PROJECT PLAN

OnlyDevs;

Content

1. Project Assignment

1.1 Context

This project is intended to solve a problem that SIOUX proposed as a university project for our 3rd semester at Fontys University of Applied Sciences as Software Engineering students.

1.1.1 Description of the problem

The problem we'll provide a solution for refers to managing the appointments that visitors make with the employees from Sioux, keeping track of the main parking lot's visitor spaces, notifying incoming visitors whether they can park there or not, and pointing them towards another Sioux parking if it's full. The Sioux employee with a scheduled meeting should be notified in how much time their visitor will arrive in the lobby based on where they park.

1.2 Goal of the project

The main goal is to satisfy our clients with our software solution. Based on the agreed requirements, we will create a software that meets every stated criteria. We intend to deliver a fast, safe and bug-free system for SIOUX company.

Our system will help our client with managing meetings scheduled with visitors and with having overview of parking spots at the firm.

System is made for three types of users. First user is secretary, with features to manage meetings(create, edit, remove) and seeing into employee's schedules. Second users are visitors who are not directly using the system, however system sends notifications to them(SMS and Emails). Last user is employee whom the meeting is scheduled for. This employee can see in his/her outlook overview of scheduled meetings and is being notified about scheduled meetings.

1.3 The assignment

We are aiming to create a web application with two fundamental functionalities. The first part is scheduling appointments with the employees of Sioux and the second part is keeping track of the parking spots in the parking lot for visitors therefore notifying them where they can park their car. The client's specific requirements are that the system should communicate with their company, the personal information management system they are currently using - Outlook Mail and Outlook Calendar. Keeping track of the parking spots will be done with a camera that reads the license plates when cars are coming in respectively coming out.

1.4 Scope

This project includes	This project does not include
The software to detect availability parking spots for a camera	The camera hardware to detect parking spots
Outlook API communication	Reavailability parking spot if the designated parking spot is taken
Scheduling a meeting with the visitor	Licenses for the software

Look into employee's schedule availability	Notifying the visitor how much time is needed till arriving without car
Canceling a meeting with the visitor	
Update a meeting with the visitor	
An employee can see the scheduled appointment in outlook	
Notifying the visitor how much time is needed till arriving by car	
Redirecting the visitor with a car to the alternative parking spot	
Visitor receiving a SMS where the visitor can park before the appointment	
Visitor receives a confirmation by e-mail for the appointment	

1.5 Conditions

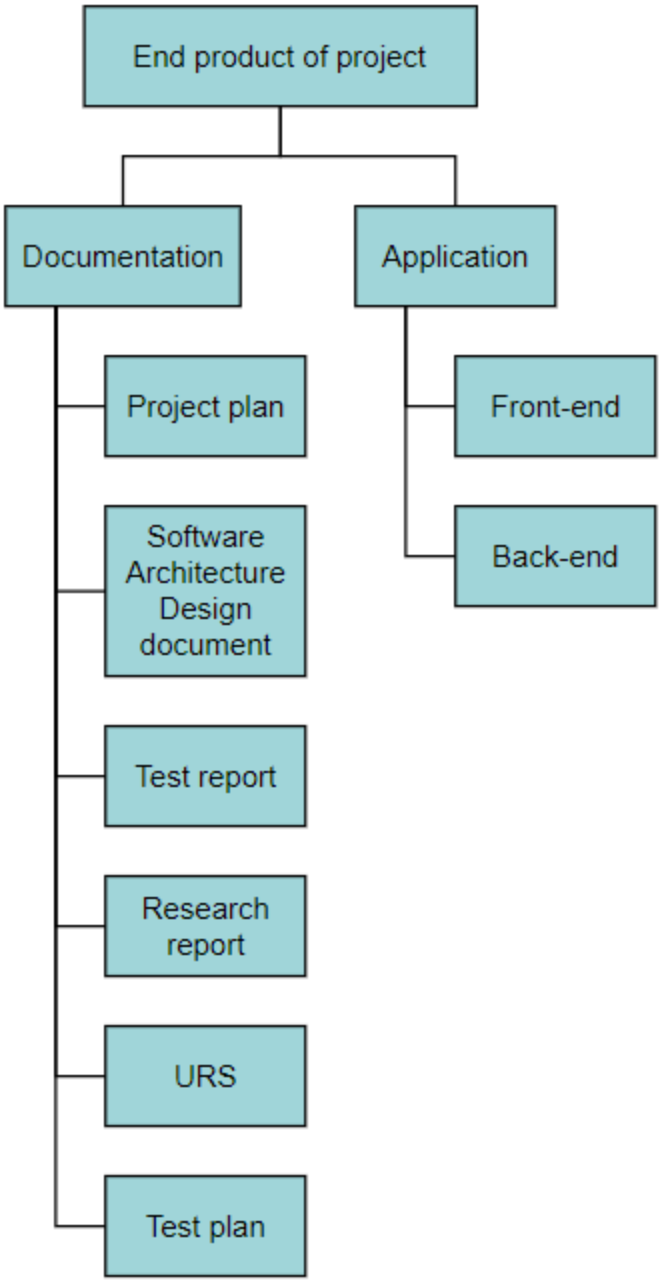
- Software
- No mandatory programming languages, programs or editors recommended by the company
- Our choice: Java(back end), React(front end), IntelliJ(editor)

- Project Management
- No mandatory tools recommended by the company
- Jira
- Microsoft Teams

- Outlook
- GitLab
- Uizard (UI & wireframing)

-> sensors

1.6 End products



Project breakdown structure (PBS)

- Documentation
  - Project plan- the project plan will include an overview of the idea behind the project and its goal. It will go over all aspects

such as planning, financing, time windows etc. without going into too much detail.
    - Software Architecture Document- The SAD will describe the system context and visualize the interactions in the whole application.
    - Test report- The test report will be a follow-up of the test plan. It will describe the results of the tests laid out in the test plan.
    - Research report- The research report will describe the team's findings concerning subjects such as how to implement camera recognition for license plates.
    - URS- The URS will go into more depth about the boundaries of the application's functionality by going over topics such as functional requirements, use cases, etc.
    - Test plan- The test plan will be a through description of what functionalities there are to be tested within the application, the expected results, the information needed for each test, etc.
- Application
    - Front-end- The front-end will give users the ability to interact with the application. React is our choice of framework for this.
    - Back-end- The back-end will be doing the magic behind the scenes. It will be made up of Java, Spring MVC & more.

2. Approach & Planning

2.1 Approach

It is common for requirements to change over the course of the product development cycle in order to meet shifting business requirements, which can be extremely challenging for development teams. That's why for the making of this product we're going to use the SCRUM method.

Agile technologies like Scrum include meetings, roles, and tools designed to help teams work on complex projects collaborate and better structure and manage their workloads.

Each sprint represents in fact a cycle in which we first implement, then review and refactor our tasks. This way, our deliverables will have continuous feedback, and that will give us time to improve or adapt our project functionalities.

2.2 Research methods

Main question

How to create a reliable, maintainable software system which simplifies the process of arranging appointments between SIOX Company clients and SIOUX employees. Software that can keep track of meetings and clients on-site, with filled in parking spots, if there are any, and based on the parking lot situation or arriving way, estimate the time of arrival to the company.

Subquestions

Which programming language should we use?

- Library
    - Available product analysis
    - Community research
    - Design pattern research
    - Literature study
- Field
    - Problem analysis
- Worskhop

    - Brainstorm
    - Multi-criteria decision making

How do we make sure to have reliable and maintainable software system?

- Library
    - Best good and bad practices
    - Literature study
- Field
    - Task analysis
- Lab
    - Component test
    - Non-functional test
    - System and security test
    - Unit test
- Workshop
    - Code review

Who will be using the software?

- Field
    - Document analysis
    - Stakeholder analysis
    - Problem analysis
    - Task analysis
    - Focus group

How do we arrange an appointment/meeting?

- Field
    - Document analysis
    - Stakeholder analysis
    - Problem analysis
    - Task analysis
- Workshop
    - Brainstorm

- Library
    - Available product analysis
- Showroom
    - Ethical check

How do we keep track of meetings and clients on-site?

- Workshop
    - Brainstorm
    - IT architecture sketching
    - Multi-criteria decision making
- Lab
    - Data analytics

How do we estimate the time of arrival to the company?

- Field

- Document analysis

2.3 Phases of the project

Phase 1:

- Activities: Problem analysis and pre-implementation work
    - Activity 1.1:
        - Analyse the problem outlined during the first meeting with the product owner
    - Activity 2.1:
        - Formulate a basic idea of what the functionalities will be
    - Activity 3.1:
        - Carry out research about what technologies and frameworks are optimal for the solution of the problem
    - Activity 4.1:
        - Decide on the technologies and frameworks which will be used for this project
            - Communicate the decision with the product owner

    - Activity 5.1:
        - Decide on what pre-implementation documents are necessary
            - Communicate the decision with the product owner
    - Activity 6.1:
        - Work on the agreed upon documents
- Deliverables:
    - Deliverable 1.1: Project plan
    - Deliverable 2.1: URS
    - Deliverable 3.1: Research conclusion document

Phase 2:

- Activities: Implementation, feedback and refactoring based on feedback
    - Activity 1.1:
        - Update documentation based on feedback
    - Activity 2.1:
        - Creation of the base architecture of the app based on the functional requirements
    - Activity 3.1:
        - Beginning the building process
    - Activity 4.1:
        - Receive feedback
    - Activity 5.1:
        - Implement feedback and continue with the implementation
    - Repeat 4.1 and 5.1
- Deliverables:
    - Deliverable 1.1: Updated Project plan
    - Deliverable 2.1: Updated URS
    - Deliverable 3.1: Walking skeletons of multiple functionalities
    - Deliverable 4.1: Minimum viable product

Phase 3:

- Activities: Finalization of the project
  - Activity 1.1:
    - Create a test plan
  - Activity 2.1:
    - Create a test report
  - Activity 3.1:
    - Finish the process of implementation
  - Activity 4.1:
    - Hand in the documentation and code
  - Activity 5.1:
    - Receive evaluation on our work
  - Activity 6.1:
    - Reflect upon the evaluation and the whole working process
  - Activity 7.1:
    - Wrap up the project
- Deliverables:
  - Deliverable 1.1: Test plan
  - Deliverable 2.1: Test report
  - Deliverable 3.1: Finished code
  - Deliverable 4.1: Reflection

2.4 Time plan

SPRINT	EFFORT	CONTENT
1	**moderate**	**-setting up environment -make documentation(Project Plan, URS etc.)**

		-research about potential solutions to the problem
2	**moderate**	**-build front end -start implementing the basic requirements**
3	**high**	**-start implementing the advanced functionalities -update documentation based on features added/modified**
4	**moderate**	**-testing our application -update/improve certain features**
5	**low**	**-making last changes -working on last documents -showing our results**

3. Project organization

3.1 Team members

- *Daniil Blagoev*

I am a 20-year-old from Bulgaria with a strong sense of responsibility and hunger for improvement in all areas of my life. I have witnessed how most of our team's members work on similar projects and I firmly believe that we will be able to fulfill what is required of us.

- *Iulia Toderascu, 19, Romania*

I have a strong sense of ambition and drive. I thrive on challenges and set goals for myself constantly so I have something to strive for. My goal is never to settle, and I'm always looking for ways to improve and succeed. In my previous projects or jobs, I enjoyed working with a team because I believe it's a nice way to learn from others' perspectives and practices.

- *Jakub Jelinek*

I am a 21 year old student from the Czech Republic. What Fontys taught so far is that the only way to improve my skills is to practice and make mistakes. I am looking forward this project because I feel that we will learn a lot. It is not this monotonous project like notes app or housing administration. I would say that my strengths are in creativity and innovative thinking, however sometimes my innovative thinking can become a minus. My weaknesses are time management and not sometimes overwhelming thinking out of box, which can sometimes lead to new functional requirements that we cannot promise to finish.

- *Rositsa Nikolova*

I am 21 years old, and I am from Bulgaria. I am looking forward to starting my career as a Software Engineer. I like working on projects

that provide me with the opportunity to apply my professional skills but also to learn more and to develop my skills further. I believe constant learning and improving are the keys to success, so I am always open for new challenges. In my opinion, for a successful group project one of the key factors is team dynamics and everyone to be on the same page about what is the end goal of the project, so I am looking forward to seeing how we are going to apply this.

- *Mike Wang*

A little introduction about myself I am born in the Netherlands, 24 years old, I have a little brother and my parents are from China. I am hardworking, always looking for new solutions, open minded and striving to improve as a software engineer. I like programming the most because you can see immediate results once you create your own magic. I look forward to this project as I can improve my professional skills and my communication skills with my group. I also like to learn from my group mates as they have different perspectives and different approaches to a certain problem. I hope we can make a great project out of this.

3.2 Communication

The communication between the team members will be done via WhatsApp – group chat and Microsoft Teams for weekly meetings although most of the meetings will be on site. There are scheduled meetings on Monday, Wednesday and Saturday. The communication with the Product Owners from Sioux will be done via Microsoft Teams, Outlook for emails and on site when possible. There is a weekly Stand-up meeting scheduled on Wednesday with all team members and Product Owners.

3.3 Test environment

Testing strategy

Backend

For testing our software we will be doing unit tests, where each method will be tested to make sure the code is without bugs. For making sure that the code is working even after adding new features and methods, regression testing will be also done.

To proof that the software is running, working and bug free we will also have manual testing following the steps in Test plan document.

We will also make use of User Acceptance Testing (UAT), where we will test our software in real world by the intended audience, in our case product owners.

Test environment and required resources

We will make use of IntelliJ Java Spring boot testing practices and Mockito framework for unit testing. Given the requirement of setting up pipeline in GitLab for contiguous integration we have also done that for automated testing.

3.4 Configuration management

Our git strategy is git flow

We use the follow branching:

features->dev->main

4.2 Risks and fallback activities

There are always risks. Depending on which importance weight we give them, they should be handled with enough effort and should be kept in mind so we can avoid them.

Risk	Prevention activities	Mitigation activities

1	Error, bugs in the system	Always unit test, create branches in GitLab	
2	Losing data	Always store locally and somewhere on server, never only one place	
3	Not following the semester requirements structure	Using Agile methodology, always ask for feedback, stay on track	
4	Losing motivation	With the weather, personal problems, length of the project it might happen. Stay motivated, don’t let it run away	
5	Communication	Miscommunication might always happen. Two group members working on the same thing, no one working on other thing.	Keep writing everything in Jira, have structure, keep everything in order. Make weekly, 2x times a week stand ups to see how is everyone doing
6	Misunderstanding	Some requirements might be understood differently by each person	Have meetings with product owners. Having minutes from meetings