

# Room service on demand

*The final iteration postmortem*

Prepared for Requirement  
Engineering assignment  
by Team 4:

Vladimir Semenov  
Vladimir Solovev  
Valeria Iurinskaia  
Marvin Lopez





# Project introduction. Business goals.

The main project goal is to automate the paid cleaning service operations in Innopolis University dormitory.

Business goals are:

1. Provide the end customers and dormitory administration the easy to use web based system to create and execute service requests for paid cleaning service.
2. Manage and control all related operations.

# Quality of the interview transcript

- Interview transcripts were performed by Team 1
- Transcripts quality is just perfect. We have fully transcribed interviews not only with the project Customers, but also with dormitory administration so we can see a fully covered requirements and additional data for subject area from domain experts.

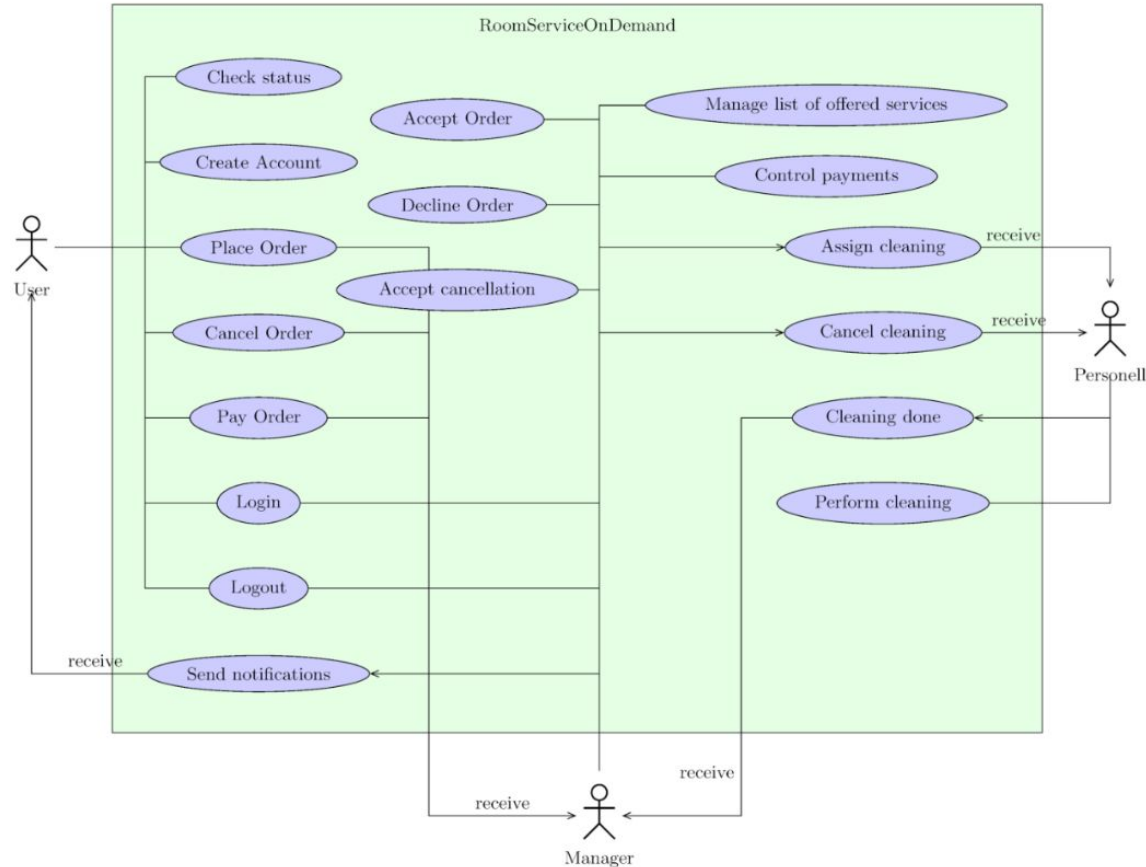
And that is definitely ...



P+

# Quality of use cases

Use cases diagrams and scenarios fully describes all the needed system functionality.



P+

# Quality of MVP and user interface

MVP quality is pretty good. All of the most important use cases were implemented and the code was well documented. As you can see user interface is also performed very stylish and modern. P+ goes to Team 6 for high professional work!



Dashboard



Admin Admin



Back

## Orders



+ Add order

TOTAL

13

APPROVED

5

PENDING

4

DECLINED

0

CANCELLED

1



Status	Name	Cleaning option	Added by	Cost	Added date	Actions
PENDING	189	Whole ap.	Vladimir Semenov		25 Nov 2019	ACTIONS
PENDING	303	Cleaning	qqq qqq		25 Nov 2019	ACTIONS
INACTIVE	444	Cleaning	Admin Admin	123	25 Nov 2019	ACTIONS

# Quality management experience (auto tests)

The screenshot shows an IDE with a project structure on the left, a code editor in the center, and a terminal at the bottom. The project structure includes folders for controllers, helpers, hooks, libraries, mocks, models, and third\_party, along with files like .gitignore, Bootstrap.php, DbTestCase.php, phpunit.xml, TestCase.php, and UnitTestCase.php. The code editor displays the `Auth_test.php` file, which contains a `class Auth_test extends TestCase` with several test methods: `test_sing_in_index()`, `test_sing_up_index()`, `test_forgot_password_index()`, and `test_method_404()`. The terminal shows the execution of the tests, indicating that all tests passed successfully.

```
<?php
class Auth_test extends TestCase
{
    public function test_sing_in_index()
    {
        $this->request('GET', 'auth/login');
        $this->assertResponseCode(200);
    }

    public function test_sing_up_index()
    {
        $this->request('GET', 'auth/signup');
        $this->assertResponseCode(200);
    }

    public function test_forgot_password_index()
    {
        $this->request('GET', '/auth/forgot_password');
        $this->assertResponseCode(200);
    }

    public function test_method_404()
    {
        $this->request('GET', 'auth/method_not_exist');
        $this->assertResponseCode(404);
    }
}
```

Terminal: Local x +  
Line: 52  
Function: main

Time: 1.26 seconds, Memory: 10.00MB

OK (2 tests, 2 assertions)

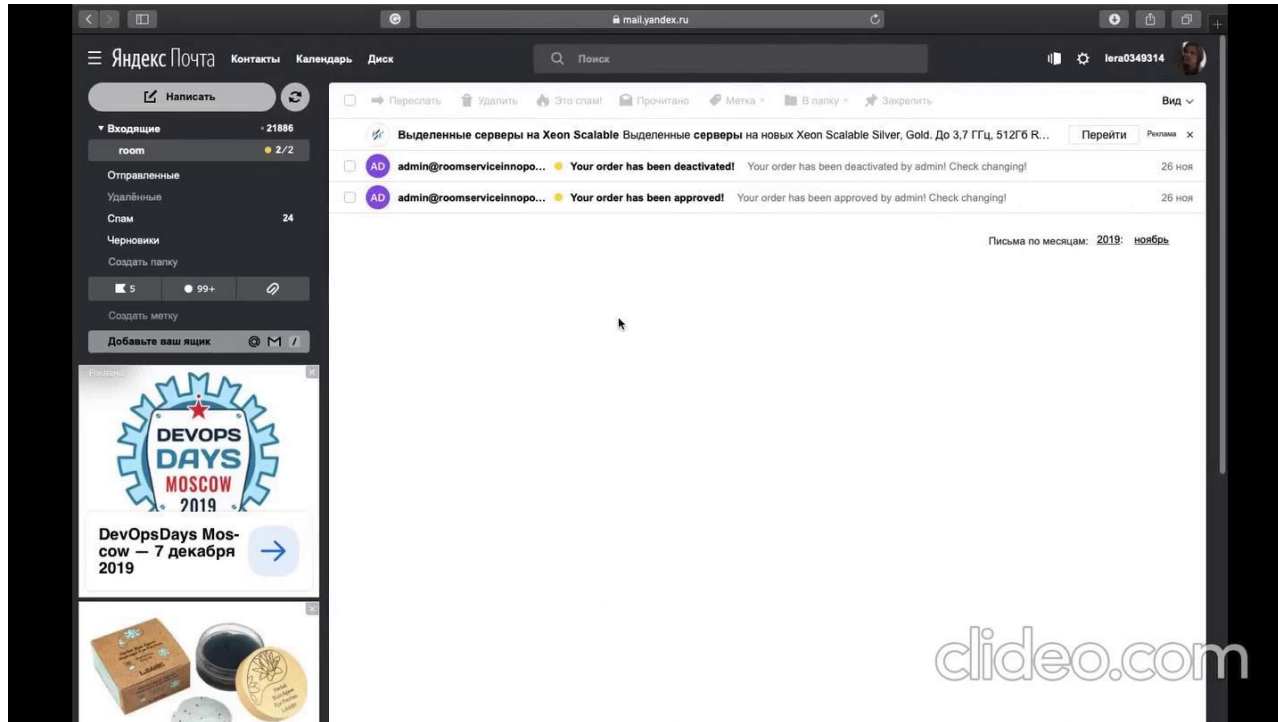
Generating code coverage report in Clover XML format ... done

Generating code coverage report in HTML format ... done

MacBook-Air-Valeria:tests valeria\$ clear

Tests passed: 0 (18 minutes ago)

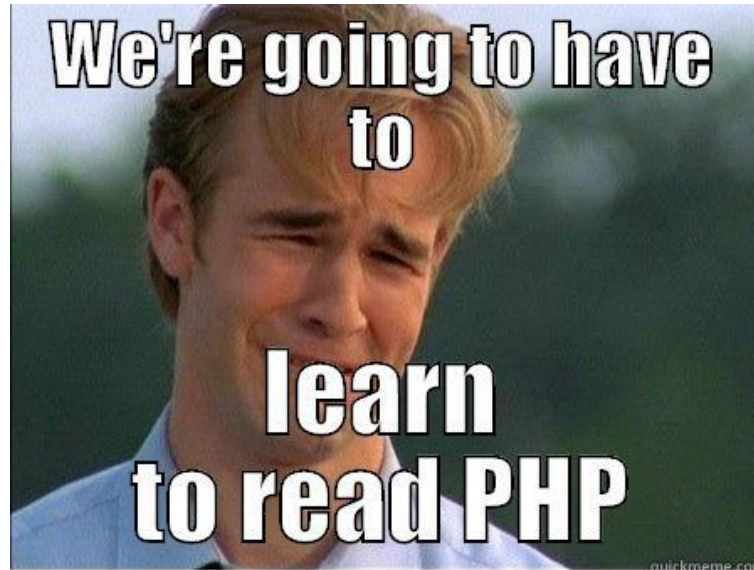
# Quality management experience (manual testing) and use cases demonstration



# Communications with other teams

Every team we worked with during the whole course was always open for help.

Special thanks goes to Boburmirzo from Team 6 for helping us understand the PHP related moments in their implementation!

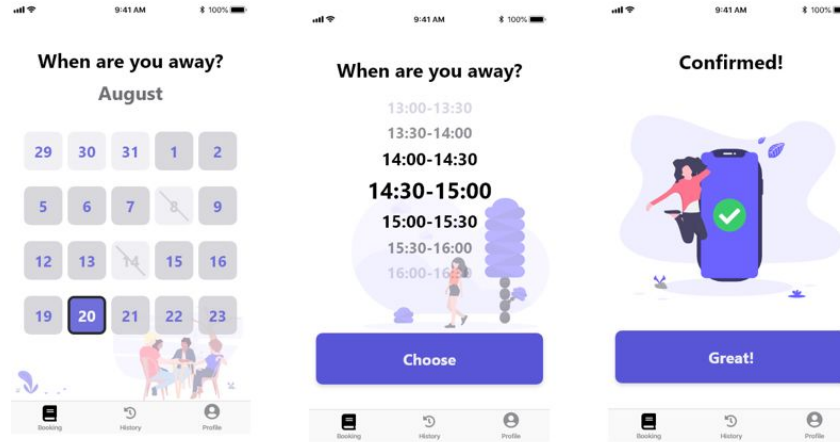




# Lessons learned

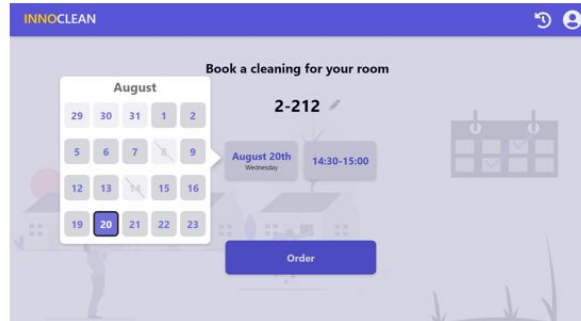
First to say, we were initiators of this project proposal and we were expecting something like this.. :)

IOS support



Android support

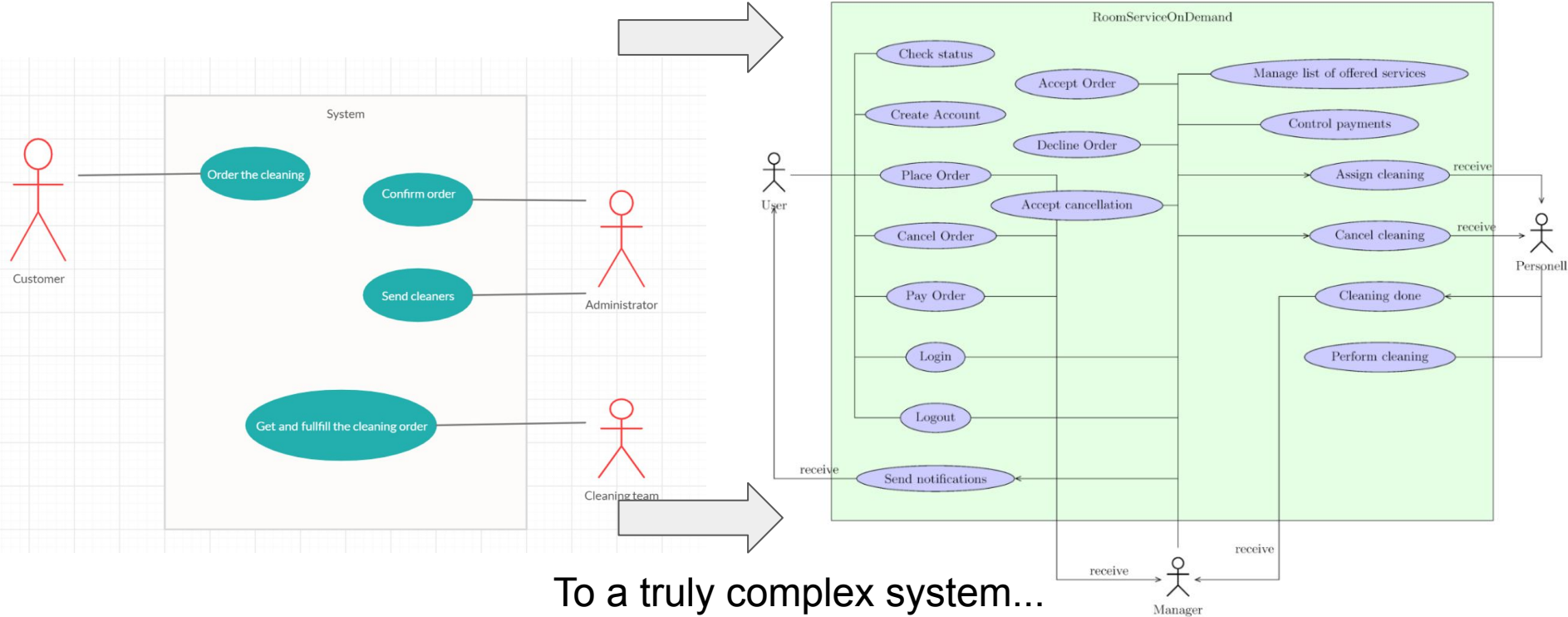
Cross browser support



Any device like Ipad  
adaptivity

# Use cases evolution

Our initial use cases were evolved a lot!



# Implementation strength and weaknesses

Good :

- Excellent user interface, high quality requirements, well documented code

Good, but not so :)

- No payment part realization, but it is absolutely ok for educational project
- Little problems with cross browser support and adaptivity (also not big issue)



# QA

We thank you all for your attention and are ready to answer any of your questions!

