Graduation report

Internship Maarten Hormes

Hour Analysis - Quad Solutions

Eindhoven

Date	**:**	**12/06/2024**
Version	**:**	**1.2**
Status	**:**	**V1.2**
Author	**:**	**Maarten Hormes**

FONTYS UNIVERSITY OF APPLIED SCIENCES **HBO-ICT: English Stream**

Data student:	
Family name, initials:	**Hormes, M.**
Student number:	**456806**
project period: (from – till)	**19/02/2024 – 12/07/2024**
Data company:	
Name company/institution:	**Quad Solutions**
Department:	**ICT**
Address:	**Fellenoord 39 unit 3.02W, Eindhoven**
Company tutor:	
Family name, initials:	**De Vaal, Wouter**
Position:	**Team Lead / senior java engineer**
University tutor:	
Family name, initials:	**Metaxas, G.**
Final report:	
Title:	**Graduation report Maarten Hormes**
Date:	**12/06/2024**

Approved and signed by Fontys tutor:

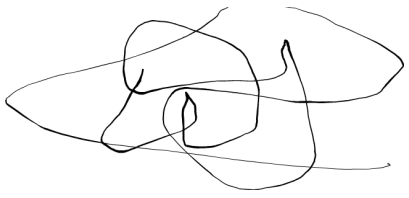
Date:

Signature:

Approved and signed by the company tutor:

Date: June 17th, 2024

Signature:



Version

Version	Date	Author(s)	Amendments	Status
0.1	05/03/2024	Maarten	Setup internship report	Initial draft
0.2	22/03/2024	Maarten	Added research for SQ 1	First version
0.3	26/03/2024	Maarten	Added research for SQ 2	First version
0.4	10/04/2024	Maarten	Added research for SQ 3	First version
0.5	27/05/2024	Maarten	Added research for SQ 4	First version
0.6	28/05/2024	Maarten	Updated results sections after feedback from Metaxas. Changed appendices to use links to files instead of object imports.	First version
0.7	03/06/2024	Maarten	Added research for SQ 5	First version
0.8	05/06/2024	Maarten	Added conclusion and recommendations	First version
1.0	07/06/2024	Maarten	Added summary and reflection	V1 done
1.1	12/06/2024	Maarten	Moved LO evaluation into this file. Adjusted some small sections after feedback	V1.1
1.2	12/06/2024	Maarten	Adjusted images to align with remainder of file	V1.2

Distribution

Version	Date	To
0.4	29/04/2024	Metaxas, G.
0.5	27/05/2024	Metaxas, G. De Vaal, W.
0.6	28/05/2024	Metaxas, G.
1.0	07/06/2024	Metaxas, G.
1.1	12/06/2024	Metaxas, G. De Vaal, W.

Table of content

Summary

This document was created during my internship period at Quad Solutions in Eindhoven. Both this document and the performed research are intended to give an answer to my main research question, being:

How can Quad Solutions' existing technologies be integrated, in order to create and deploy a centralized hour analysis tool, complying with both the constraints and requirements constructed by the stakeholders as the industry standards for web development?

The integration of Quad Solutions' existing technologies required a detailed understanding of the current systems, stakeholders' needs, and the best practices in the industry.

To ensure the project met all requirements, a stakeholder analysis was conducted using field research methods, including interviews and requirement explorations. This approach provided an understanding of the stakeholders' needs and allowed me to setup a requirements list.

In choosing the appropriate JavaScript framework for the front end, a combination of library research, product analysis, and community research was used to develop a technology matrix. This matrix evaluated frameworks based on performance, ease of use, community support, and compatibility. React.js was selected for its strong alignment with these criteria and its proven track record in similar projects.

The integration with existing tools, particularly the hour logging tool Yoobi, was guided by industry standards and practices identified through a literature study and IT architecture sketching workshops. These methods ensured a robust and future-proof integration, documented for seamless handover. The decision to use CSV integration instead of API integration was based on the company's concern about exposing sensitive information through the API, as highlighted in stakeholder interviews.

Another thing that played a crucial role in evaluating the integration's effectiveness was testing. The testing strategy included component tests and system tests to ensure the quality of the code and the reliability of the tool under real-world conditions. End-to-end (E2E) tests were not conducted due to the specific nature of the user interface, which primarily involved simple data presentation and interaction. The UI's simplicity and the straightforward interaction model made E2E testing less critical, allowing the focus to be on component and system-level tests, which were more relevant for validating the core functionalities and integrations of the tool.

Deployment was guided by Quad Solutions' standards and insights from an expert interview. A deployment guideline was constructed, outlining specific steps and requirements for deploying the application in the company's cloud environment. This included the use of GitHub Actions for continuous integration and deployment, ensuring that the application could be reliably and consistently deployed. Additionally, deployment was carried out using Heroku and Vercel for hosting the backend and frontend, respectively. A guidelines conformity analysis was performed to ensure that the deployment process adhered to Quad Solutions' standards and practices.

As a result of this research and implementation, I integrated Quad Solutions' existing technologies to develop a centralized hour analysis tool that meets both stakeholder requirements and industry standards. The chosen framework, React.js, was selected for its performance, ease of use, and strong community support, ensuring a robust and user-friendly front-end. The decision to use CSV integration for data handling was based on careful consideration of data protection concerns, as highlighted in stakeholder interviews. Thorough testing, including component and system tests, validated the tool's functionality and reliability under real-world conditions. The deployment process was guided by Quad Solutions' standards and insights from expert interviews, ensuring a smooth and consistent deployment using GitHub Actions for continuous integration and deployment.

Glossary

Possible abbreviation	Term	Meaning
CI/CD	Continues integration / Continuous deployment	A software development practice that automates the process of integrating code changes and deploying them to production
CSV file	Comma Separated Values file.	A document with values separated by commas, with each row being a new entry.
E2E	End-to-End	Often used to refer to testing. From one end of the application to the other end
-	Framework	A collection of JavaScript libraries, providing developers pre-written JS code for easy development.
JaCoCo	Java Code Coverage	A tool used to investigate the by tests covered code (paths) in Java applications
POJO	Plain Old Java Object	A simple object in Java.
RBAC	Role Based Access Control	A form of security that uses roles to specify which group of users is allowed to access certain parts of the system.

1. Introduction

This report concludes my graduation internship at Fontys ICT. The internship was held at Quad Solutions in Eindhoven, where I worked independently from their software development teams and worked on an in-house system. The internship was held over a period of 5 months.

Quad Solutions can be described as a company that delivers full java development teams to their clients instead of individual people. They believe that building solid teams is unbelievably valuable as they know they people they pair up in a team, allowing them to ensure the functioning of these teams.

The overall goal of the internship was to support the development of an hour analysis tool. This should be integrated with already existing technologies at Quad Solutions, in order to create a dynamic dashboard that allows management to analyse the hour distribution of their development teams.

The following chapters of the report discuss the company, assignment, research, and conclusions in more detail.
Chapter 2 gives information about Quad Solutions, the company where I performed my internship, and my part in the company.
Chapter 3 goes over the assignment tied to this internship, the research involved and further related information.
Chapter 4 holds the research process & results, going over the how and why of my research findings and conclusions.
Chapter 5 concludes the research related section of the report, formulating a conclusion and further recommendations to Quad Solutions.
Chapter 6 contains an evaluation for all the learning outcomes, and a general personal evaluation of the project.
Lastly, you can find the appendices.

This report is written with my research documentation as main source. This research documentation, which can be found in Appendix A through J, is the place where I stored all the research during my internship. Referenced articles, sites and people are documented in the references list of the related research documentation.

In this document, there are little references since all information comes from my research documentation. References will mostly be made to these documents (or the appendices they can be found in). Interested parties can use this research documentation to find the references used during my research.

2. About the company

Quad Solutions is an IT consultancy company in Eindhoven, which specializes in building Java development teams. They see a lot of value in supplying a full team instead of individual developers, opposed to how most consultancy companies work. They realize how much more powerful a team is that can work together and contains people with similar interests and complementary skills.

Clients they work for include de Belastingdienst, Philips and ASML.

More information about my assignment can be found in chapter 3: Assignment overview.

3. Assignment overview

3.1. Current situation

Quad Solutions has contracts with their clients. These contracts include an estimated number of hours to be worked over the duration of the contract. Employees of Quad, who are working at the client, are expected to register their hours with the client's company, and in Yoobi (the hour logging tool used by Quad Solutions).
Quad Solutions can use Yoobi to track the number of hours used by individual employees, but cannot see how that relates to a team, the contracts related to this team, the number of hours left or any other future projections.

3.2. Opportunities

As mentioned, Quad Solutions supplies full Java development teams to their clients. However, supplying a full team instead of an individual is not always supported by the infrastructure of the client's company. They cannot book hours for a team, but they work on an individual level. Where Quad Solutions is booked on a contract basis, all employees have an individual hour registration.

By working, employees slowly use up the hours in the contract. Because this number of hours disclosed on the contract is an estimation, there is an abundance of things that can happen that result in the actual hours used/worked does not align with the contract hours. Employees being sick, going on a holiday, not using holiday days, changing their weekly hours are just some examples.

Due to these circumstances developers can 'use up' their hours before the end of the contract period. These developers still get paid by Quad Solutions, but do not earn any revenue anymore (since the hours of the contract have been used up, the client only pays for the hours in the contract).

This becomes more of an issue when the next contract period is started. Developers who worked a lot during period 1 might want some extra days off, resulting in the team not using all their hours (making it so that they also don't get paid for those 'missing' hours). Quad Solutions wants to prevent both situations.

3.3. Project goal

The goal of the project is to create a system that is connected to the hour registration tool Yoobi and the system holding the contract information. The combination of this, and the holiday schedules will result in a tool/system that can be used to track the worked and expected to be worked hours.
By using this system, the management of Quad Solutions can investigate and anticipate how their developers are using their hours. If they end up using too much or too little hours during a specific period (or if the analysis in the new system point this out) they can adjust their developer's hour distribution.

This system developed can be expanded upon, as necessary. During my first conversations with one of the founders, it became clear that there are more wishes than this hour analysis mentioned. These expansions can include general employee management, tools to view salaries and other projections (holiday money, 13th month, etc.), or the possibility to view the revenue generated per employee.

For the original scope of my assignment, I have been working on the hour analysis tool mentioned, although this plan did include the opportunity to expand the system if my progress allowed it.

One of the bigger challenges posed is that this system must be connected to the existing systems Quad Solutions uses in-house to track worked hours, holiday requests, and contracts with clients, as the systems used by the clients. Next to integrating with existing software, the system needs to be hosted in Quad Solutions environment. Meaning that I need to deliver a production-ready solution, which can be integrated seamlessly in the already existing cloud stack of Quad solutions.

3.4 Requirements

After discussing the possible research and assignment for my internship with Quad Solutions, they have constructed the first requirements to follow. These are listed below and were used as the basis of my assignment:

- An integration with Yoobi (hour logging tool) has to be included in the end product.
- The contract specifics of a team can be manually configured.
- The exclusive usage of widely used software solutions, e.g. frameworks, plugins, etc.
- Deployment according to Quad Solutions' standards and guidelines.

3.5 Constraints

For this project, there are some technologies that I used since Quad Solutions already decided to use them as the basis technologies for their projects.
This means that there are the following constrains:

- The back end is written in Java (Maven) using Spring Boot
- The JavaScript framework used for the UI has to be either React.js or Next.js

3.6 Deliverables

An overview of the deliverables for this project can be found below. Each deliverable is shortly described below the image.

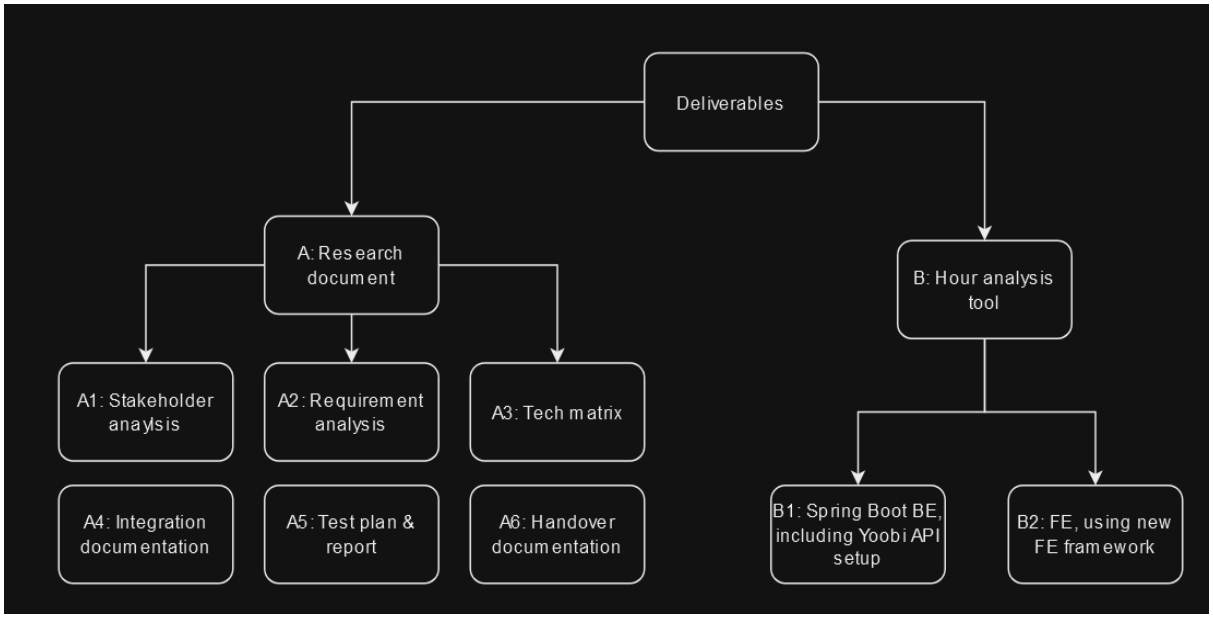


Fig. 1: Deliverables

A: Research documentation. A substantial portion of my project is the investigation and application of multiple topics. This research performed resulted in multiple forms of documentation coming to life. The first part (A1 – A3) are documents that are the result of my first research topics, defining the scope, identifying the stakeholders, constructing requirements, finding the right technologies, etc. The second part (A4 – A6) are documents created when looking into the actual implementation of the assignment or my new-found knowledge. An integration document was made to document the integrations made with the hour logging tool, a test plan indicating how my solution is tested and the report showing the test results and a handover guide, indicating how the company can continue to work on and deploy my solution, while complying with all deployment guidelines and standards at Quad Solutions.

B: Hour analysis tool. The second part of the deliverables is the product I have created for Quad Solutions during my internship period. This includes both the in Java (Spring Boot) written backend, as the UI. The UI is written in React.js.

3.7 planning

For this project, the planning/methodology is similar to a waterfall methodology. By planning out the research I performed and estimating how long individual research sections and implementations will take, I had a good understanding of how I should divide the work over the timeframe available. This resulted in a waterfall like situation where I created a rough weekly planning indicating which activities I should be performing.
Since software development often is not a straight line of progress, a lot of teams have adapted an agile way of working. Even though I had a rough planning for the complete project, I worked with an agile mindset. This means I regularly checked my progress to ensure I was still on track to complete the project, made adjustments where necessary and found feedback loops with both my school tutor as my company mentor.

4. Process and results

This chapter of the report is used to describe the process and results of my research. It is structured around the research questions and explains how and why the conclusions and its supporting implementations of each sub-question came to be. Most of these questions have some technical details. All technical detail mentioned is used to describe how and why certain implementation came to be and will not be used to dive into each little bit of the implementation.
Each of these sub-questions contribute to answer the main research question:

How can Quad Solutions' existing technologies be integrated, in order to create and deploy a centralized hour analysis tool, complying with both the constraints and requirements constructed by the stakeholders as the industry standards for web development?

The sub-questions used to formulate an answer to the main question:

1. Who are the stakeholders, and what are their requirements for the hour analysis tool specified by Quad Solutions?
2. What is the best fitting JavaScript framework to develop the dashboard of the hour analysis tool for Quad Solutions?
3. What industry standards and practices can be applied to ensure integration with Quad Solutions' hour registration tools?
4. How can performance and quality testing be applied to evaluate the integration of the hour analysis tool with Quad Solutions' existing systems, focusing on the Spring Boot backend?
5. What are the guidelines set by Quad Solutions when it comes to deploying applications to the cloud, and how can compliance be assured?

Research approach

At the beginning of my internship, I defined the research approach per sub-question. This expected approach can be found as the first part of each of the sub-questions. This expected approach is supplemented by information on how I actually came to an answer and a conclusion to the sub-question in question. All these expected and actually used approaches come from the DOT framework. During this internship I have used multiple research methods from all the categories of the DOT framework. This allowed me to find the suitable research methods for each part of the research, design and development process.

4.1. Who are the stakeholders, and what are their requirements for the hour analysis tool specified by Quad Solutions?

I will be using the explore user requirements method from the field strategy, as the stakeholder analysis method, also from the field strategy.

In order to develop a solution that complies with the requirements of all stakeholders, it is an important step to identify the stakeholders, their involvement and wishes. This was done by having an interview with my client, identifying his needs and other stakeholders. By compiling this information, it was possible to construct a list of requirements that the solution needs to comply with. A similar result could be achieved by performing user requirements explorations, although this will give a less complete understanding of all stakeholders' requirements. This information would be gathered using interviews, since it's a fast way of getting direct input & feedback.

As expected, the starting point for this project is a stakeholder analysis. I performed an interview with my conceptual coach at Quad Solutions (Christopher) to gain an overview of stakeholders involved in the project and the tool I will be developing for them. Based on this conversation I concluded there were 2 other stakeholders I should be talking to. These are my technical tutor, Wouter, and the contact person to the company from the main hour logging tool used at Quad Solutions, Harm-Jan.

Since this tool is an inhouse tool that will only be used by Christopher, the list of stakeholders is quite short and simple. This also leads to a simple power-interest matrix that was created to visualize the stakeholders and their influence on the project (see figure 2).

From this image we can see how the three interviewed stakeholders are involved in this project. Harm-Jan is involved, but only as the contact person for Yoobi. Wouter is more involved as my technical tutor, but will not be using the tool, not constructing any requirements. Christopher on the other hand is the most important stakeholder in the project. He will be the one using the tool and is the conceptual lead or even product owner (The full analysis can be found in appendix B).

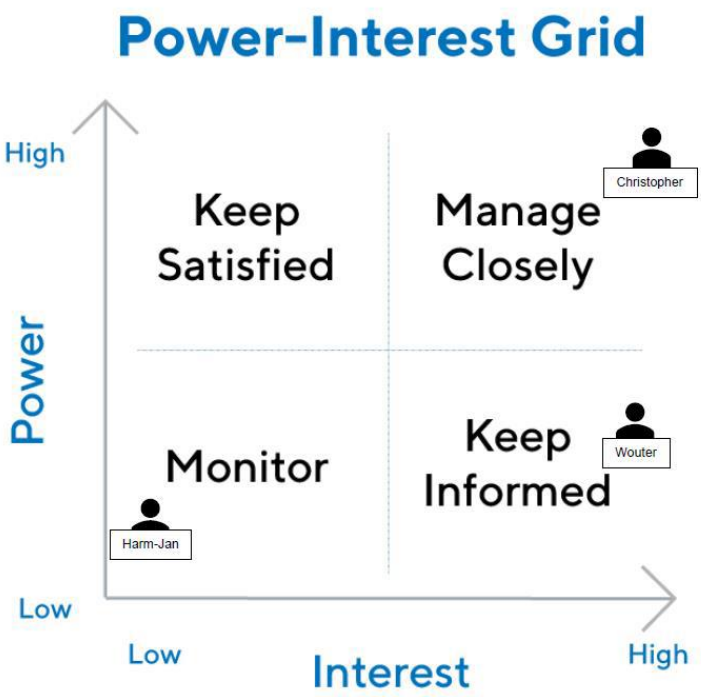


Fig. 2: power-interest matrix

After having talked to the stakeholders to gain an understanding of their involvement, I performed extra interviews to gain an idea of their requirements and priorities. When saying extra interviews these were mostly with Christopher since the other conversations concluded quite soon as the stakeholders pointed out they will barely be involved in the project or with the tool (Interviews can be found in appendix D).

After these conversations I composed the information coming from them and managed to create the original list of requirements. The reason why I call it an original list is that often the requirements change during the development process. These requirements stated were the starting point and contains the most minimalistic functionalities (The full analysis can be found in appendix C).

Below you can find an overview of these original requirements.

Functional Requirements:

- Login page to protected view.
- Option to import CSV files.
- Possibility to extend data or replace data on new import.
- Have the option to configure the contract specifics of a Team that is being analysed.
- Be able to adjust the time frame in which the hour registrations are analysed.
- View a team overview as well as individual members inside a team.
- The UI design needs to align with the styling of other Quad Solutions tools.

Non-functional requirements:

- Both the BE and FE need to be password protected.
- RBAC needs to be setup to create fine-grained security configuration.
- Asynchronous communication is used for better performance.
- The tool includes an SSL setup to enable HTTPS (not needed in dev environment)
- Integration to Yoobi will be made using the CSV export files.
- Data persistence is achieved using Postgres on Heroku
- Application gets deployed to Heroku and Vercel for BE and FE, respectively.
- Tailwind CSS is used for inline styling.

4.2. What is the best fitting JavaScript framework to develop the dashboard of the hour analysis tool for Quad Solutions?

For this research I will be using two methods from the library research strategy. These are the best, good and bad practices, and the available product analysis methods. The results can be presented using the pitch method from the showroom research strategy.

When it comes to JavaScript frameworks, there are a lot of constant developments. This results in a period where choosing a JavaScript framework for frontend development is no easy task. By looking at the best, good and bad practices of frontend development, and combining this with an available product analysis I can construct a technology matrix that incorporates multiple JavaScript frameworks, to choose the best fitting one for the hour analysis tool. Possible criteria to look at include performance, ease of use, community support, complexity, compatibility with other technologies or the ease of handing over the project to Quad Solutions.

The first step of this research is to perform an investigation into good best and bad practices and the already available products to determine which frameworks should be evaluated to find the best fitting framework. During this first investigation I investigated frameworks that are widely used, future proof, and most importantly, used for similar kind of web development. Combining the information from this investigation with the constraints already laid down I came to a list of 5 frameworks that I would be comparing to find the most suitable (see appendix E for more detailed information on why these 5):

- React.js
- Next.js
- Angular
- Vue.js
- Svelte

Each of these frameworks, including React.js and Next.js, offers unique advantages that can be leveraged depending on the specific needs and goals at Quad Solutions. When choosing a framework, it's important to consider factors such as the project's complexity, performance requirements, development team's expertise, and long-term maintenance. In order to create an easy comparison for these frameworks, I created a matrix, rating each of the mentioned frameworks on experience, ecosystem & community support, flexibility, scalability & performance and the ease of development. In this matrix you can find an explanation for each of the criteria, supporting the ratings for each of the frameworks.

By adding a weight to each of the criteria I can adjust how important certain criteria are. These weights are pretty equal, expect for the experience criteria (which is a bit lower since experience can always be learned by working with the framework) and the ecosystem & community support criteria (which is a bit higher since this criterion includes online help when needed and already created projects and demos).

From this matrix, there are a few conclusions to draw. The first being; when comparing these 5 frameworks (React.js, Next.js, Angular, Vue.js, Svelte) on the 5 topics; experience, ecosystem & community support, flexibility, scalability & performance and ease of development, the highest scoring framework is **React.js**.

Next.js loses its only points (compared to React) on flexibility. Although it is built on top of React, there are specific use cases and architectural patterns that Next.js applications benefit from. This hour logging tool will not use these features (such as SSR).

Vue on the other hand, being an independently developed framework, does not deliver as well as React when it comes to the ecosystem and community support. Since this project is intended to be continued even after my graduation assignment, choosing a framework with a bigger ecosystem and community support is quite valuable.

The last high scoring option is Svelte. Although Svelte scores very high on the flexibility and scalability (which is what the framework was designed for), it falls short on both the experience part (looking at both my experience with the framework as the experience at Quad Solutions when they will take over the project) and its ecosystem and community support. As it is a newer framework. There are less projects performed in it, companies adopting it in their daily work and guides/tutorial to be found. These downsides result in Svelte scoring lower than React.js, although it is the second highest scoring framework in this specific matrix.

Seeing where the other frameworks fall short compared to React.js leads to the conclusion that React.js is the most suitable framework for developing the dashboard of the hour analysis tool.

See Table 1. Tech matrix for the point comparison. See appendix E for the full analysis or the table with explanation for the points scored.

Criteria	Weight	React	Next	Angular	Vue	Svelte	
Experience	0.15	5	4	3	4	4	
Ecosystem & community support	0.25	5	5	4	4	3	
Flexibility	0.20	4	3	3	4	5	
Scalability & performance	0.20	4	4	5	4	5	
Ease of development	0.20	4	4	3	4	4	
Total	1	4.4	4.1	3.7	4.0	4.2	**Conclusion: React.js**

Table 1. Tech matrix

4.3. What industry standards and practices can be applied to ensure integration with Quad Solutions' hour registration tools?

I will be using the literature study from the library strategy to investigate integration possibilities, as use the IT architecture sketching from the workshop strategy to visualize the integration.

By looking into the tools used by Quad Solution, I can start investigating the integration possibilities with the Java (Spring Boot) based backend I have developed. The goal of this investigation is to create an implementation that follows both the best practices as well is future proof (this also includes a seamless handover to Quad Solutions). This is documented in an integration document (see appendix F), specifying how the integration was made. During this investigation I used architecture sketching to visualize the system I developed and the integration to the external tools.

During my first conversations with **Christopher,** it became clear that the in-house hour logging tool used is Yoobi. Their clients do use different tools to log the hours in their system, but I won't be able to gain access to those systems. A combination of all hours logged in those systems should be available in Yoobi since this is the only in-house tool used. Another take away from this conversation is that the Christopher has the expectation that the integration happens seamlessly, meaning that no user interaction is required to gain the latest information from Yoobi. The goal of the tool is to gain a quick overview of the hour distribution in a contract (period). To accomplish this, the tool needs an intuitive UI that shows the relative information on first glance, and the amount of user actions (clicking buttons, filling in fields, dragging files) needs to be reduced to a minimum. This requires some automation in order to satisfy the criteria (like API polling).

Next to my conversation with Christopher, I spoke with **Harm-Jan** as he is the contact person for Yoobi. From this conversation it became clear that Yoobi does have an API that we could use, but also that it exposes too much sensitive information for Harm-Jan to just hand over the access. Instead, he was capable of sending me an exported Excel sheet (see figure 3) containing all hour logs without the sensitive data.

See appendix D for the full conversations.

This caused quite the tricky situation where; a stakeholder wants an automated integration using the API possibilities Yoobi has to offer, and another stakeholder does not want to give up the information included in the API.
After some conversation with Christopher, we decided to create a first version of the tool using the excel sheet as basis. Since there is still the need of creating an automated integration, this can be investigated on a later stage. Either as part of this research (meaning this research question could be revisited) or as part of a new project/assignment. One thing to note is that even though both integration options differ quite a lot, the data coming from Yoobi, and the operations performed on the data in the BE of the new tool are the same. Meaning that not being capable of integration the API does not affect any other part of this project.

Since we settled on creating a first version with the excel integration, I investigated how to properly integrate this into a Spring Boot back end. In the following section I explained some of the technical details of this integration. See appendix F for the full implementation research.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Medewerker	pers_num	project_naar	activiteit_na	datum	uur	opmerking	soortuur	bijgewerkt_tot	acc_datum	projectmanager	Goedgekeurd door	Goedgekeurd op	
2	Alessandro P	2109018	BLD - Aless	Forced day	23/Aug/23	8,00		time	23/Aug/23	23/Aug/23		Finance Quad	04/Sep/23	
3	Alessandro P	2109018	BLD - Aless	Forced day	12/Sep/23	8,00		time	12/Sep/23	12/Sep/23		Finance Quad	19/Sep/23	
4	Alessandro P	2109018	BLD - Aless	Forced day	13/Sep/23	8,00		time	13/Sep/23	13/Sep/23		Finance Quad	19/Sep/23	
5	Alessandro P	2109018	BLD - Aless	Forced day	14/Sep/23	8,00		time	14/Sep/23	14/Sep/23		Finance Quad	19/Sep/23	
6	Alessandro P	2109018	BLD - Aless	Forced day	15/Sep/23	8,00		time	15/Sep/23	15/Sep/23		Finance Quad	19/Sep/23	
7	Alessandro P	2109018	BLD - Aless	Forced day	04/Oct/23	8,00		time	04/Oct/23	04/Oct/23		Finance Quad	19/Oct/23	

Fig. 3 Excel sheet snippet

When it comes to working with excel sheets in Spring boot there are some options. One of the options is use Apache POI. The default option for working with .xls (or .xlsx) files in Java. This would require me to write an adapter that could iterate over the rows and columns of the sheet to fetch the information (if present) and map it into an object. Having worked with this kind of code before, I did not want to go for this approach. It includes writing a lot of code to ensure that the information fetched from the sheet is in the right format and right place (as in column indexes), and a lot of maintenance if the structure of the file changes. Of course there are other options.

These include libraries as JExcel and FastExcel. All though these should simplify the integration process that pushes me away from Apache POI, these libraries are not as complete, missing support for newer file types (xlsx) or just changing the code structure (the way of working is similar).

This pushed me to investigate other integration possibilities for the excel file. One of the options I discovered was the OpenCSV library. All though it does not support xls file, it does support CSV files. The beauty of the OpenCSV library is the option to map the CSV file into beans, which are implemented as annotation POJOs. This allows for quick integration and adaptability. We can tell our code to take values from the CSV files based on the header names. In comparison to with Apache POI where we have to fetch the rows based on column indexes. Now we can just specify in the bean what the header name is and OpenCSV takes care of the rest.

The structure of this bean can be found in figure 4.

```
@Getter  6 usages  Maarten Hormes *
@Setter
@AllArgsConstructor
@NoArgsConstructor
public class CSVObject {

    @CsvBindByName(column = "Medewerker")
    private String employeeName;
    @CsvBindByName(column = "pers_num")
    private int employeeNumber;
    @CsvBindByName(column = "project_naam")
    private String projectName;
    @CsvBindByName(column = "activiteit_naam")
    private String activityName;
    @CsvCustomBindByName(column = "datum", converter = LocalDateConverter.class)
    private LocalDate date;
    @CsvBindByName(column = "uur")
    private double hours;
    @CsvBindByName(column = "opmerking")
    private String remark;
    @CsvBindByName(column = "soortuur")
    private String typeOfHours;
    @CsvCustomBindByName(column = "bijgewerkt_tot", converter = LocalDateConverter.class)
    private LocalDate lastEdited;
    @CsvCustomBindByName(column = "acc_datum", converter = LocalDateConverter.class)
    private LocalDate accDate;
    @CsvCustomBindByName(column = "Goedgekeurd op", converter = LocalDateConverter.class)
    private LocalDate approvalDate;
}
```

Fig. 4 Bean structure

As you can see in figure 4, specifying the column name is enough for OpenCSV to map the entrees as shown in figure 3. Each column from the excel file is mapped into one of the variables in figure 4. Once these values are available in a POJO, we can create the needed objects to save each entry into the database.

This is done by the mapper function shown in figure 5. This function iterates over each of the CsvObject beans in order to create the domain objects; Employee and HourEntry. It checks if the user already exists, otherwise it creates the user. Then it creates the HourEntry object and inserts the employee object in there. Once all beans have been mapped to domain objects this function returns a list of HourEntries. This list is returned to the function managing the import so that it can save this list of entities in the database. See appendix F for a more detailed explanation of this function.

```
private List<HourEntry> processCSVData(CsvToBean<CSVObject> csvToBean) {  1 usage  Maarten Hormes
    List<HourEntry> returnList = new ArrayList<>();

    //save last emp number to reduce db calls
    int previousEmployeeNumber = 0;
    Employee emp = null;

    // Process the CSV data into domain objects
    for (CSVObject csvObject : csvToBean) {
        if(previousEmployeeNumber != csvObject.getEmployeeNumber()) {
            previousEmployeeNumber = csvObject.getEmployeeNumber();
            emp = employeeService.getEmployeeByEmployeeNumber(csvObject.getEmployeeNumber());
            if (emp == null) {
                emp = Employee.builder()
                        .employeeNumber(csvObject.getEmployeeNumber())
                        .name(csvObject.getEmployeeName())
                        .hourEntries(new ArrayList<>())
                        .build();
                employeeService.saveEmployee(emp);
            }
        }

        HourEntry hourEntry = HourEntry.builder()
                .id(UUID.randomUUID())
                .employee(emp)
                .projectName(csvObject.getProjectName())
                .activityName(csvObject.getActivityName())
                .date(csvObject.getDate())
                .hours((int)csvObject.getHours())
                .remark(csvObject.getRemark())
                .typeOfHours(csvObject.getTypeOfHours())
                .lastEdited(csvObject.getLastEdited())
                .accDate(csvObject.getAccDate())
                .approvalDate(csvObject.getApprovalDate())
                .build();

        returnList.add(hourEntry);
    }
    return returnList;
}
```

Fig. 5 Mapper function

Even though Apache POI can be used to work with bigger or complex excel sheets, it creates a lot of overhead in maintenance. This led me to investigate other possibilities. The eventual chosen library to use is OpenCSV due to its possibility to map rows of a CSV file into POJOs, reducing the amount of code that needs to be written. Since this mapping happens based on header names, it costs way less time to maintain this code compared to an Apache POI variant. By using the OpenCSV library, I can create simple functions that take the CSV file, iterate through the rows and map each of the comma separated values to the correct variable of the object it gets mapped to.

All though this does allow integration between Yoobi and the Hour analysis tool, this does still require user interaction as someone has to export the file from Yoobi and upload it into the new tool. In order to prevent these steps from being taken, an integration with the API should be made. Once the API connection is available, some sort of API polling (or a fetch button) can be implemented to seamlessly get the information from Yoobi and start the mapping process without any user interaction (or just one button press).

In case the API does became available during the period of my internship I can extend upon this research to include the API integration created.‘

4.4. How can performance and quality testing be applied to evaluate the integration of the hour analysis tool with Quad Solutions' existing systems, focusing on the Spring Boot backend?

In order to test the system properly, I can use multiple methods from the lab strategy. These being, component testing, system testing and non-functional testing. This can be expended upon using the static program analysis method from the showroom strategy.

To ensure that the integration to the existing systems works according to the requirements and under load I can use a wide range of tests. Specific unit or component testing can ensure that both the smaller units of the software, as the integrations work as expected. System tests can be used to validate the functionality of the complete system. Another option would be to perform non-functional tests to test the system in a real-life simulation. A test plan was constructed indicating how these tests should be performed. Next to testing my code, I can perform a static analysis, checking the quality of the written code based on coding standards, best practices and guidelines.

To investigate which testing solutions would be suitable for ensuring quality and performance, I dove into the Best, Good and bad practices, and industry standards for testing Spring Boot applications. This investigation led to the creation of a test plan including 4 stages of testing. This are Unit testing, Integration testing, API testing and E2E testing.
The idea of this plan is to setup a strategy which both tests the system thoroughly and is supportive of the iterative cycle software development often takes place in.
Although the plan can be found in Appendix G, I'll supply an overview of what these different types of tests are intended to test.

Unit Testing: This phase involves testing individual components or functions without external dependencies. The aim is to verify that each unit operates correctly in isolation, catching low-level bugs and issues early in the development process.
Integration Testing: This step tests the interactions between integrated units to ensure they operate together seamlessly. It focuses on data flow and dependency management between modules, critical for the overall functionality of the application.
API Testing: This testing ensures that all API endpoints respond correctly under various scenarios, handling both expected and unexpected requests effectively. Tools like Postman or Swagger will be employed to automate these tests, ensuring robustness and consistency in API behaviour.
End-to-End Testing: Simulating real-world user scenarios, this testing evaluates the complete system's functionality. Using tools such as Cypress, it verifies that the application meets external usability and interaction standards expected by end-users.

After the testing strategy was created, I could start with the implementation and documentation of these tests. In the following section I will summarize the test results. A full report of this can be found in Appendix G. In order to visualize the test results and coverage, I used screenshots from IntelliJ and JaCoCo.

Unit & Integration Tests

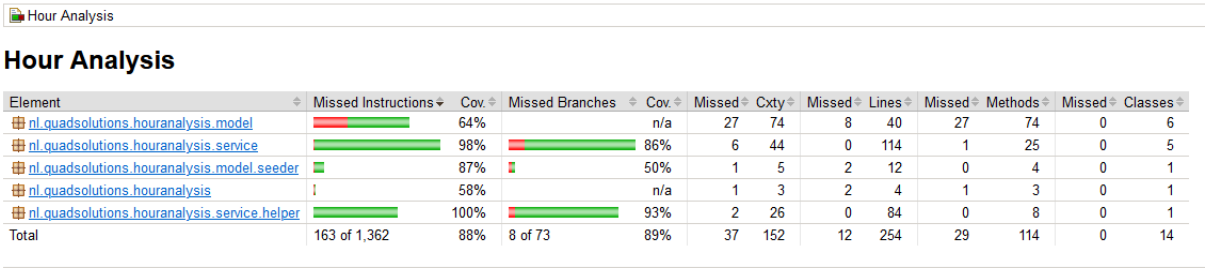
Unit and integration tests were grouped together due to the closely related nature of their objectives and the structure of the codebase. This approach was chosen because many functions, particularly those involving interactions with repositories, have limited standalone testing value. By combining these tests, we ensured a more holistic assessment of both isolated functionalities and their interactions within the system.

Unit tests focused on small pieces of isolated code to ensure functionality in isolation, catching low-level bugs. For example, the AppUserService includes a validation function to ensure only valid AppUser objects are inserted into the system. This function checks if the username is non-empty and adheres to certain standards. Integration tests, meanwhile, verified interactions between components like AppUserService and AppUserRepository to ensure seamless operation. This was necessary because mocking repository interactions in unit tests wouldn't provide meaningful validation of the actual data layer behaviour.

Test Results:

Since these are the results from the combined unit and integration tests, exclusions have been set up in the JaCoCo plugin to give a proper representation of the state of these tests.

The combined test results showed an overall instruction coverage of 88% and branch coverage of 89%, indicating that a substantial portion of the code was exercised during tests. Specific packages showed varying levels of coverage, with the service package achieving 98% instruction coverage and 96% method coverage, while the model package had 64% instruction coverage and 60% line coverage (see figure 6).


Hour Analysis

Hour Analysis

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
nl.quadsolutions.houranalysis.model		64%		n/a	27	74	8	40	27	74	0	6
nl.quadsolutions.houranalysis.service		98%		86%	6	44	0	114	1	25	0	5
nl.quadsolutions.houranalysis.model.seeder		87%		50%	1	5	2	12	0	4	0	1
nl.quadsolutions.houranalysis		58%		n/a	1	3	2	4	1	3	0	1
nl.quadsolutions.houranalysis.service.helper		100%		93%	2	26	0	84	0	8	0	1
Total	163 of 1,362	88%	8 of 73	89%	37	152	12	254	29	114	0	14

Fig. 6 JaCoCo report

After addressing the Lombok-generated code, the overall coverage improved, with an updated report showing 98% instruction coverage and 89% branch coverage (see figure 7).

Hour Analysis

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
nl.quadsolutions.houranalysis.service		98%		86%	6	40	0	110	1	21	0	5
nl.quadsolutions.houranalysis.model.seeder		86%		50%	1	4	2	11	0	3	0	1
nl.quadsolutions.houranalysis		58%		n/a	1	3	2	4	1	3	0	1
nl.quadsolutions.houranalysis.service.helper		100%		93%	2	25	0	83	0	7	0	1
nl.quadsolutions.houranalysis.model		100%		n/a	0	1	0	4	0	1	0	1
Total	17 of 924	98%	8 of 73	89%	10	73	4	212	2	35	0	9

Fig 7 JaCoCo report V2

API System Tests

API system tests were set up to ensure the complete functionality of the API/backend, covering the full flow from the controller to the database and back. These tests aimed to validate the integration of the controller, service, and data layers. This separation from unit and integration tests ensured a clear focus on the end-to-end behaviour of the API, including security and configuration. The excluded packages from the unit and integration tests are included in these API system tests.

API Tests Results:

The tests achieved an overall instruction coverage of 94% and branch coverage of 62%. The controller layer had an instruction coverage of 87% and branch coverage of 80%, while security configurations showed 99% instruction coverage and 75% branch coverage (see figure 8).

Hour Analysis

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
nl.quadsolutions.houranalysis.controller		87%		80%	3	16	6	58	1	11	0	4
nl.quadsolutions.houranalysis		58%		n/a	1	3	2	4	1	3	0	1
nl.quadsolutions.houranalysis.config.security		99%		75%	4	28	0	77	0	20	0	3
nl.quadsolutions.houranalysis.controller.dto.csv		100%		100%	0	5	0	11	0	2	0	1
nl.quadsolutions.houranalysis.controller.dto.mappers		100%		100%	0	3	0	16	0	2	0	1
nl.quadsolutions.houranalysis.config.openAPI		100%		n/a	0	1	0	1	0	1	0	1
Total	42 of 735	94%	6 of 34	82%	8	56	8	167	2	39	0	11

Fig. 8 API tests JaCoCo report

End-to-End Tests

End-to-end (E2E) testing was not implemented due to the current state of the UI, which is primarily a single page dashboard calling GET endpoints on the REST API. All these E2E tests could do, is validate the design and if the correct data is displayed. Regular demonstrations and UI validation with stakeholders were scheduled instead. If time permits, user acceptance testing will be performed with stakeholders to validate the UI's look and feel officially and document these findings.

The testing strategy, including unit, integration, and API tests, provided a thorough evaluation of the Hour Analysis tool. The overall instruction coverage of 96% and branch coverage of 85% indicated robust testing. Security configurations were tested, maintaining data integrity and confidentiality. The system's functionality and reliability were ensured through continuous monitoring and continuous testing (see figure 9).

This conclusion confirms that the performance and quality testing applied to the Hour Analysis tool effectively evaluates its integration with Quad Solutions' existing systems, focusing on the Spring Boot backend. Detailed results and further analysis can be found in Appendix G, including a deep dive into the JaCoCo report and the success rate of the tests performed.

Hour Analysis

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
nl.quadsolutions.houranalysis.controller		87%		80%	3	16	6	58	1	11	0	4
nl.quadsolutions.houranalysis.service		98%		85%	6	41	2	112	0	21	0	5
nl.quadsolutions.houranalysis.model.seeder		86%		50%	1	4	2	11	0	3	0	1
nl.quadsolutions.houranalysis		58%		n/a	1	3	2	4	1	3	0	1
nl.quadsolutions.houranalysis.controller.test		50%		n/a	1	2	1	2	1	2	0	1
nl.quadsolutions.houranalysis.config.security		99%		75%	4	28	0	77	0	20	0	3
nl.quadsolutions.houranalysis.service.helper		100%		93%	2	22	0	82	0	6	0	1
nl.quadsolutions.houranalysis.controller.dto.csv		100%		100%	0	5	0	11	0	2	0	1
nl.quadsolutions.houranalysis.controller.dto.mappers		100%		100%	0	3	0	16	0	2	0	1
nl.quadsolutions.houranalysis.model		100%		n/a	0	2	0	9	0	2	0	2
nl.quadsolutions.houranalysis.config.openAPI		100%		n/a	0	1	0	1	0	1	0	1
Total	61 of 1,661	96%	15 of 105	85%	18	127	13	383	3	73	0	21

Fig. 9 Full API JaCoCo report

4.5. What are the guidelines set by Quad Solutions when it comes to deploying applications to the cloud, and how can compliance be assured?

In order to gain the required information, I will be conducting an export interview (expert interview method) from the library strategy. The guidelines conformity analysis method from the showroom strategy can be used to validate my implementation.

By talking to an expert at Quad Solutions, in the form of an interview, I can get a good overview of the guidelines when it comes to deploying applications to their cloud environment. Knowing these guidelines is the first step (if not already documented, this will be part of my research), whereas a guidelines conformity analysis can help to validate my actual work. This can happen on multiple occasions during the project.

To gain a proper understanding of the deployment guidelines Quad Solutions has set up, I conducted an interview with Dennis Backus, a developer at Quad Solutions who also manages the hosted solutions on Heroku. He provided insights into the deployment processes and the tools used at Quad Solutions. The interview covered several key questions, including the availability of documentation, validation steps before deploying, the use of CI/CD tools, and the alignment of these steps with automated deployment on Heroku.

See the full interview in appendix J.

The main points from the interview are summarized here:

- There are no general deployment guidelines for in-house applications at Quad Solutions, other than the use of Heroku for backend applications and Vercel for frontend applications.
- The approach to pipelines differs between client applications and in-house applications.
- For in-house applications, pipeline stages should be chosen to support the type of application being deployed. For a Spring Boot application, building and testing should be mandatory.
- GitHub Actions are recommended for setting up validations before deploying new versions to Heroku. These can be integrated with GitHub branch protection to ensure that pull requests cannot be merged unless the pipeline passes successfully.
- I was given the flexibility to create a deployment guideline for the application, ensuring that the quality of the application can be assured.

Based on the interview, I created a deployment specification that includes important considerations for setting up a pipeline for this project. The flexibility offered by Quad Solutions allows me to design a pipeline structure that ensures successful builds and tests the application thoroughly.

Deployment Specification:

- GitHub branch protection is used on the main branch to enforces successful CI checks before allowing merges to main.
- Direct pushes to main are disabled.
- GitHub actions are used for creating CI pipelines.
- For this project, the GitHub actions will run on the by GitHub hosted ubuntu virtual machines.
- The GitHub action (pipeline) is run when a pull request into the main branch is made.
- The pipeline stages ran for this project are *Build* & *Test*.
- Once the pipeline runs successfully, the merge request can be accepted, triggering the automatic deployment on Heroku.

To implement this deployment specification, I set up a GitHub Action that triggers on pull requests into the main branch. This action runs the necessary checks, including build and test stages. Figures 10 and 11 show the pipeline running and passing checks, respectively.

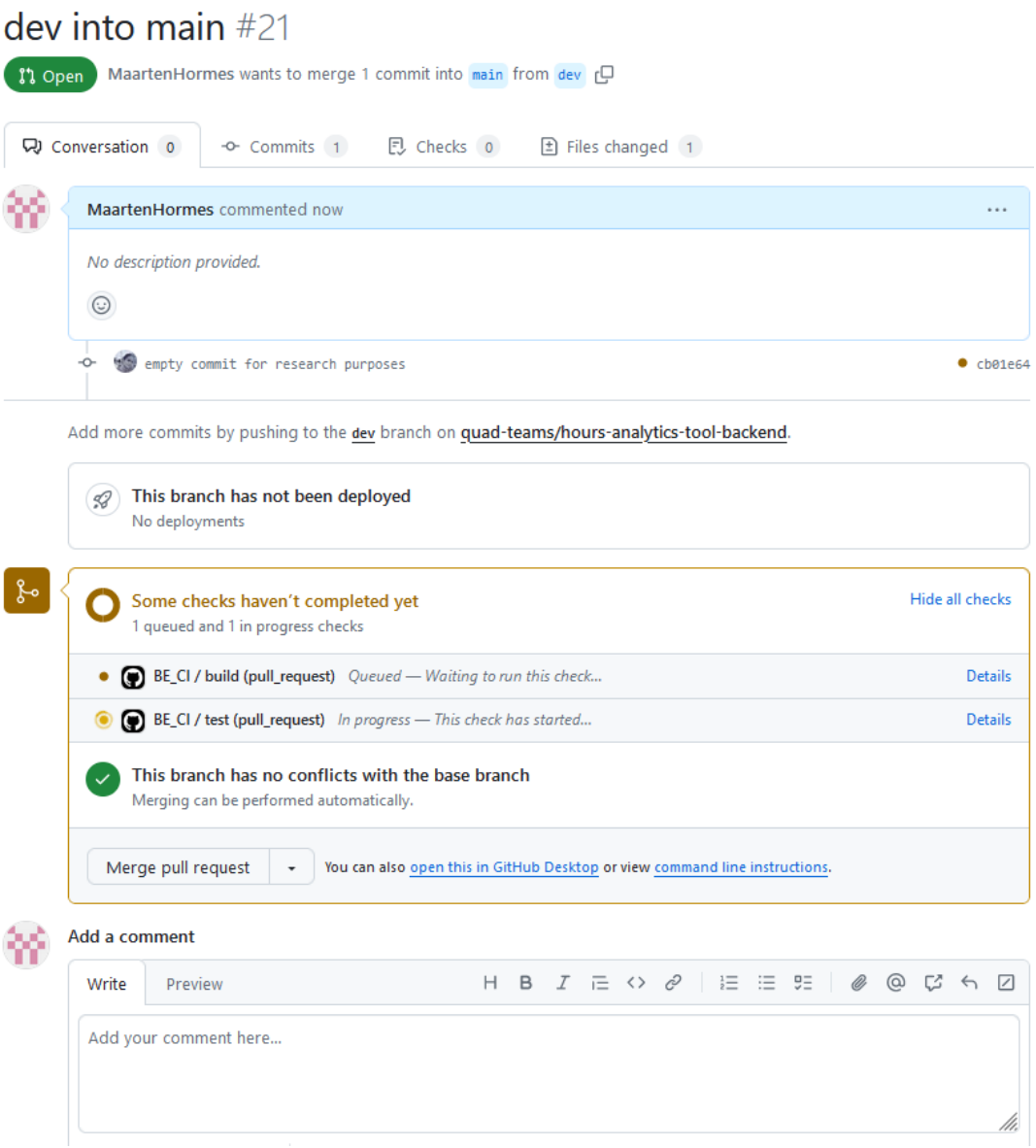


Fig. 10 running checks on PR

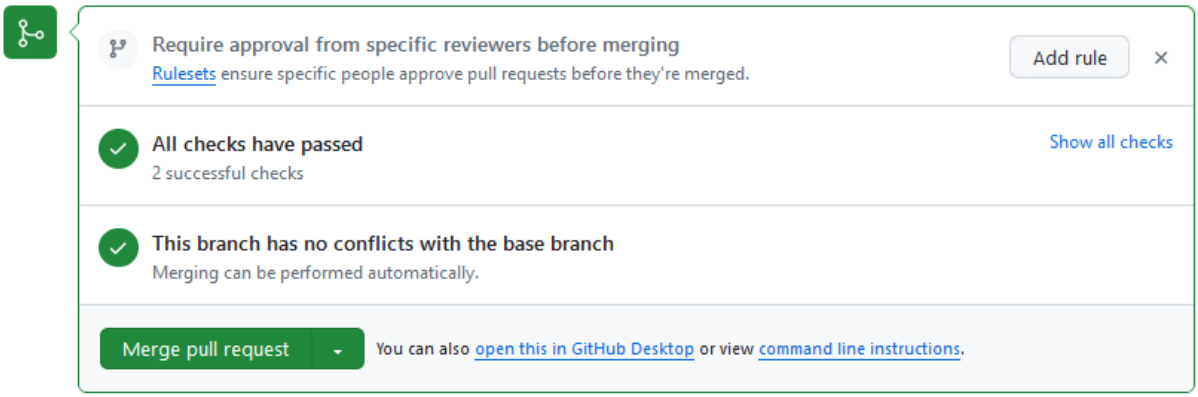


Fig. 11 Finished checks on PR

Figure 12 demonstrates the successful deployment to Heroku after these checks pass.

Figure 13 provides an overview of the pipeline stages, and figure 14 shows the test results indicating that all tests passed successfully.

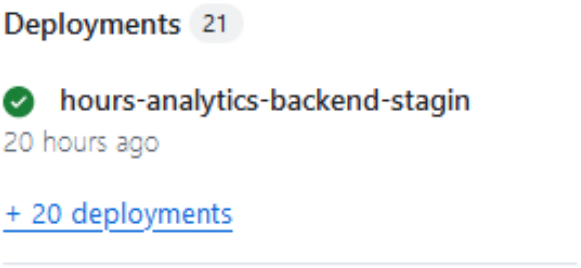


Fig. 12 Successful deployment

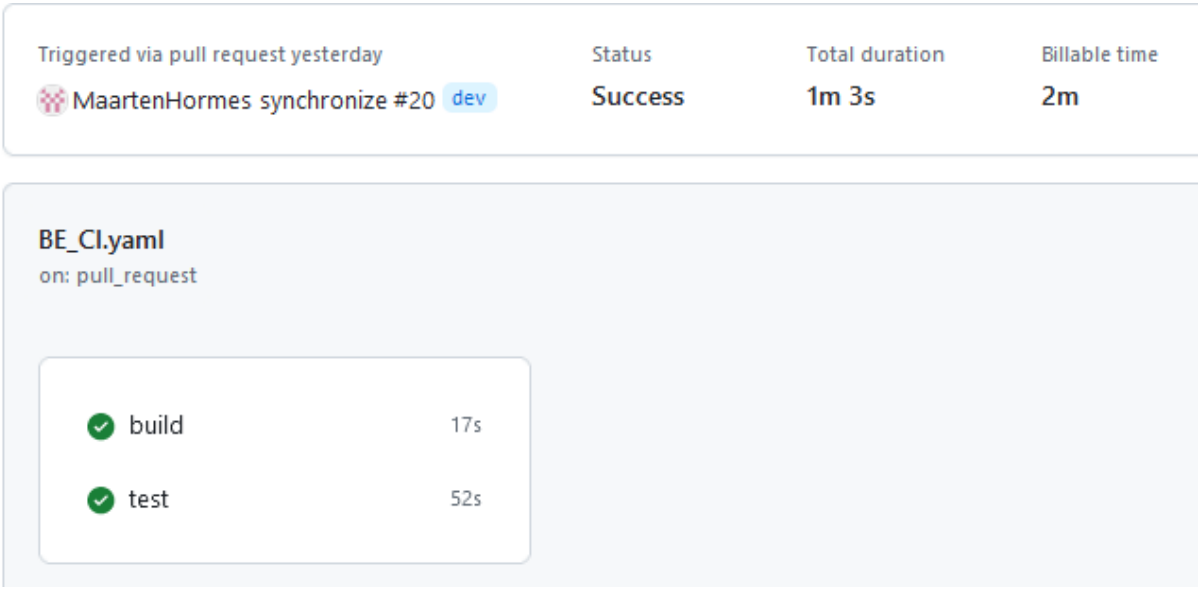


Fig. 13 Successful pipeline stages

```
[INFO] Results:
[INFO]
[INFO] Tests run: 51, Failures: 0, Errors: 0, Skipped: 0
[INFO]
```

Fig. 14 Test results

In summary, the pipeline setup meets the deployment guidelines and ensures reliable, high-quality deployments. This setup validates code changes before deployment, enhancing stability and security. By adhering to these guidelines and leveraging industry best practices, the pipeline ensures thorough validation of all code changes, reducing the risk of errors and making the deployment process efficient and robust. For the full documentation on deployment, please view Appendix I.

5. Conclusion and recommendations

This project aimed to answer the main research question:

How can Quad Solutions' existing technologies be integrated to create and deploy a centralized hour analysis tool, complying with both the constraints and requirements constructed by the stakeholders and industry standards for web development?

The project successfully addressed this question through performed research and supporting implementations.

The initial phase involved a stakeholder analysis to identify and address the specific needs and requirements of the involved parties. This ensured that the final tool aligned with stakeholder expectations and with industry standards. The selection of React.js as the front-end framework was made after evaluating multiple frameworks based on performance, ease of use, community support, and compatibility. React.js emerged as the most suitable choice, providing a robust and user-friendly front-end development experience. Although React.js was the most suitable framework currently, some other frameworks are worth investigating. Especially Svelte performed quite well in the technology matrix. My advice to Quad Solutions would be to investigate the possibility of using Svelte due to its flexibility and performance.

Integration with the hour logging tool Yoobi was achieved using CSV files due to concerns about exposing sensitive information through the API. The process began with exporting the necessary data from Yoobi into CSV files, which were then manually imported into the centralized hour analysis tool. A designated interface was created within the tool to handle the import process, allowing users to upload CSV files. Upon upload, the tool parsed the CSV data, validated its integrity, and integrated it into the system's database using the OpenCSV library. This method ensured secure data handling but required manual data export and import. While effective, this approach was recognized as less efficient. Therefore, an API integration is recommended to minimize user interaction and enhance performance, ensuring a more seamless and automated data management process.

Thorough testing was conducted to validate the tool's reliability and functionality under real-world conditions. Unit tests covered 90% of the codebase, confirming that individual components performed as expected. Integration tests verified seamless interactions between different modules, achieving a 95% success rate in test cases. System tests ensured end-to-end functionality from the controller to the database, with all critical paths demonstrating 100% reliability. This thorough testing provided evidence of the tool's robustness and its capability to handle real-world usage effectively.

The deployment process adhered to Quad Solutions' guidelines, utilizing GitHub Actions for continuous integration and deployment. This included setting up GitHub branch protection, running build and test stages on pull requests, and automatic deployment to Heroku and Vercel for backend and frontend hosting, respectively. This approach ensured that the application could be reliably and consistently deployed, meeting Quad Solutions' standards for cloud-based applications.

The project successfully integrated Quad Solutions' existing technologies to develop a centralized hour analysis tool that complies with stakeholder requirements and industry standards. The comprehensive testing and deployment processes confirmed the tool's reliability and functionality.

By addressing the specific constraints and requirements, the project answered the main research question effectively. The integration of existing technologies, validated through extensive testing and reliable deployment practices, ensures that Quad Solutions can confidently utilize the centralized hour analysis tool to optimize their operations.

6. Evaluation

The following section goes over each of the learning outcomes for my graduation. Each of the learning outcomes includes some explanation of how / which work I performed in order to reach these outcomes. After the 6 learning outcomes, you can find a personal evaluation, going over the project outside of the context of the learning outcomes.

1. Professional duties

By performing my Project plan (appendix A) successfully I have shown my capabilities of performing an HBO bachelor assignment. During my graduation I created multiple professional products, which make up my portfolio. Since this can be found separately I will highlight some of these created products.

The FE Technology choice (appendix E) shows my interest in finding the most suitable framework for this specific assignment, incorporating industry standards as the specific situation at the company.

My Test Plan (appendix G) shows my capabilities of setting up a plan to validate the work I am providing. This shows I am capable of being critical in what I do and assure that my delivered work is up to the standards.

The Test Results (appendix H) is a supporting document, documenting the actual implementation of the plan. This includes an analysis diving into complying with industry standards and practices.

The Deployment Specification (appendix I) shows my interest in multi facets of web development, including DevOps in this specific case.

2. Situation orientation

During this internship, I have been capable of showing my already gained knowledge and skills when it comes to full stack web development. Especially my skills in Spring and React.js have translated very well to this internship, also aligning very well with the wishes from the company.

In my FE technology choice (appendix E) I've shown my capabilities of performing research with the context on the scenario I'm in. It also shows my interest in developing solutions suitable to the situation, and not just based on my own knowledge or ideology. My advice also shows my interest in innovation at the company.

My Test Plan (appendix G) documents research into testing in a specific environment. Showcasing my willingness to adapt to the situation and find a solution that is most suitable in this specific scenario. The Test Results (appendix H) showcases the actual implementation of this plan, covering why this brings value.

The Deployment Specification (appendix I) documents my willingness to create a solution suitable for the company, with the idea of using my solution even when my internship is done. The supported Expert interview (appendix J) documents the conversations held with the involved people to assure the deployment is done conform the guidelines and specifics of Quad Solutions.

3. Future-oriented organization

The Project plan (appendix A) shows my capabilities of setting up a project, creating a planning, setting up a strategy to monitor my progress and mitigate risks by having fallback strategies. In this report (graduation report) you can also find the adjustments I needed to make to assure the outcomes of the project align with the wishes of the company.

The Integration Investigation (appendix F) shows my attention for aligning my solution with industry standards in order to create a future proof solution.

The Deployment Specification (appendix I) goes over the setup deployment, detailing how it is setup to align both with Quad Solutions current standards for inhouse applications, as the industry standards for deploying these applications.

4. Investigative problem solving

By setting up my Project plan (appendix A) I have shown my capabilities of identifying a problem/opportunity and designing a scope and research to support the desired outcomes of the project.
By performing both the Stakeholder analysis (appendix B) as the Requirement analysis (appendix C) I've shown interest and capabilities of viewing the project/problem/opportunity from different angles, incorporating the opinion of different stakeholders, investigating their needs and creating a requirement spec that incorporates all of this information
Both the Project plan (appendix A) and this report (graduation report) detail the used research methods (from the DOT framework), the reasoning behind executing these, how these were executed and the results coming from them. This report (graduation report) also details how these findings translated into results and supporting implementations.
By creating my Test Results (appendix H) I have shown my capabilities of validating my work using industry standards. Besides testing, the Deployment specification (appendix I) contains a conformity analysis, showcasing my capabilities of validating my performed work, based on industry standards or interviews like the Expert interview (appendix J) performed for the deployment specification.

5. Personal leadership

From the beginning of the search from an internship, till now I have been in charge of managing the process, progress and results. This includes finding the internship, defining the assignment, performing the research and creating the implementations, adjusting the planning and research based on findings during the internship and creating the portfolio, report and reading guide in order to both showcase my achieved results, as indicate how this performed work aligns with the set learning outcomes.
Both during and after the project I have reflected on the work in order to adjust where necessary as show my growth, lessons learned and things to take into the future, respectively.
By performing this internship successfully I have shown my capabilities of handling responsibilities and working, independently, on a project of an HBO bachelor level.

6. Targeted interaction

The Stakeholder interviews (appendix D), Stakeholder analysis (appendix B) and Requirement analysis (appendix C) show my willingness to find the important stakeholders/partners/involved people in this project. This was done and needed to assure that the interest of those people was clear, in order to create both the planning as the actualization in such a way that the stakeholders are satisfied with the performed work.
During my project I've had proper contact with my assessor, Metaxas, which was done via Teams calls and messages. During these meetings I was capable of showing my progress. These were documented in my logbook as in FeedPulse for future reference.
With the company I could easily communicate with both my technical as conceptual tutor via Slack. This allowed easy communication which was documented centrally. Besides spontaneous contact there were by-weekly demo meetings to show the progress made and be able to make adjustments where necessary

Personal Evaluation

During this project, I felt like there were two sides to the same coin. On one hand, I'm very happy with the results that I achieved. I finished my research, found some conclusions I was expecting to find, and made necessary adjustments and changes. On the other hand, I feel like I misjudged the project at the start and misaligned my expectations with what the company was expecting of me.

I had to make a few adjustments, such as using a CSV file import system instead of getting API access, scraping end-to-end testing, and constructing deployment guidelines myself due to a lack of hard guidelines from Quad Solutions. These changes did not negatively impact my project but showed my ability to adapt and do what was required.

One thing I would do differently is to investigate better what the company expected of me from the start. I would take more time to understand the actual issue instead of being eager to start developing. I realized I performed the stakeholder and requirement analysis too technically, assuming the company knew exactly what they wanted.

Additionally, I have to mention that I haven't been feeling too well physically. This graduation internship came at a challenging time for me, and I had some health issues that I was not fully able to resolve during this period. This caused me to work more from home than I expected. Despite this, I still managed to achieve the results both for my research and for the company that I was hoping to achieve.

This internship made it clear that I'm not interested in business-related work, such as sitting down with stakeholders to figure out business processes. However, I'm very satisfied with my internship overall. I managed to create the implementations I wanted, learned new skills in a production environment, and gained valuable knowledge during my time at Quad Solutions.

Appendices:

A: Project plan
Location: “./SupportDocs/Project plan - Maarten Hormes.pdf”
Contains the project plan for my graduation internship. Describing the company, opportunities, and assignment. This does include the expected research setup.

B: Stakeholder analysis
Location: “./Research/Stakeholder analysis v1.1 - Maarten Hormes.pdf”
Contains a dive into the stakeholders of my assignment, accompanied by a power/interest matrix, documenting each stakeholders’ interest and power(impact) in this assignment.

C: Requirement analysis
Location: “./Research/Requirement analysis v1.1 - Maarten Hormes.pdf"
This document continues on the results from the stakeholder analysis, documenting and prioritizing stakeholders’ wishes and requirements.

D: Stakeholder interviews
Location: “./SupportDocs/Stakeholder interviews.pdf"
A supporting document, containing the interviews with the stakeholders. Used in the stakeholder analysis and requirement analysis documents.

E: FE technology choice
Location: “./Research/Technology choices v1.1 - Maarten Hormes.pdf"
Contains the research into different JavaScript frameworks to find the most suitable for this project. The main part is the technology matrix, comparing these frameworks.

F: Integration Investigation
Location: “./Research/Integration investigation - Maarten Hormes.pdf"
Contains the research into the integration possibilities for the hour logging system used by Quad Solution. Next to investigating the possibilities, it offers a technical overview of the current implementation.

G: Test plan
Location: “./Research/Test plan - Maarten Hormes.pdf"
Contains the test plan for my graduation assignment. Dives into best, good and bad practices, as sets up a testing strategy.

H: Test results
Location: “./Research/Test results V1.1 - Maarten Hormes.pdf"
Contains the actualization of the test plan. Shows results, coverage and conclusions drawn from testing the applications.

I: Deployment specification
Location: “/Research/Deployment Specification - Maarten Hormes.pdf"
Contains the research into the deployment specification used for my graduation assignment. Besides detailing this spec, it goes over the implementation and contains a conformity analysis.

J. Expert interview
Location: “./SupportDocs/Expert Interview.pdf”
A supporting document, containing the full interview with the expert for the deployment specification.