



PROJECT PLAN

Driessen

Langen, Tim

Date	2024/12/17
Version	4.0
State	Sprint E
Author	Sanchez, Francisco; Thomas, Gilton; Pestana, Cristiano; Werneck Roale, Miguel; Le, Minh

Version History

Version	Date	Author(s)	Changes	State
0.1	2024/09/10	Sanchez, Francisco; Thomas, Gilton; Pestana, Cristiano; Werneck Roale, Miguel; Le, Minh	-	Draft
1.0	2024/10/07	Sanchez, Francisco; Thomas, Gilton; Pestana, Cristiano; Werneck Roale, Miguel; Le, Minh	Research Methods; Time Plan and Milestones	Sprint B
2.0	2024/11/05	Sanchez, Francisco; Thomas, Gilton; Pestana, Cristiano; Werneck Roale, Miguel; Le, Minh	Time Plan and Milestones	Sprint C
3.0	2024/11/25	Sanchez, Francisco; Thomas, Gilton; Pestana, Cristiano; Werneck Roale, Miguel; Le, Minh	Time Plan and Milestones. Sprint C Backlog.	Sprint D
4.0	2024/12/17	Sanchez, Francisco; Thomas, Gilton; Pestana, Cristiano; Werneck Roale, Miguel; Le, Minh	Time Plan and Milestones	Sprint E

Distribution

Version	Date	Receivers
0.1	2024/09/10	Chua, Jessie; Langen, Tim
1.0	2024/10/08	Chua, Jessie; Langen, Tim
2.0	2024/11/05	Chua, Jessie; Langen, Tim
3.0	2024/11/25	Chua, Jessie; Langen, Tim
4.0	2024/12/17	Chua, Jessie; Langen, Tim

Table of Contents

Version History	1
Distribution	1
1. Project assignment	3
1.1 Context.....	3
1.2 Goal of the project	3
1.3 Scope and preconditions	3
1.3.1 Scope.....	3
1.4 Strategy	3
1.5 Research questions and methodology.....	4
1.5.1 Main Research Question	4
1.5.2 Sub-Questions	4
1.6 End products	5
1.6.1 Full-stack application.....	5
1.6.2 Documentation	6
2. Project organisation	6
2.1 Stakeholders and team members	6
2.1.1 Stakeholders.....	6
2.1.2 Team members.....	6
2.2 Communication	7
2.2.1 Microsoft Teams.....	7
2.2.2 E-mail.....	7
2.2.3 Discord.....	7
2.2.4 On-campus	8
3. Activities and time plan	8
3.1 Phases of the project.....	8
3.2 Time plan and milestones	8
3.2.1 Sprint A.....	8
3.2.2 Sprint B	9
3.2.3 Sprint C.....	9
3.2.4 Sprint D.....	9
4. Testing strategy and configuration management.....	9
4.1 Testing strategy.....	9
4.2 Test environment and required resources	10
4.3 Configuration management	10
5. Risk	10
5.1 Risk and mitigation.....	10

1. Project assignment

1.1 Context

Driessen, an employment agency for the government, education, and other meaningful sectors, has been working exclusively for vital sectors for over 30 years, and they are now one of the largest employment agencies in the Netherlands.

However, Driessen has a lot of home workers and office workers who share an office with limited space. Therefore, the team has been assigned the project to develop a system where users can reserve a table as an individual or a team; users can see who in their team is coming into the office or staying home; and users can see the availability of seats in the office space.

1.2 Goal of the project

This project's main goal is to develop a full-stack application that optimizes workspace usage and facilitates flexible work arrangements for users. By implementing key features such as scheduling, team management, room management, and room reservations, the system will allow the organization to efficiently manage their workspaces and teams. The system will enable users to:

- Reserve a workspace for the team or themselves.
- Schedule their work hours, including flexible and part-time arrangements, with the ability to mark remote or in-office days.
- Manage team memberships and roles to ensure proper collaboration and oversight.
- Enable administrators to oversee and configure users, teams, and room resources with ease.

1.3 Scope and preconditions

1.3.1 Scope

Inside scope	Outside scope
Log In	SSO (Single Sign-On)
Role Management	
Room Management	
Room Overview	
Team Management	
Team Overview	
Home/Office Overview	
Scheduling	
Reservation Management	
Notification System	
Recurring Reservation System	
Reservation Waiting List	
User Management	

1.4 Strategy

Scrum, a core component of agile methodology, divides software development into iterative cycles known as sprints, typically lasting between one to four weeks. In Scrum, teams engage in collaborative planning during sprint planning, conduct daily stand-up

meetings to track progress, and review and adapt during sprint reviews and retrospectives.

The project adopts Scrum as its framework, leveraging its adaptability, collaboration, and emphasis on delivering value in short cycles. With sprint duration set at three weeks, the team strike a balance between rapid iteration and meaningful progress, enabling continuous improvement and timely delivery of value to stakeholders.

1.5 Research questions and methodology

The research questions that are most relevant to our project

1.5.1 Main Research Question

The project aims to develop a full-stack office reservation system that optimizes workspace usage and enhances flexible work arrangements by implementing core features such as room reservations, room management, work scheduling, team management, and administrative control. To determine the optimal way to integrate these functions into a cohesive system, the team has chosen the main research question that is guiding this project:

• **How to implement an office reservation system to optimize efficiency and security while supporting room reservations, work scheduling, team management, and administrative control?**

This question is essential because it will help the team determine the optimal way to integrate these functions into a cohesive and scalable system. By answering this question, the team aims to provide an effective solution that allows the organization to efficiently manage their workspaces and teams.

1.5.2 Sub-Questions

The team aims to develop a comprehensive and effective software solution that meets the specific needs of an office reservation system by employing the following methods.

1. **How can encryption techniques be employed to protect sensitive information collected?**
 - Spring Boot (Back-end): Implement encryption methods to secure sensitive data such as personal user information. Ensure data is encrypted both at rest (in the database) and in transit (using HTTPS).
 - React (Front-end): Ensure secure transmission of data using HTTPS and handle encryption and decryption processes as needed for any sensitive user data.
2. **What methods are there for implementing a real-time notification system?**
 - Spring Boot (Back-end): Use WebSockets or an event-driven architecture to send real-time notifications about changes, such as new reservations or cancellations.
 - React (Front-end): Implement real-time notification features using components like Toast or modal popups to instantly notify users of booking updates or other important information.
3. **What measures can be implemented to ensure optimal performance when storing and retrieving data?**
 - Spring Boot (Back-end): Use efficient data structures and optimize database queries to improve data retrieval performance.
 - React (Frontend): Create forms where users can set their work schedule and sync them with the backend.

4. How can continuous integration (CI) and continuous deployment (CD) practices be implemented to automate the testing, build, and deployment processes of the software solution?

- Spring Boot (Back-end): Use GitLab CI to set up CI/CD pipelines that automate the testing, building, and deployment processes of the back end.
- React (Front-end): Automate the deployment of the front-end using CI/CD pipelines, including unit tests and integration tests to ensure reliability at each stage of development.

5. How can room availability be maximized and booking conflicts avoided?

- Spring Boot (Back-end): Implement scheduling algorithms and use database transactions to ensure that room booking conflicts are handled properly, preventing double bookings.
- React (Front-end): Create a real-time availability checker that dynamically updates to show available rooms and prevent users from selecting rooms already booked by others.

6. How can the email system improve communication for employees with different work patterns?

- Spring Boot (Back-end): Integrate email services like JavaMail or third-party services like SendGrid to automate email notifications and reminders for bookings.
- React (Front-end): Show in-app notifications and integrate with calendar APIs to remind users of bookings.

7. How can single sign-on (SSO) be added to the system?

- Spring Boot (Back-end): Implement SSO using OAuth 2.0 or OpenID Connect to allow users to securely authenticate on the system using a single login.
- React (Front-end): Integrate authentication libraries to handle SSO, allowing users to log in with their existing credentials seamlessly across platforms and services.

1.6 End products

Besides the requirements that has been handed to the team by the client, the curriculum of the Bachelor: ICT Software & Engineering establishes the end products in the deliverables of each sprint on Canvas. The following are the end products of the project.

1.6.1 Full-stack application

The team must submit a full-stack application with Minimum Viable Product (MVP) features implemented. Furthermore, the full-stack application must contain the requirements that the client has established, which are listed below:

- Login
 - Username/password
- System
 - Role management
- Room
 - Room management
 - Room Overview
- Teams
 - Team management
- User
 - Schedule
- Reservation system
 - Notification System
 - Recurring reservation system

- Admin
 - Create users
 - Link any user to any team

1.6.2 Documentation

The team must document various components of the project. The documentation that is expected to be delivered has been established in the curriculum, and the required documentation is listed below:

- Project plan
- Applied research document proposal
- Cultural awareness
- Final applied research document

2. Project organisation

2.1 Stakeholders and team members

2.1.1 Stakeholders

Name	Abbreviation	Role and function	Availability
Langen, Tim Tim.Langen@driessen.nl	-	Client	Tim Langen can be contacted throughout the week via e-mail, except for Fridays to Sunday
Chua, Jessie j.chua@fontys.nl	CHUJ01	Semester Coach	Jessie Chua will be present in class on Monday and Tuesday from 9:00 a.m. to 12:00 p.m. Otherwise, Jessie can be contacted through e-mail or Microsoft Teams

2.1.2 Team members

Name	Abbreviation	Role and function	Availability
Sanchez, Francisco f.sanchez@student.fontys.nl	FAS	Product Owner, Software Developer	Francisco Sanchez can be contacted throughout the week, except for the weekends
Thomas, Gilton g.thomas@student.fontys.nl	GT	Scrum Master, Software Developer	Gilton Thomas can be contacted throughout the week, except for Saturdays

Goncalves Pestana, Jose Cristiano j.goncalvespestana@student.fontys.nl	JCGP	Software Developer	Jose Cristiano Goncalves Pestana can be contacted on Monday, Tuesday, and Thursday after 5:00 p.m., and he can be contacted on Wednesdays and Fridays
Werneck Roale, Miguel m.werneckroale@student.fontys.nl	MWR	Software Developer	Miguel Werneck Roale can be contacted on Monday, Tuesday, and Thursday after 5:00 p.m., and he can be contacted on Wednesday except for the timeframe from 10:00 a.m. to 11:30 a.m.
Le, Minh minh.le@student.fontys.nl	HNH	Project Manager, Software Developer	Le Minh can be contacted throughout the week, except for Saturday, and Wednesday between 4:00 p.m. to 7 p.m.

2.2 Communication

2.2.1 Microsoft Teams

The team has been encouraged to contact the semester coach on Microsoft Teams whenever the team members have a question for the semester coach and cannot locate her on campus at Fontys. Moreover, whenever a team member cannot be present in class due to unforeseen circumstances, the team member must contact the semester coach to inform her of their absence.

2.2.2 E-mail

The client, Tim Langen, is available through e-mail. An agreement has been reached with the semester coach to inform the client of our progress on the product backlog, sprint backlog, and project plan. Moreover, the team can contact the client to ask for information. However, the client has requested to be contacted solely by the product owner who will represent the team.

2.2.3 Discord

Collectively, the team had decided that Discord would be the medium for communication. Discord is an instant messaging and VoIP social platform which allows communication through voice calls, video calls, text messaging, and media and files.

The team has reached the decision to use Discord on Wednesdays due to the following: most of the team actively use Discord; Discord has the functionality to create servers where you can separate channels; and Discord has the functionality of voice channels where someone can enter a voice channel and wait for others to join instead of initiating the call until everyone picks up.

2.2.4 On-campus

Even though the team meet on Wednesday on Discord, the team can be found Mondays and Tuesdays on campus. Moreover, the team has the guidance of the semester coach on Mondays and Tuesdays, hence, the importance that the entire team is present on those days.

Furthermore, the team will submit tri-weekly sprints that contains the deliverables of defined sprint on JIRA. Jessie Chua will revise these deliverables, and she will hold a feedback session with the team.

Finally, the client, Tim Langen, has formally agreed to attend a meeting triweekly. When the client attends the meeting, the team will present the progress of the project, and receive input on the deliverables.

3. Activities and time plan

3.1 Phases of the project

As the team has decided to follow the Scrum framework, the team will consider these common Scrum phases for the project:

- Initiation
- Planning and estimation
- Implementation
- Reviewing
- Releasing

3.2 Time plan and milestones

As the team transitions from the planning phase into development, the project's time plan and milestones will become more detailed. The sprint backlogs will be regularly updated based on progress, feedback, and evolving requirements. The project plan will remain a living document, ensuring that any changes to the time plan and milestones are captured as the project progresses. Below is an updated outline of the team's goals for the current and upcoming sprints.

3.2.1 Sprint A

- **Duration:** 16 September – 8 October
- Acceptance criteria met for the project's architecture, e.g., the team has discussed with the client the architectural choices the team has made for the product
- First draft of the project plan complete and reviewed by semester coach
- The online agile management environment is ready for sprint B and the proper stakeholders have access.
- The product backlog has sufficient tasks and/or user stories (US) ready for the next sprint, reviewed, estimated and accepted by the product owner
- The team has a proposal of tasks/US for sprint B based on the results from their discussions with the client and the team's Planning poker session.

3.2.2 Sprint B

- **Duration:** 9 October – 5 November
- Implement user login functionality.
- Develop team management, allowing team managers and admins to add and remove users from teams.
- Implement role management, enabling admins to assign roles to users.
- Implement user management, allowing admins to create and remove users.
- Design and implement a basic schedule overview.
- Develop room management, allowing admins to create rooms with tables, update table quantities, and remove rooms.

3.2.3 Sprint C

- **Duration:** 6 November – 26 November
- Implement user login functionality.
- Implement role management, enabling admins to assign roles to users.
- Design and implement a basic schedule overview.
- Develop team management, allowing team managers and admins to add and remove users from teams.
- Implement user management, allowing admins to create and remove users.
- Develop room management, allowing admins to create rooms with tables, update table quantities, and remove rooms.

3.2.4 Sprint D

- **Duration:** 27 November – 17 December
- Display available timeslots for table.
- Develop reservation management, allowing users and team managers to reserve tables and cancel reservations.
- Implement a recurring reservation system.
- Implement a waiting list for reserved tables.

3.2.5 Sprint E

- **Duration:** 18 December – 14 January
- Partial integration test for controllers.
- Implement a notification system.
- Implement a waiting list for reservations.
- Develop a recurring reservation system, allowing users to reserve seats in advance.
- Implement scheduling, allowing users to schedule leave days or workdays.

4. Testing strategy and configuration management

4.1 Testing strategy

To ensure that the full-stack application is functioning properly, the team intends to implement the testing that has been established in the curriculum. The testing strategies are listed below:

- Unit testing
- Integration testing
- End-to-end testing
- Acceptance testing

4.2 Test environment and required resources

A CI/CD pipeline will be initialized in GitLab. Moreover, the team will incorporate SonarQube into the CI/CD pipeline, which enriches the testing environment by providing static code analysis and code quality metrics. SonarQube helps identify and mitigate code smells, bugs, and security vulnerabilities early in the development process.

By integrating SonarQube into the pipeline, it ensures that the code meets high-quality standards and is more resilient to potential issues, enhancing the overall reliability and maintainability of the full-stack application.

4.3 Configuration management

GitLab, a foundational tool in higher-level configuration management, will be used for the project. The software solution must be uploaded onto GitLab, where stakeholders can clone the repository.

The team must commit each feature of the software solution onto GitLab, but these changes must be committed in their appropriate branch. Only when these branches are completed can they be merged into a branch where these changes in code will be tested.

Finally, when the code has been tested, it will be merged into the main branch of the GitLab repository. This will ensure that the main branch will remain bug-free. Moreover, it will result in a clear overview of the changes made throughout the project, with each feature having a timestamp of when it was committed.

5. Risk

5.1 Risk and mitigation

The risks that could affect the project throughout its duration have been investigated and identified. To prevent the risk from being actualized, the prevention activities have been defined. Moreover, in case the risk eventuates, the mitigation activities have been defined. The following is a list of identified risks alongside the prevention activities and mitigation activities.

Risk	Prevention activities	Mitigation activities
1. The project team added their own product features that are not in requirements or change request	The requirements given by the client will be placed higher in priority than the features of the team	The project manager, product owner, and scrum master will investigate the cause to prevent it from occurring once more, and henceforth, the team must follow the prevention activities
2. Estimates are inaccurate	Prioritizing the functionalities will help in identifying the tasks that should be completed first	The product owner and the scrum master must enter discussion to re-evaluate the estimates
3. User Interface does not allow users to complete tasks	End-to-end tests will be conducted to ensure that the application functions as intended	Tasks that cannot be completed must be revised and fixed or placed on higher priority in the consequent sprint
4. Inputs from stakeholders that are low quality	The product owner will communicate with the client	The product owner can contact the client whenever

	to elaborate on certain aspects of the project	uncertainties arise, and the team members require a higher quality input to continue with their tasks
5. Project team misunderstand requirements	Throughout the semester, at the end of each sprint, a sprint review is held where a clarification can be given on the unclear requirements. Frequently, stand-up meetings will be held to ensure that team members do not face uncertainties in their tasks	The product owner must contact the client to receive a clarification on the unclear requirements
6. Under communication within the team	The scrum master has been tasked with holding stand-up meeting to keep the team members informed of all occurrences in the project	The project manager has been tasked with ensuring that the scrum master and product owner are executing their tasks
7. Impacted individuals are not kept informed	The product owner has the responsibility to keep impacted individuals informed on the progress of the project	The team members must confront the product owner who has failed to perform his responsibilities. Moreover, a choice could be made to remove the role from the team member
8. Learning curves lead to delays	The team member should contact the team to request help with their task. Tasks can be exchanged within the group if it does not lead to delays	Delays must be kept into consideration during the planning of a sprint
9. Team members who perform below expectations	Stand-up meetings will be held frequently to inform team members on the progress of the tasks	The team will confront the team member to resolve the conflict internally. If the conflict could not be resolved internally, then the semester coach will be informed about the situation
10. The team lacks motivation	The team has come to an agreement that their motivation dims at the end of the semester, therefore, the workload at the semester will be larger to reduce the workload at the end of the semester	The project manager, product owner, and scrum master must ensure that the workload has been completed at the end of each sprint
11. Technology components have security vulnerabilities	The top 10 list of OWASP informs software developers of the most critical security risks to web applications, and it provides preventative activities along with each item on the list	The team will refer to the top 10 list of OWASP to follow the preventative activities. Furthermore, the team must conduct research on the security vulnerabilities to understand the problem that must be solved