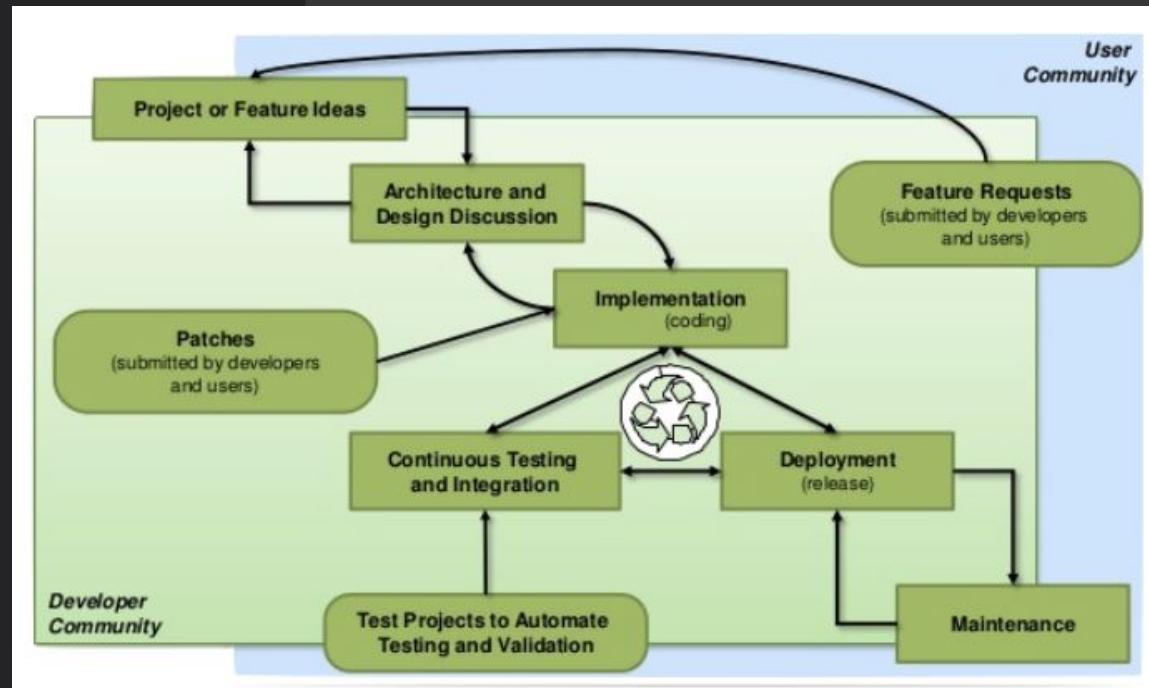
- Here are sample roles in a open-source project.
  - **Project leaders** review architectural changes, evolution
  - **Core developers** work on core code
  - **Contributing developers** work on specific features
  - **Bug reporters** regularly review and test code
  - **Readers** evaluate code
  - **Passive users** use software



# Software Process Methodology

## Open-Source Model

1. Project or feature ideas.
2. Architecture and design discussion.
3. Collaboration on implementation.
4. Source code submission and evaluation.
5. Continuous testing and integration.
6. Source code release and deployment.



# Communication Plan

Communication	Method	Frequency	Goal	Owner	Audience	Others
Team Standup	WeChat	Daily	Discuss what each team member did yesterday, what they'll do today, and any blockers	Project Team	Project team	
Task Progress Updates	Github	Daily	Share daily progress made on project tasks	Project Team	Project team	Ephemeral: Wechat Preserved: Google Doc
Project Progressing Discussion	Zoom + Google Doc	Weekly	Review project status and discuss potential issues or delays. Use google doc to record the discussion.	Project Team	Project team	
Project Status Report	Email	Weekly	Provide the project status to the stakeholders. Ask if there are additional requirements or modifications.	Project Team	Project team + Stakeholders	Compare against milestone
Project Milestone Review	Piazza / Short Email Notice	At milestones	Gather feedback, accumulate potential customers and discuss next steps.	Project Team	Customers	Demo with Tiktok clips / Keep them inform / Decision --> business strategy(coming soon.. newstuff...)
Post-mortem meeting	Meeting	At end of project	Assess what worked and what did not work and discuss actionable takeaways	Project Team	Project team	

# Documents & TOOLS

Github - Repository

- 1. Coding**
  - Frontend
  - Backend
- 2. Docs**
- 3. Plans**
- 4. Weekly report**
- 5. README.md**

master		4 branches	0 tags	Go to file	Add file	Code
	austin110068	Delete the overlapping communication plan.	0086ec6 now	34 commits		
	backend	fix(master): edit user role and state		2 days ago		
	docs	feat(master): adjust on webpage setup date		26 days ago		
	frontend	final front-end		4 hours ago		
	plans	Add plan and report directories.		7 hours ago		
	weekly report	Add plan and report directories.		7 hours ago		
	README.md	Update README.md		3 hours ago		
	assignment-5-GRADED.md	Create assignment-5-GRADED.md		13 days ago		

# Documents & TOOLS

## Github - Backlog

Code Issues Pull requests Projects 2 Wiki Insights Settings

Q is:open

2 Open ✓ 0 Closed Sort ▾

**Backend** Private Updated 2 days ago Create a backend project board for backend development issue. ...

**Frontend** Private Updated 3 hours ago Create a frontend project board for frontend development issue. ...

New project

**Frontend** Updated 3 hours ago

To do

- Align background color and text language. Added by austin110068

In progress

- Set up dev environment and create website. Added by austin110068
- Develop and import login components and configuration. Added by austin110068
- CSS style. Added by austin110068
- Design login validation rule. Added by austin110068
- Router for front-end to back-end. Added by austin110068
- Implement sport module. Data is fixed in frontend. Added by garnetwzy
- Implement calorie indexing module. Added by austin110068

Done

- Upcoming tasks

**Backend** Updated 2 days ago

To do

- i18n adoption on menu routers. Added by valentinzhaoy

In progress

- Unit tests. Added by valentinzhaoy

Done

- Upcoming tasks

# Documents & TOOLS

Github - Wiki

The screenshot shows a GitHub Wiki page for the "CS5500 Project Fitness Pal" repository. The top navigation bar includes links for Code, Issues (1), Pull requests (1), Projects (2), Wiki (selected), Insights, and Settings. The main content area is titled "Home" and shows a message from valentinzhaoy edited 11 days ago with 4 revisions. Below this, the title "CS5500 Project Fitness Pal Wiki" is displayed, followed by a welcome message and a description of the purpose of the wiki. A sidebar on the right contains sections for "Pages" (with 3 items: Home, API, Introduction), a search bar, and a link to "Add a custom sidebar". At the bottom, there is a section for cloning the wiki locally with the URL <https://github.ccs.neu.edu/>.

Code Issues Pull requests Projects 2 Wiki Insights Settings

Home

valentinzhaoy edited this page 11 days ago · 4 revisions

Edit New Page

## CS5500 Project Fitness Pal Wiki

Welcome to the project-fitness\_pal wiki!

This is the place where we record knowledge about this project and share our decisions for project features and designs, architecture of the project and implementation details.

## Portals

### Member Introduction

Member Intro

### Backend endpoints

Backend API Doc

Pages 3

Find a Page...

Home

API

Introduction

+ Add a custom sidebar

Clone this wiki locally

<https://github.ccs.neu.edu/>

# Documents & TOOLS

## Weekly Report

CS5500 My\_Fitness\_Pal Weekly Report

將郵件翻譯為: 繁體中文 | 一律不翻譯自: 英文

Chien-Yu Chen  
週日 2021/4/11 下午 05:48  
收件者: Gust, Philip

Hi Prof.Gust,

I'm the primary contactor of Team My Fitness Pal.  
Here's the weekly report of our team.

<https://docs.google.com/document/d/1mB1y41c3DiLBMsNjEODyXNFBTY5NcRa2dJ8G3bLhk4/edit>

**Google Docs - create and edit documents online, for free.**

Create a new document and edit with others at the same time -- from your computer, phone or tablet. Get stuff done with or without an internet connection. Use Docs to edit Word files. Free from Google.

[docs.google.com](https://docs.google.com)

If you have any concern or suggestion, please don't hesitate to contact me.

Thanks!

Best Regards,  
Austin

Gust, Philip  
Hi Austin, I apologize about the delay in responding. I have sent a request to access your document using my Gmail account philgust@gmail.com. I will review it once I can access it. -Phil

Chien-Yu Chen  
Hi Prof.Gust, I apologize for the late response! I have confirmed your request two days ago, but I forgot to inform you right away/got stuck in some other things at that time...). Please notice me if you still can't access to our report document or you have

將郵件翻譯為: 繁體中文 | 一律不翻譯自: 英文

Gust, Philip  
週四 2021/4/15 下午 09:23  
收件者: Chien-Yu Chen

Thanks, I can access it now. The report looks good. I will look forward to the next one.

-Phil

...

CS5500 Project Weekly Report

檔案 編輯 查看 插入 格式 工具 外掛程式 說明 上次編輯是在數秒前

←  
4/17 Weekly Report  
4/10 Weekly Report

### 4/17 Weekly Report

- What we've done this week
  - Finished login page development & start working on features

To do

- Align background color and text language.  
Added by austinchens

In progress

- Implement sport module. Data is fixed in frontend.  
Added by austinchens
- Implement calorie tracking module.  
Added by austinchens

Done

- Set up dev environment and create website.  
Added by austinchens
- Create user import sign components and configuration.  
Added by austinchens
- CSS style.  
Added by austinchens
- Design login validation rule.  
Added by austinchens
- Router for front end to back end.  
Added by austinchens

Backend

Updated 8 days ago

- Unit tests.  
Added by austinchens

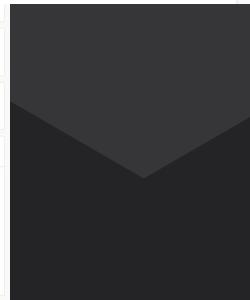
In progress

- Set up dev environment and MySQL database.  
Added by austinchens
- Create user, interfaces, services and controller.  
Added by austinchens
- Backend routes dev.  
Added by austinchens
- Subtitles dev.  
Added by austinchens

Done

- Frontend routes.  
Added by austinchens
- Main menu, layout, services and controller.  
Added by austinchens
- Create test data in database.  
Added by austinchens
- Deploy MySQL database using CreateDB.  
Added by austinchens
- Deploy Server on Heroku.  
Added by austinchens

● Modify repository contents to make it more structured and readable.  
○ Readme



# Feedback

## Announcement

 note @295  

### Team Fitness Pal needs your feedback

Hi pal!

Do you have a hard time trying to stay motivated in keeping healthy?  
Do you want to lose weight, lower BMI and gain strength by diet and sports planning?  
Stay tuned to our product, My Fitness Pal! We're trying to build a platform app that could record our personal data and track whether we're keeping healthy and staying fit.

As we're working on our development, we would love to hear from your voice to improve our product.  
We really appreciate for your help by filling out this [survey](#).

[https://docs.google.com/forms/d/e/1FAIpQLScKQhxNOddEzjqcKDjZfCu-XLiqQ3crETGeVP7\\_N1b8RGPlmG](https://docs.google.com/forms/d/e/1FAIpQLScKQhxNOddEzjqcKDjZfCu-XLiqQ3crETGeVP7_N1b8RGPlmG)

Check out our [web page](#) for latest news and features!

- Team Fitness Pal

[project](#)

[edit](#) · good note 0



### My Fitness Pal - Review

Hi guys! Thanks for participating in our product review. Your statements and suggestions are a big motivation 

What's your age? \*

under 18  
 18 - 26  
 26 - 40  
 40 above

Did you have experience in using this kind of app before? \*

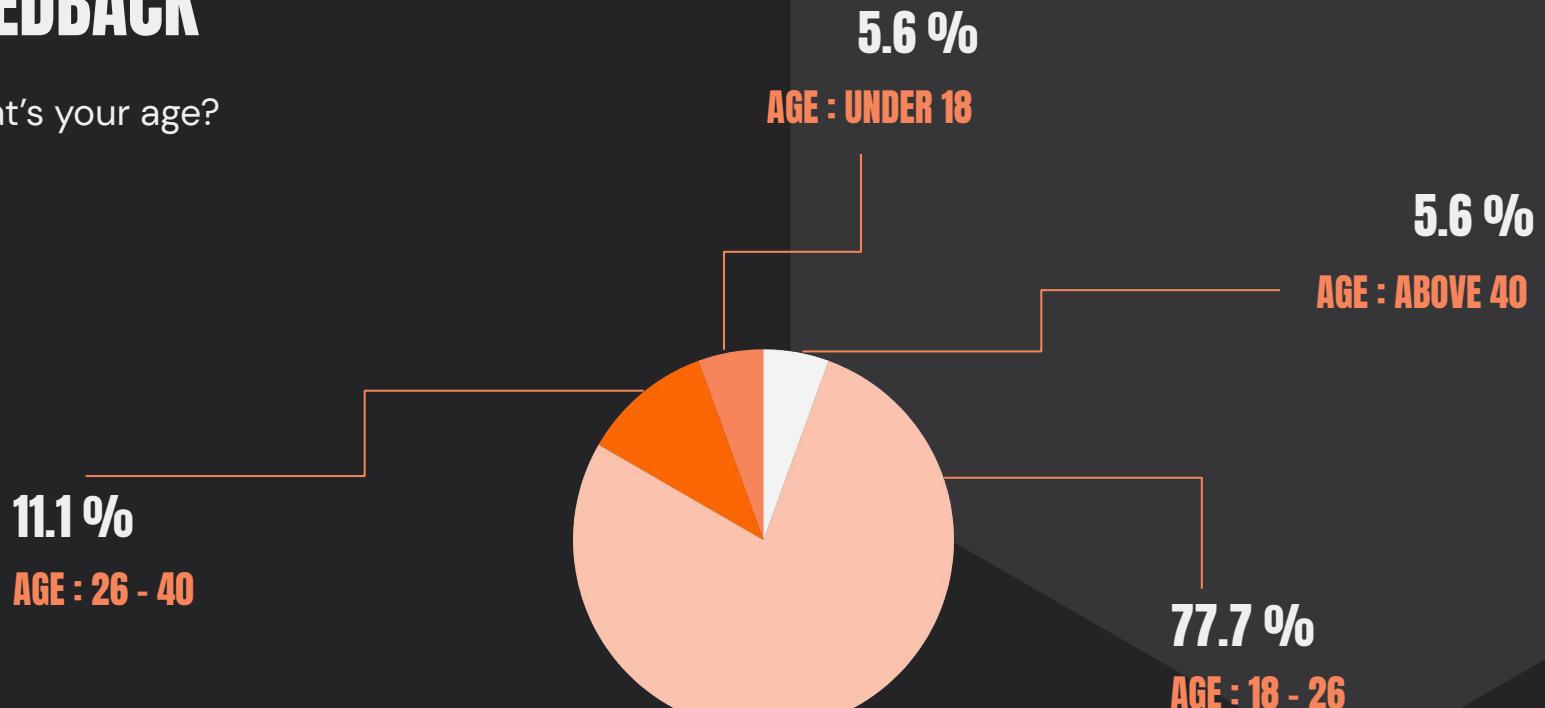
Strongly agree  
 Agree  
 Normal  
 Disagree  
 Strongly disagree

Do you like our website page design? \*



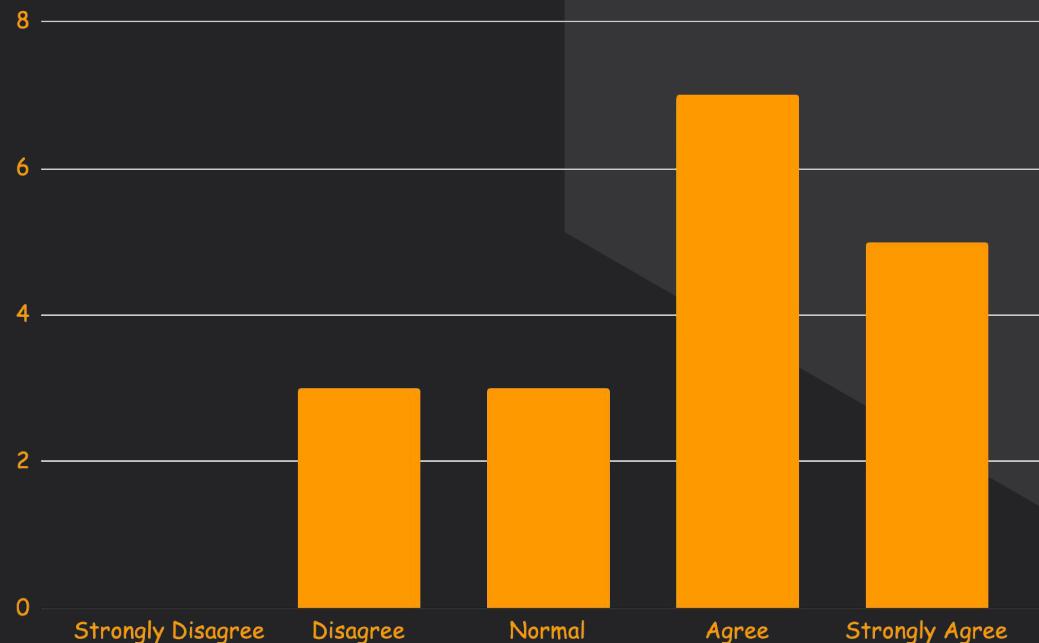
# FEEDBACK

What's your age?



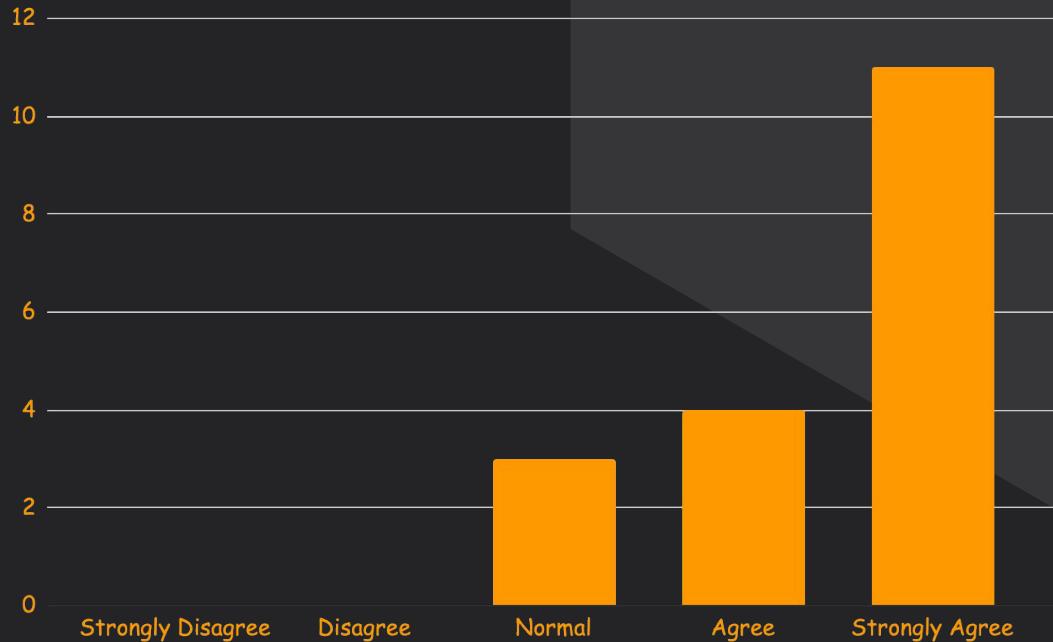
# FEEDBACK

Did you have experience in using this kind of app before?



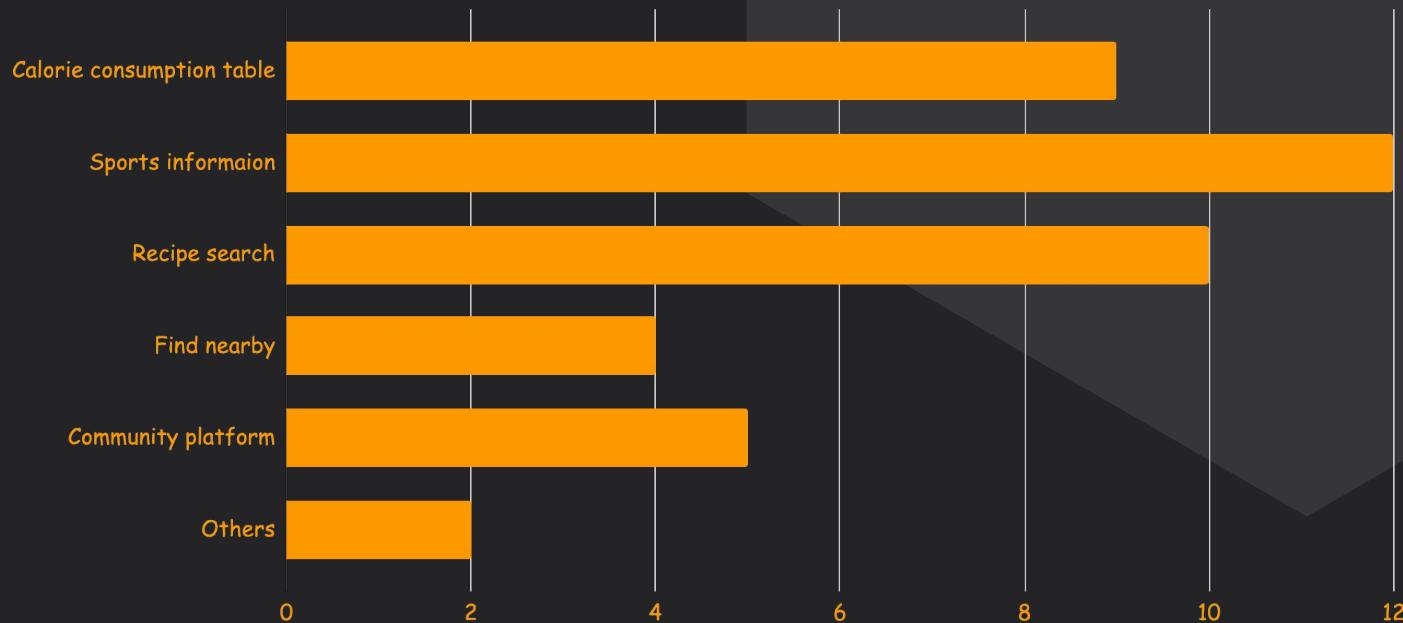
# FEEDBACK

Do you like our website page design?



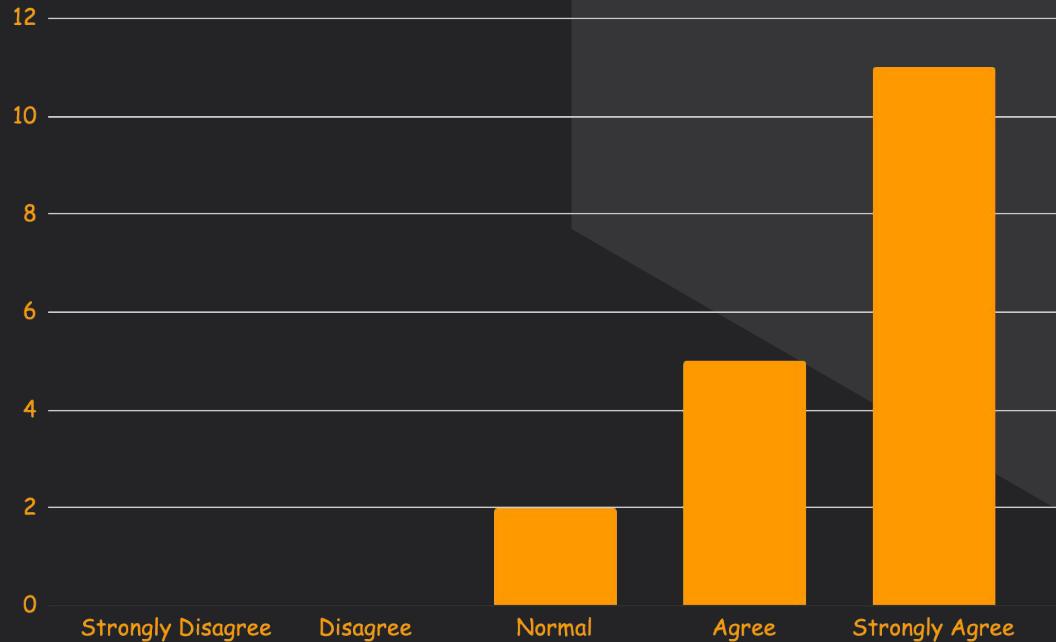
# FEEDBACK

Which features do you like or need in our product?



# FEEDBACK

Will you recommend others to use our product?





# RECAP

04

Q & A

# THANKS!



[https://github.ccs.neu.edu/2021SPCS  
5500SB/project-fitness\\_pal](https://github.ccs.neu.edu/2021SPCS5500SB/project-fitness_pal)



<https://my-fitness-pal.herokuapp.com/#/login>

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

**PLEASE KEEP THIS SLIDE FOR ATTRIBUTION.**