ECE 18-652
Final Project Presentation

Hello!

We are Team A1

Congshan Lv, Guangyu Chen, Jin Gu, Jiyu Shi, Pan Li

Overview

- Software Architecture
- Features of Our App
- How to Test
- Practices and Quality attributes
- Lessons learned

1 Software Architecture

Software Architecture

HTML, CSS, JavaScript	Basic web building blocks
Bootstrap	Mobile compatibility

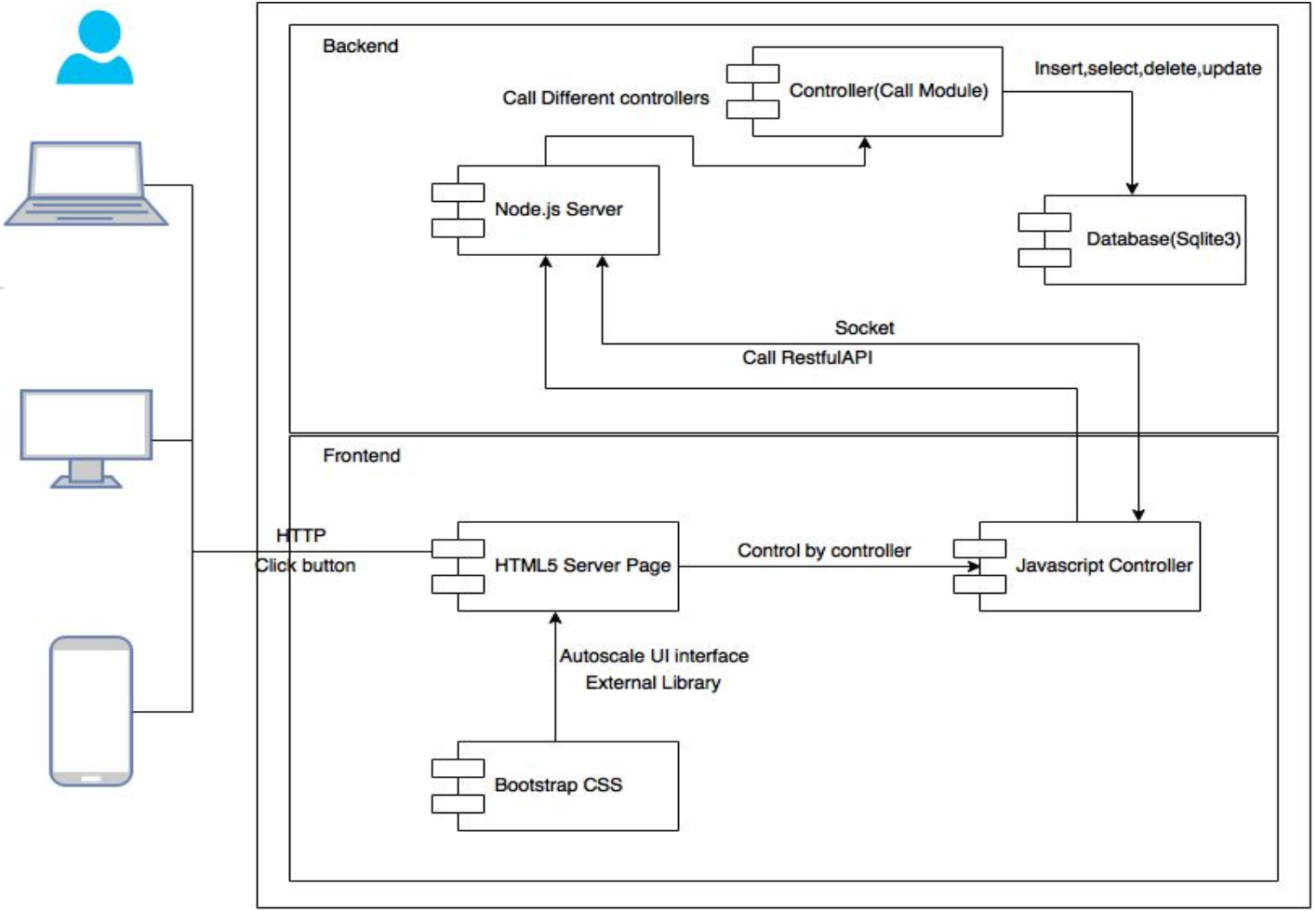
Express.JS	Server framework
Node.JS	Runtime environment
SQLite3	Database

Software Architecture

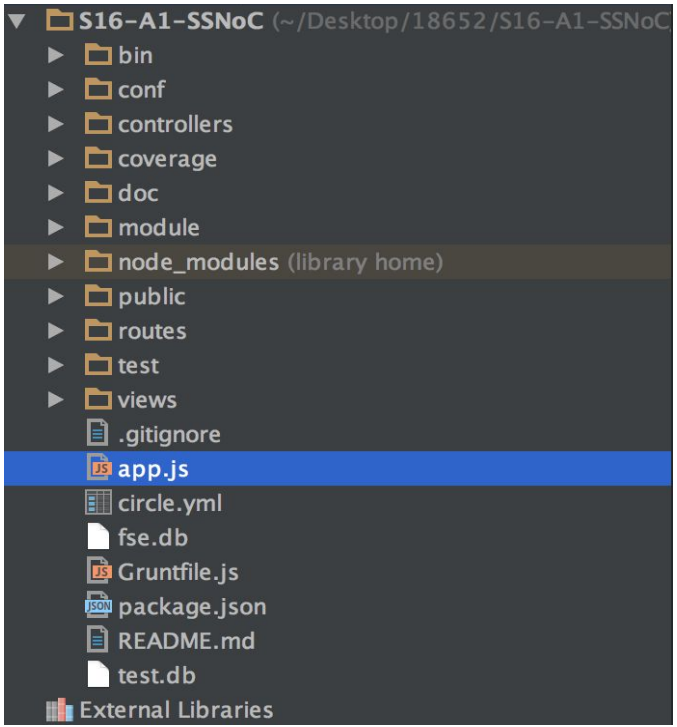
Github	Version control and issue tracking
Trello	Project and backlog management
CircleCI	Continuous integration

Software Architecture

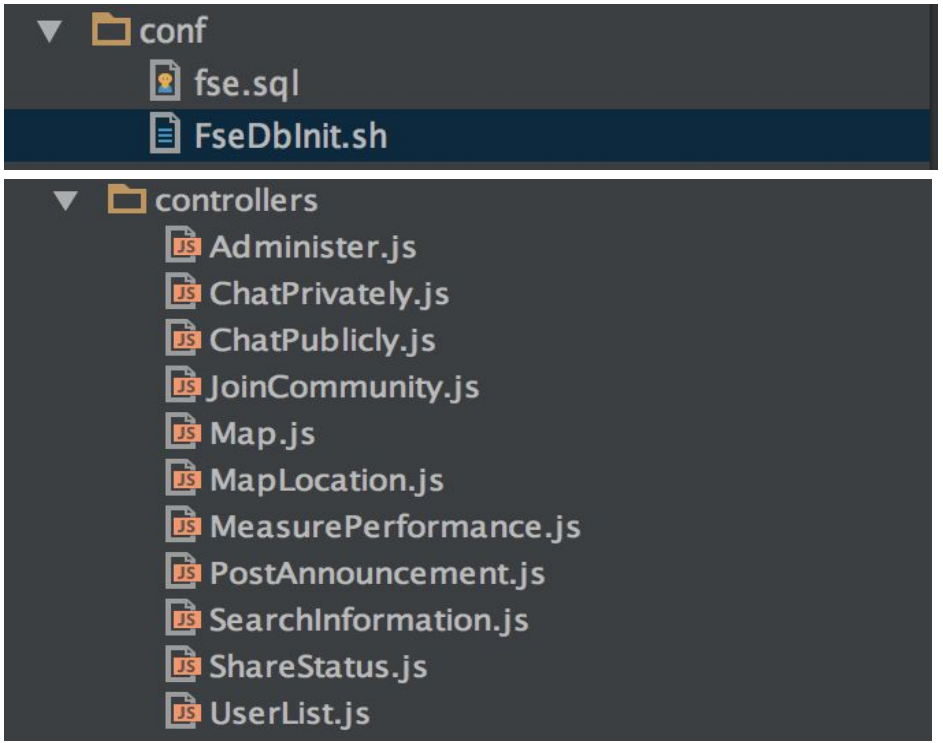
Selemium IDE	UI, system and acceptance testing
Excel	Documenting for application’s RESTful API
StarUml	Diagram editing, creating UML artifacts
Webstorm	Integrated Development Environment
Istanbul	CodeCoverage
Mocha & Grunt	Used for unit test


Backend
Call Different controllers
Controller(Call Module)
Insert,select,delete,update
Node.js Server
Database(Sqlite3)
Socket
Call RestfulAPI
Frontend
HTTP
Click button
HTML5 Server Page
Control by controller
Javascript Controller
Autoscale UI interface
External Library
Bootstrap CSS

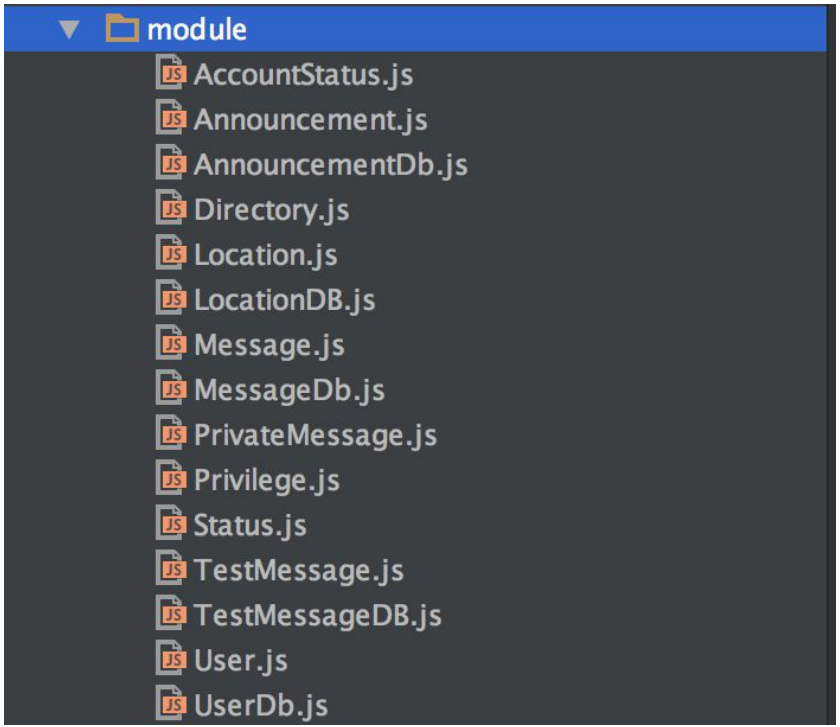
Code Architecture



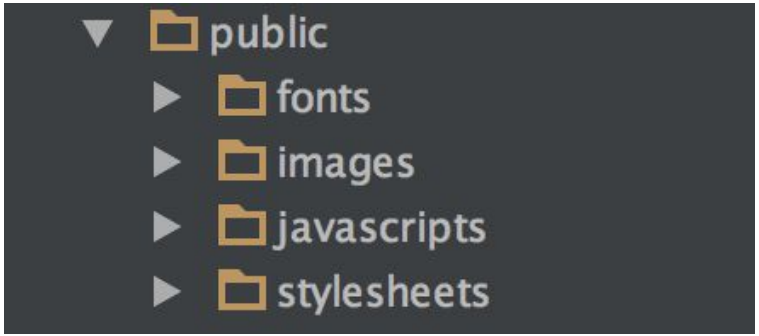
Code Architecture

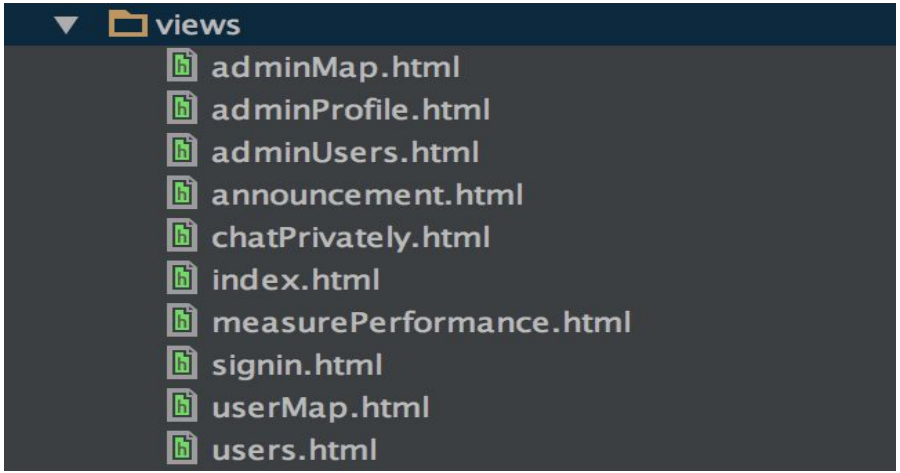


Code Architecture

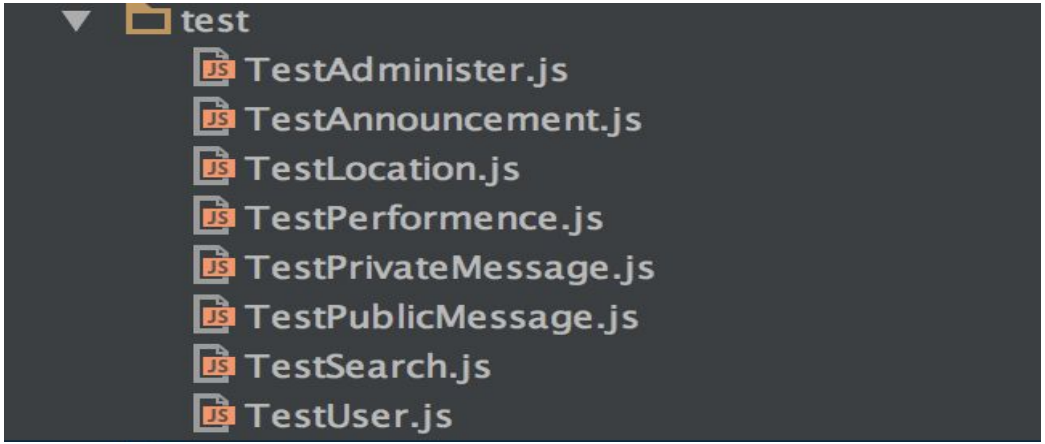


Code Architecture





Code Architecture



Code Push

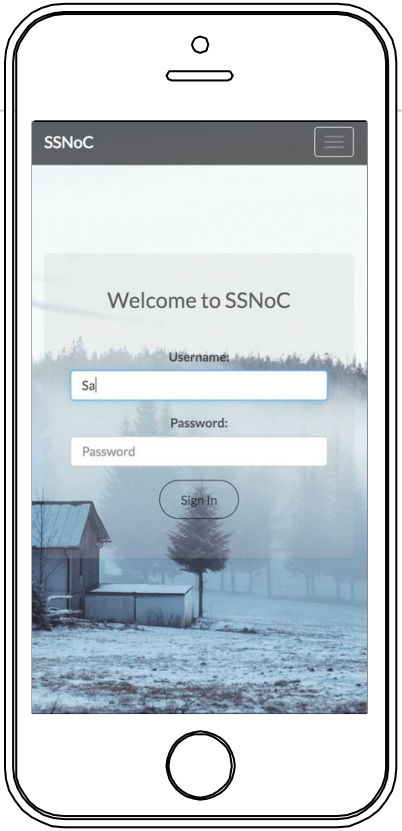
2

Features of Our App

What can citizens do?

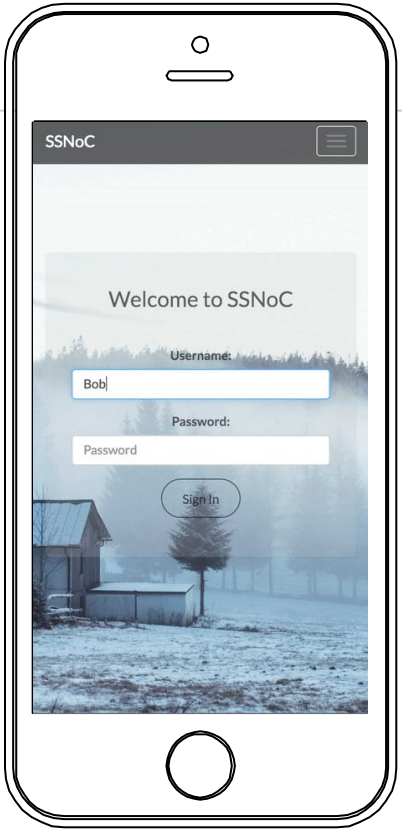
Login

As an existing user, you can login to SSNoC with your username and password.



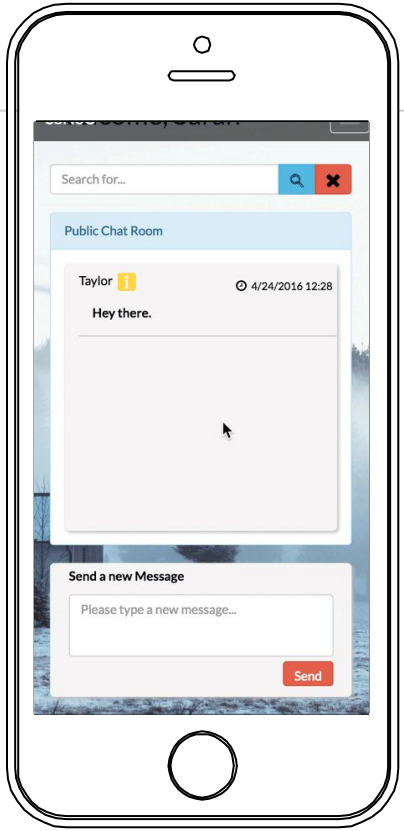
Login as New User

As a new user, you can register and login to SSNoC with your username and password.



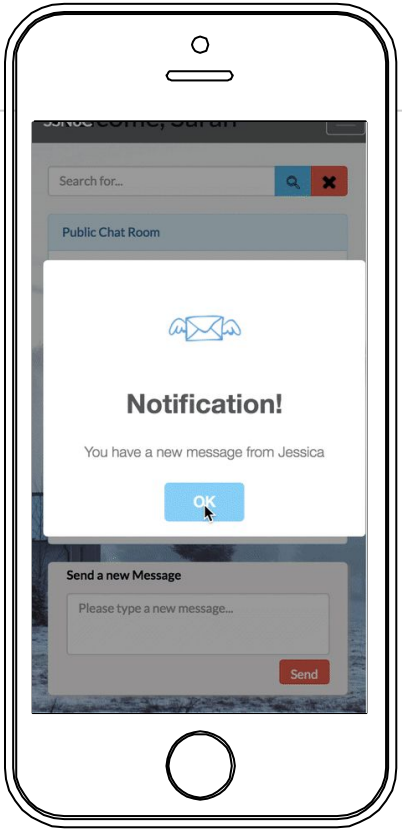
Chat Publicly

Users would enter the public chat room after login, where they would be able to join the public chat.



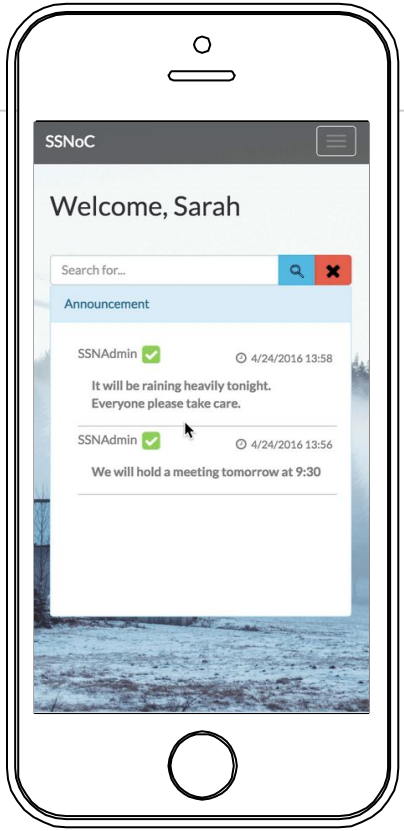
Chat Privately

Users can also have one-on-one conversation with another user.



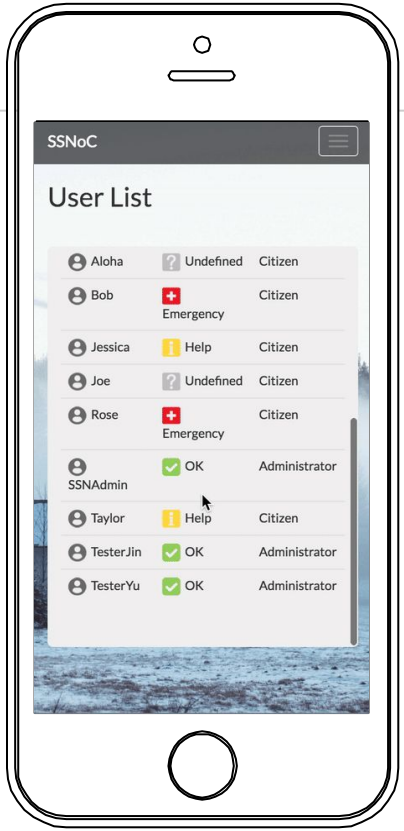
Announcements

Citizens can view/search the announcements posted by the administrator.



User List

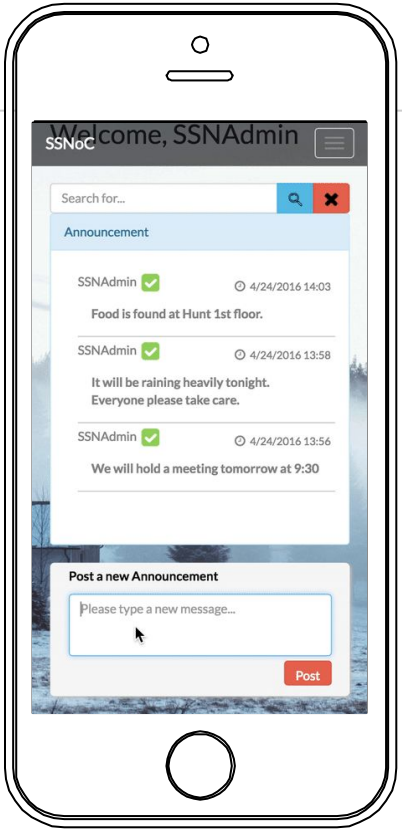
Users can view a list of all users and their status.
Users can also search a user by status or username.



What can administrators do?

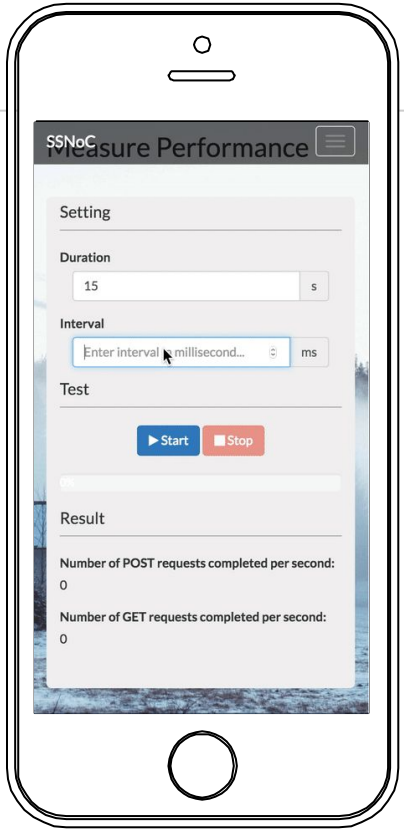
Announcements(admin)

Besides viewing and searching, the administrator also have the privilege to post an announcement.



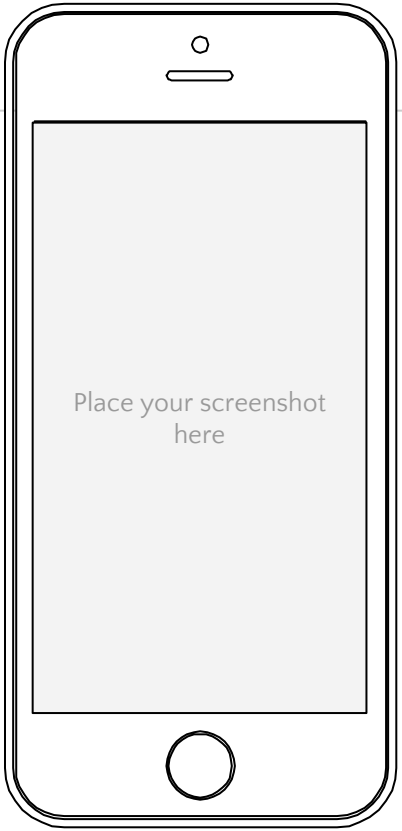
Measure Performance(admin)

Administrators can measure the performance of the system by evaluating the running time of GET/POST requests.



Administer User Profile(admin)

Administrators can modify user profile. The user would be logged out if there is a modification on his profile.

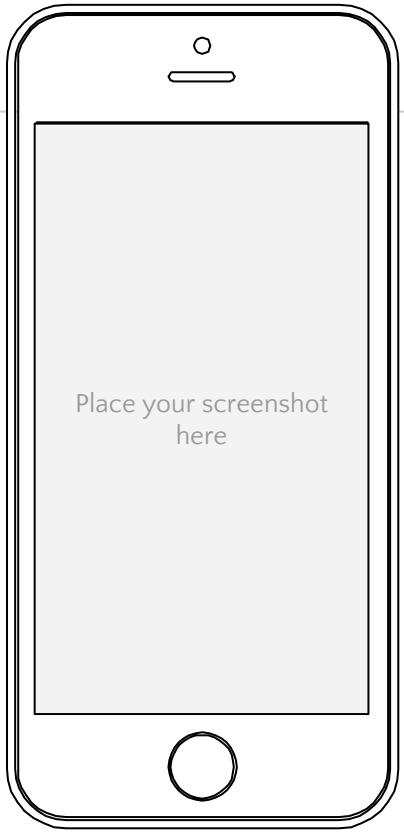


Unique Features

Map(unique feature)

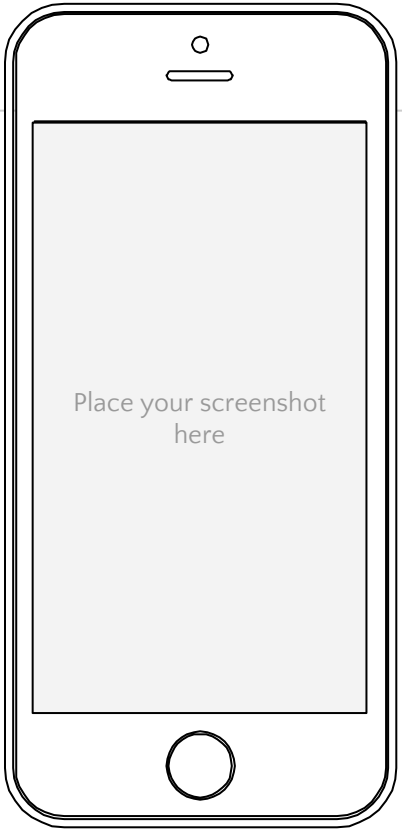
Users can share their locations on the map, and also see where other users are.

- Status: OK
- Status: Help
- Status: Emergency
- Status: Offline



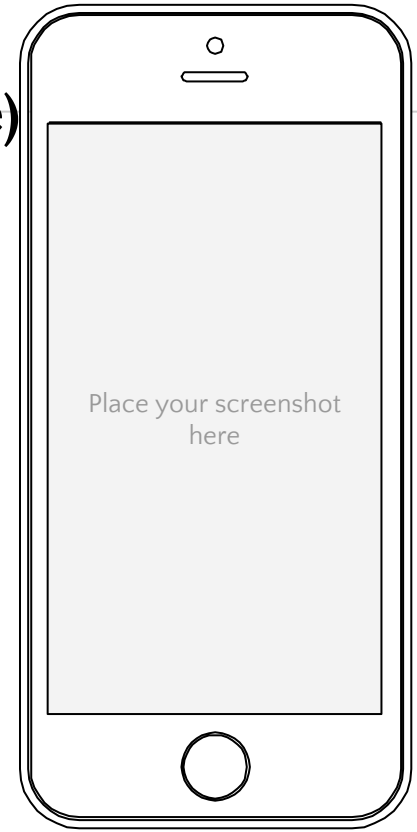
Map(unique feature)

Administrators can mark the map with markers such as food, medicine, and water.



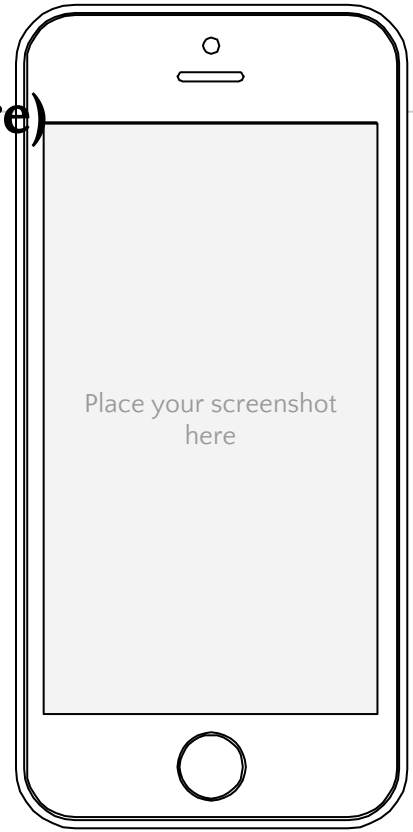
Modify Profile(individual feature)

Users can modify their own profiles.



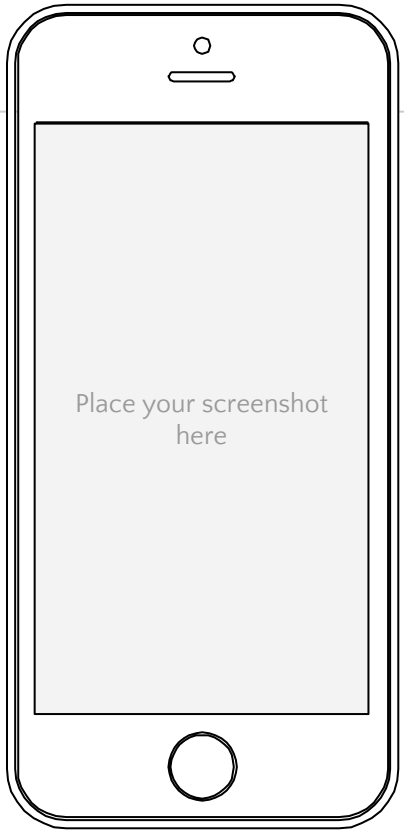
Delete Message(individual feature)

Users can delete messages in multiple ways: single, multiple, time period.



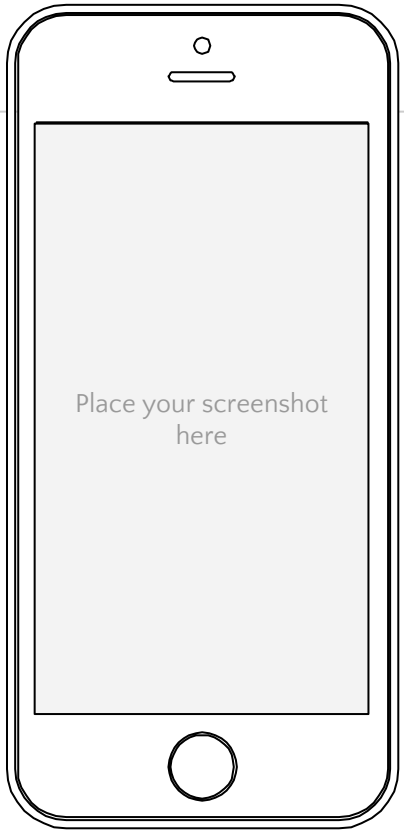
Posts(individual feature)

Users can send posts, and also view posts from other people.



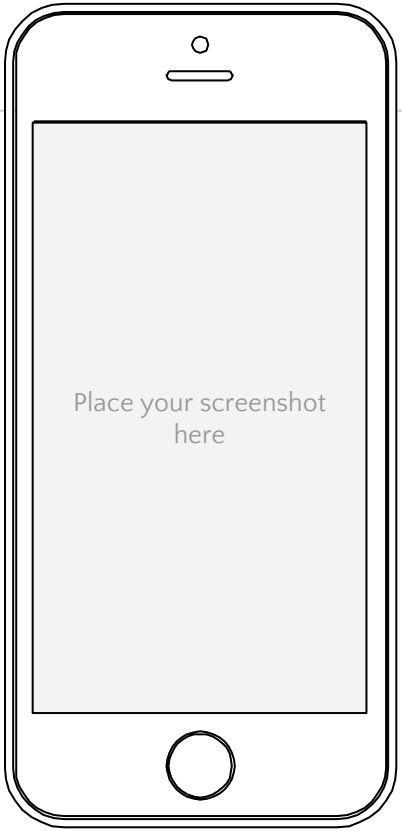
Group Chat(individual feature)

Users can invite other people into a chat group, and hold private conversations within the group.



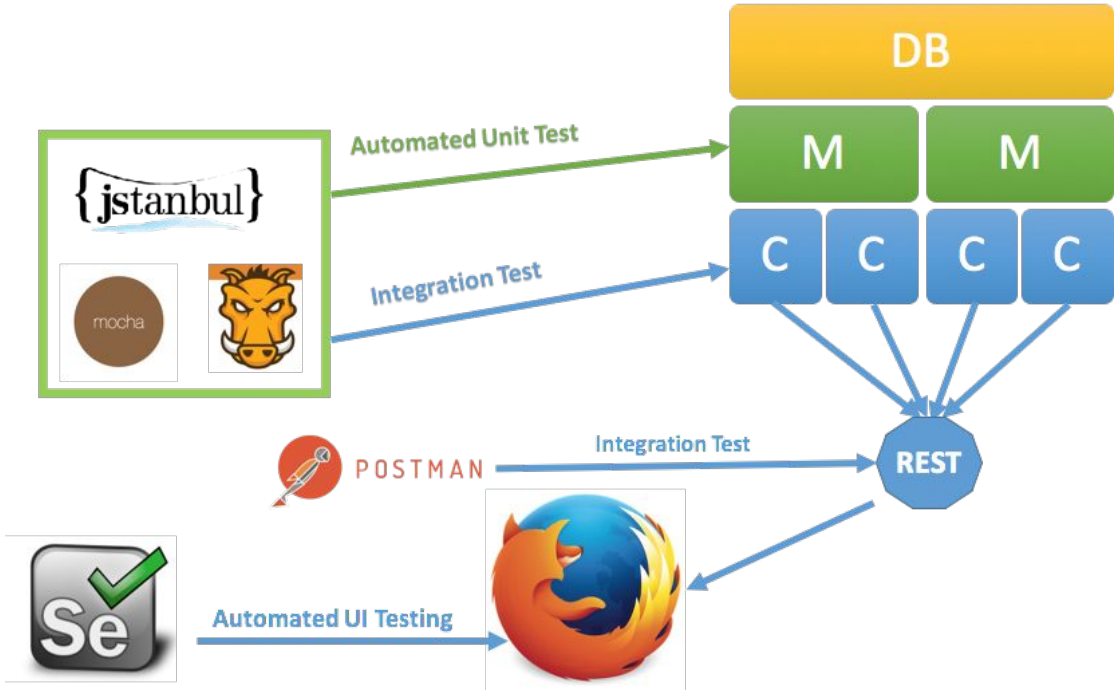
Alerts(individual feature)

Users can receive alert, in situation of emergency. The sound of the alert is different in different situations.

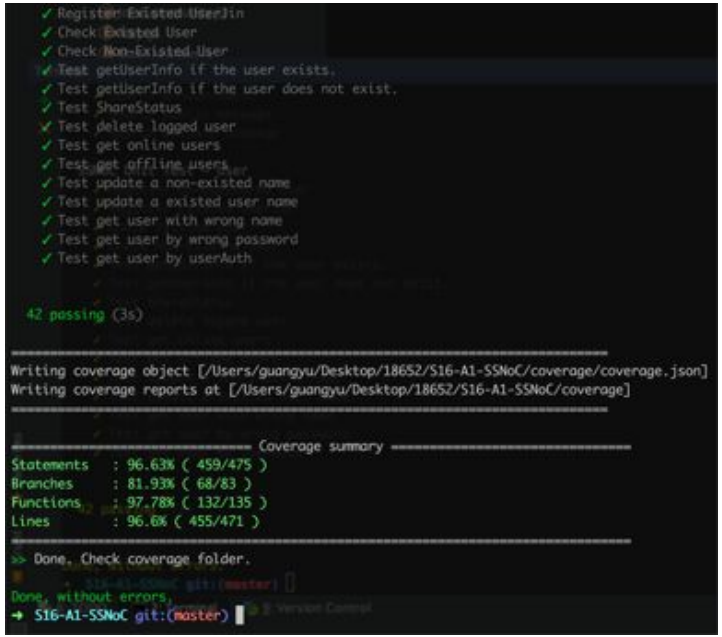


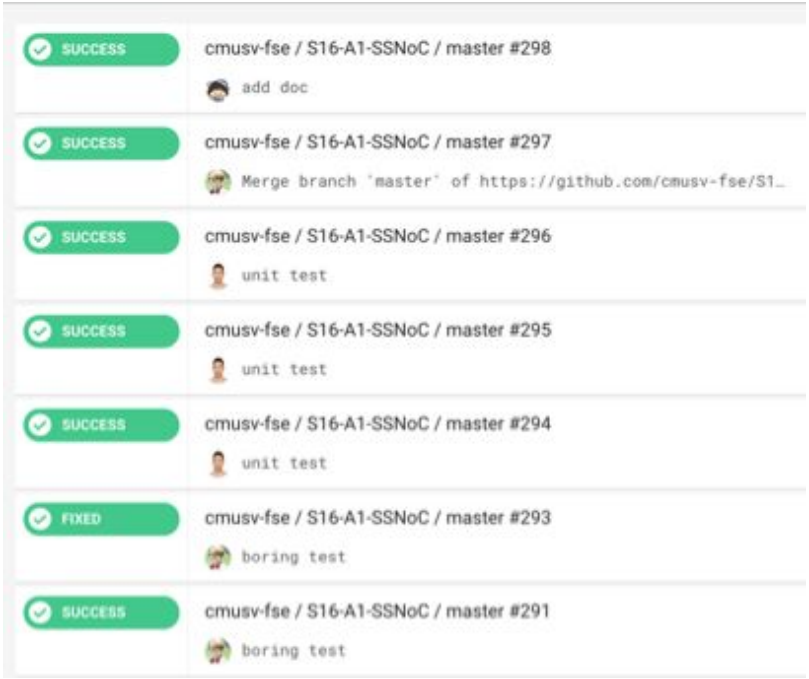
3 How to Test

Test Overview

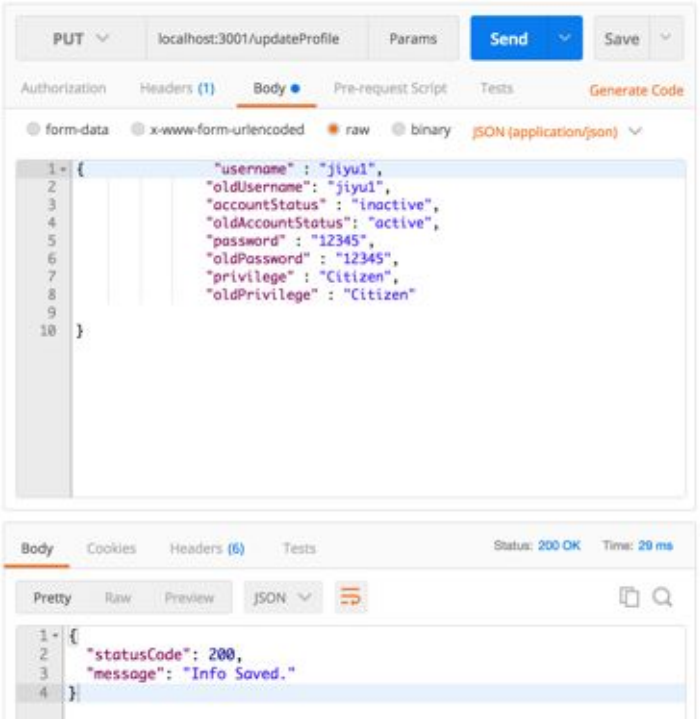


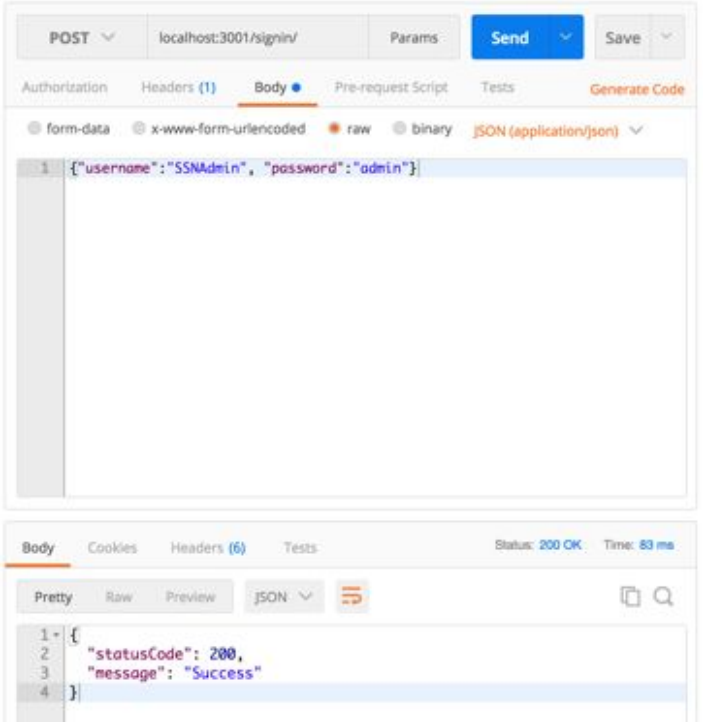
Automated Unit Test



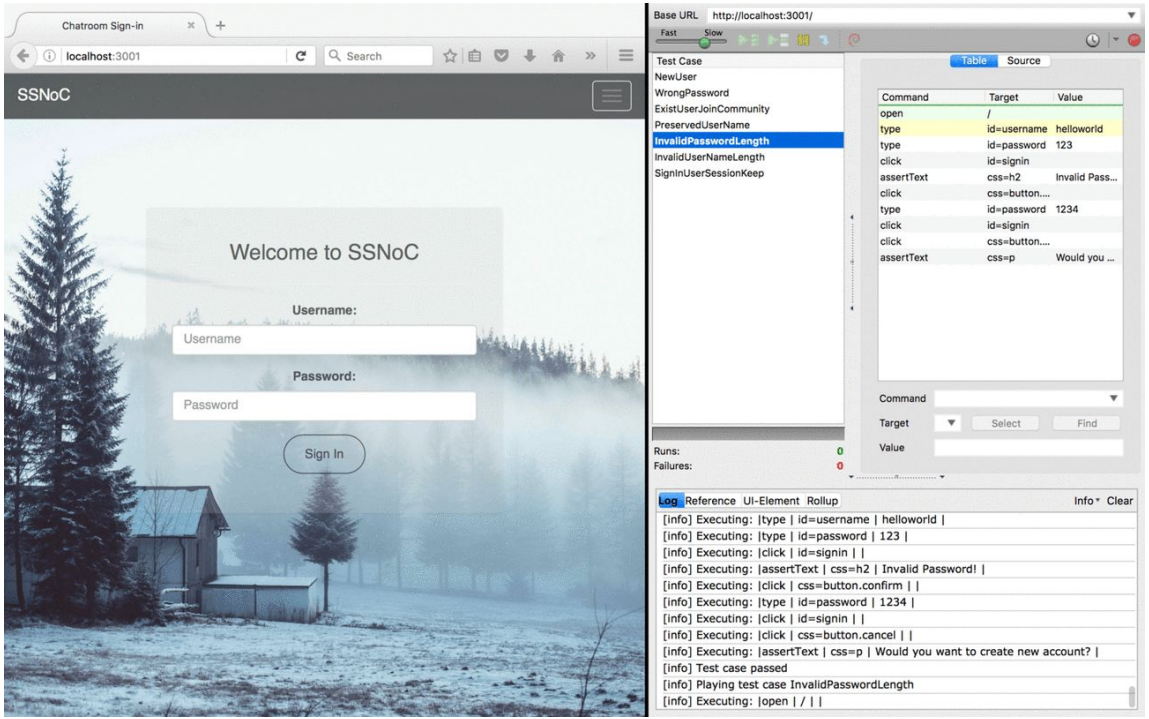


REST Integration Test



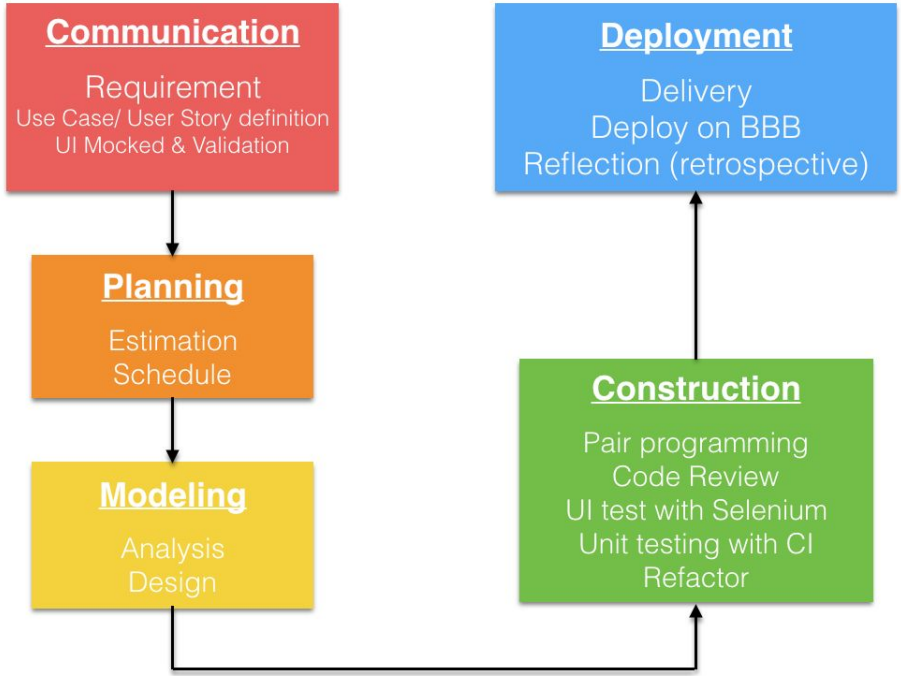


Automated UI Test



4 Practices and Quality attributes

Software Engineering Practices



Software Engineering Practices

Planning

Estimation

Schedule

Iteration1:Join Community [V = 3, P=5]

1 13/17 uc

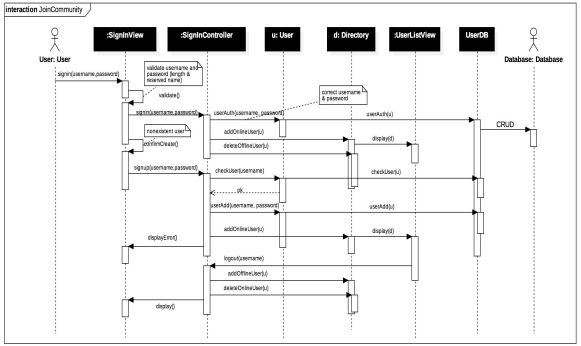
Iteration 1:Chat Publicly[V = 2 , P = 8]

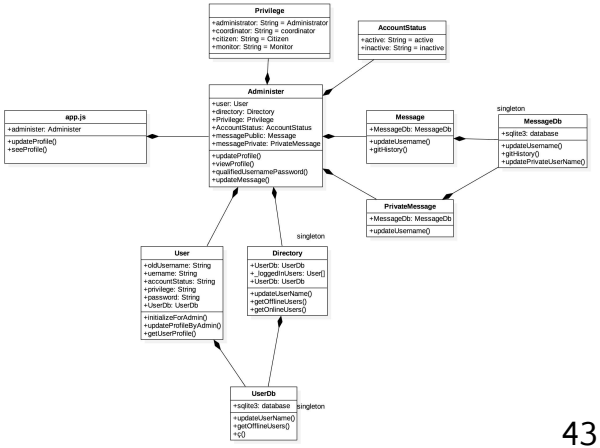
1 11/17

Software Engineering Practices

Modeling

Analysis
Design





Software Engineering Practices

Construction

Pair programming
Code Review
UI test with Selenium
Unit testing with CI
Refactor

Software Engineering Practices

Construction

Pair programming
Code Review
UI test with Selenium
Unit testing with CI
Refactor

Quality Attributes

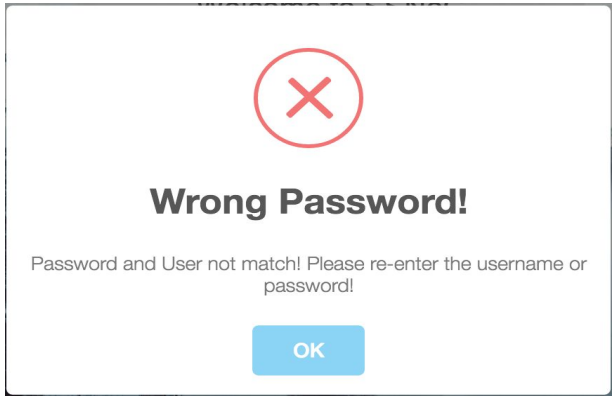
External/ Runtime/ Visible

- Performance
- **Robustness**
- Reliability
- …

Internal/No-Runtime/Invisible

- Testability
- Portability
- Reusability
- **Extensibility**
- …

Software Engineering Practices




```
$.post("/signin",{
        username: username,
        password: password
    },
    function(response) {
        console.log(response);
        if (response.statusCode === 200) {
            window.location = "/index?username=" + username;
        } else if (response.statusCode === 403) {
            swal({
                title: "Wrong Password!",
                text: "Password and User not match! Please re-enter the username or password!",
                type: "error",
                confirmButtonText: "OK"
            });
            $password.val('');
        } else if (response.statusCode === 405) {
            swal({
                title: "Invalid Username and/or Password!",
                text: "Password and User not valid! Please re-enter the username or password!",
                type: "error",
                confirmButtonText: "OK"
            });
            $password.val('');
```

Front-end: Handle “wrong password” login

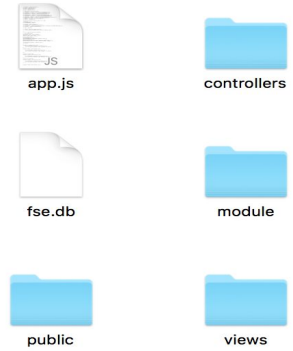
```
exports.checkSignIn = function(req, res) {
    var username = req.body.username;
    var password = req.body.password;
    new User().initialize(username).exist(function(result){
        if (result == 401){
            res.json({"statusCode":401, "message": "User not found"});
        } else if (result == 403){
            new User().initialize(username, password).userAuth(function(result,user) {
                if (result == 403) {
                    res.json({"statusCode": 403, "message": "Wrong password"});
                } else {
```

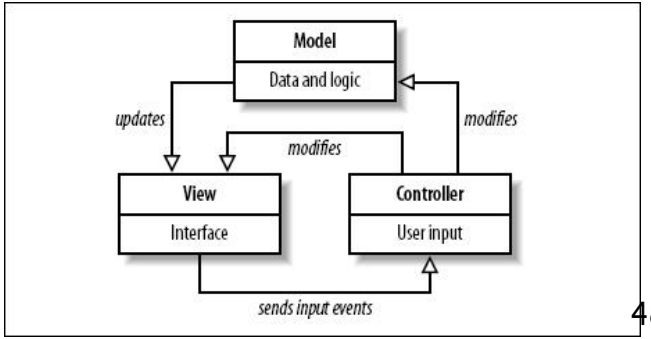
Back-end: Handle “wrong password” login

Software Engineering Practices

Separation of concerns

- Model: data, domain knowledge
- Controller: user interactions, flow
- View: display, layout, presentation, visual elements





5 Lessons Learned

Lessons Learned

- Four Most valuable lessons
  - Give TDD a Try
  - Happy Pair Programming
  - Pay Your Debt Early
  - Build You Project with MVC Architecture

Give TDD a Try

Before

- Write tests later
- No idea about the direction
- Unit tests always broke

After

- Write tests first
- Clearly where to go
- Life is better again

Happy Pair Programming

Before

- Stay at home, died alone
- Find bugs and fix them by individual

After

- Efficiency improves
- Find bugs easier
- Team work is happy

Pay Your Debt Early

Before

- Work later before the due day
- Spend much time to pay the debt

After

- Efficiency improves
- A more elegant use case
- Life is better again

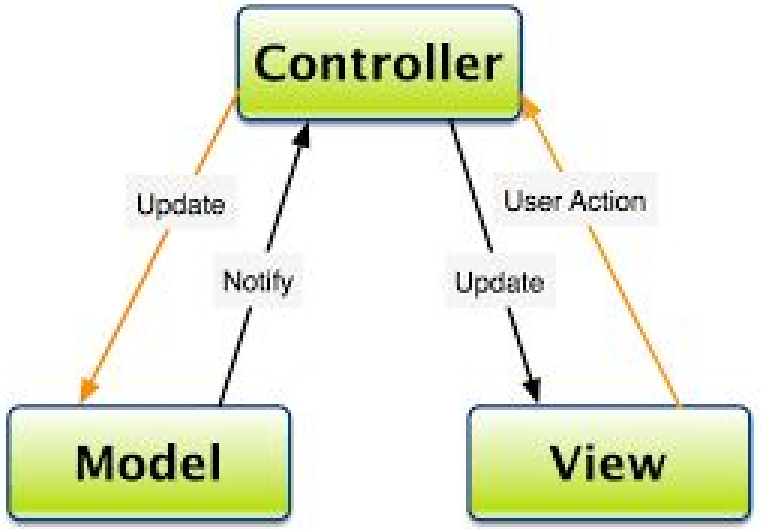
Welcome to MVC Architecture

Before

- Everything is in a mess
- Hard to follow the code

After

- Low coupling
- High cohesion
- Easy to add a new user case
- Easy for a beginner to start



Thank you!