KMD A/S Ferieplanlægger

cs-24-sw-3-01
Dandan Zhao, Elma Vukicevic, Graziella Zafir Rosager Lund, Mads Ludvig Timm Fagerlund, Marc Nygaard, Sille Dalberg Korsbakke and Thorbjørn Dige Larsen

AALBORG UNIVERSITY
STUDENT REPORT

Computer Science
Aalborg University
http://cs.aau.dk

Title:
KMD A/S Ferieplanlægger

Theme:
Insert theme name

Project Period:
Fall Semester 2024

Project Group:
cs-24-sw-3-01

Participants:
Dandan Zhao
Elma Vukicevic
Graziella Zafir Rosager Lund
Mads Ludvig Timm Fagerlund
Marc Nygaard
Sille Dalberg Korsbakke
Thorbjørn Dige Larsen

Supervisor:
Asger Horn Brorholt

Date:
2024-10-07

Copies: 1

Number of Pages: 14

Abstract:

This is the abstract, please write something here. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit.

Contents

1 Styleguide

1.1 Language

We will use american english for this report

1.2 Tops and Tails

Tops will only be used for chapters.

We will use tops and tails for the sections of the document. A top will only be used where it makes sense. Tails are required. Tops will only be used in chapters.

1.3 Workflow

1. Create issue on kanban board, and assign yourself to it
2. Pull from main
3. Create new branch based on main
4. Link branch to issue on kanban board
5. Push and Commit to branch while you work
6. Create Pull Request to main, where you explain the addition
7. Add reviewers to pull request (Try to mix up the reviewers of your PR's)
8. Squash merge into main, when PR is approved

2 Components

Title for Text Box with Footer

The main content of the text box goes here!

Footer text goes here

Title for Text Box

The main content of the text box goes here!

Code Block Title

```rust
fn main() {
    println!("Hello, world!");
}
```

Rust

```rust
fn main() {
    println!("Hello, world!");
}
```

Rust

Note Title

Inside a Carnot cycle, the efficiency η is defined to be:

$$\eta = \frac{W}{Q_H} = \frac{Q_H + Q_C}{Q_H} = 1 - \frac{T_C}{T_H}$$

3 Introduction

The project is being carried out in collaboration with KMD, a company located in Aalborg that develops IT solutions to meet the evolving digital needs of modern societies.

The process of finding a collaborative company began by searching through the company guidebook "Peiling", which was delivered by attending a student job fair. After finding a few interesting companies, emails were sent to arrange meetings. During the initial meeting with KMD, they expressed significant interest in the collaboration and wants to provide as many resources as possible, such as providing interview participants and access to a hosting server. Their enthusiasm gave motivation and a desire to collaborate with them.

The aim of this project is to develop a functional application that prioritizes user experience and meets the client's needs. The primary requirement for this system is to develop a holiday planner to track non-working days for the company's employees. KMD already has a holiday planner, but since it's outdated, they want to replace it with an imporved version.

3.1 Current System at KMD

Aside from using Microsoft Outlook Calendar and the calendar integrated within their planning boards, KMD currently have an outdated vacation planning solution called KMDFerieplan. This system, developed approximately 10 years ago, operates on a Microsoft backend and is integrated with SAP SuccessFactors for managing vacation codes. Previously, it could automatically synchronize vacation requests with the calendar, but this functionality is no longer available following KMD's migration to SAP S/4HANA, as the system is incompatible with the updated enterprise resource planning (ERP) platform.

the current product, KMDFerieplan, is accessed through a web application, where employees log vacation requests by specifying start and end dates and times. Scrum masters or leaders are responsible for creating teams in the application, with each employee assigned a distinct color for easy identification in the calendar. Teams are organized through a drop-down menu, where specific employees can be added. However, only the team creator or owner can make modifications. This can be problematic if an employee leaves the company, because the corresponding team would become static and unable to be modified, since the original owner would no longer be registered in the system.

4 Problem Analysis

4.1 Feature Requests and Potential Problems

This chapther defines both the potential problems that KMD have with their current solution, and the features required in the new one. The different finds in this chapter was aquired through interviews with three company employees, their lead developer, Mats Lindberg, their project portfolio manager, Caroline Ramsdal and their manager of modern workplace, Dion Andersen, respectivley.

add a table of feature requests specifically requested by people that have been interviewed. (niche requests - to further on get a in depth analysis and thereby a greater defined design, with greater begrundelse) - Si

Transcriptions of said interviews can be found in appendix XYZ

insert relevant apendices

To find out why KMD needs a new system for managing their holiday calender i was important to find the problems with the current solution. The interviews revealed problems with synchronizing with their current SAP system. system but surprisingly most of the problems with the current system is a lack of features due to the system being developed 10 years ago. The features that are missing can be seen in Table 1.
The absence of these features and the synchronizing problems with SAP have led to most of the employees using other programs to plan their holidays with most of the active users being employees that have worked at the company for a longer time.

Missing feature	Desciption
Holiday balance	The program has to display the individual employees current available absence days. Prefereably there should be a function to loan absence days from the future which then are regained throughout the year.
Lookup specific team	The current system is missing a way to quickly lookup a specific team in any department, there should be a function to search for the specific team and its members.
Create teams across departments	Too fully utilize the teams the company should be able to create teams with members across different departments.
Viewable Holidays	When looking up other teams, the user should be able to see what days the members of said team are absent.
Toggleable UI	The user should be able to switch between different calendar types, e.g. weekly, monthly, yearly.
synchronizing with SAP	The new program should be able to synchronize with SAP and vice versa. The program has to use their preset absence codes.

Table 1: Features that are missing in the current holiday planner at KMD.

4.2 PACT

4.2.1 People

The users comprise all employees of KMD. This includes employees of various roles, such as software developers, financial analysts, human relations, and administrative staff.

These users need to be able to intuitively record their vacation days, in a simple system. Furthermore they need to be able to compose teams of employees from different departments, for cross-department projects. At last, they need to be able to manually register vacation, since they sometimes do not have accrued enough vacation days in their SAP SucessFactors system.

These users also have varying levels of technical proficiency. Therefore the systems interface must be intuitive for both technical and non-technical staff.

4.2.2 Activities

The primary activities of the system, will that of recording vacations days accompanied by the vacation reason and being able to compose teams consisting of employees from different departments.

Users will firstly log in with their company supplied Microsoft account, whereafter can

either register vaction, select a team to see vacation days for, or record new vacation days.

The system will be engaged with regularly, especically during the peak vacation periods. Project managers usually check it multiple times a week, to know they status of their team.

4.2.3 Context

The vacation planner will be accessible from all locations with internet connectivity, including at home, the office, or on the go with smartphones. This should happen trough a secure VPN with gives access to te local server.

Furthermore the product will facilitate collaboration among employees of different departments, allowing them to get a better overview of their teams.

The useres will consist of both Danish and Polish users, which arises some cultural differences, including the specific holidays of Denmark and Poland, as well as varying approaches to vacation planning, communication styles and workplace expectations.

4.2.4 Technologies

The users are currently using a mix of the old vacation planner (the one being rewritten in this project), Outlook Calender, and SAP SuccessFactors. They will continue to use Outlook Calender for meetings, and SAP SuccessFactors for primary vacation planning. The vacation planner system will merely be a way of viewing the vacation registered in SAP SuccessFactors, with the ability to override by manually registering vacations, when not possible in SAP SuccessFactors. This system is built with C# and a Microsoft SQL database.

The new system will be built using Java with Spring Boot for the API, and React for the frontend, ensuring the ability to create a modern and responsive webapplication.

For persistence, PostgreSQL will be used to efficiently and reliably store user data.

It needs to integrate with SAP SuccessFactors to be able to show the vacation registered there. Furthermore it must also integrate with Microsoft Entra Id, to be able to authenticate the users.

4.3 FACTOR

The FACTOR model has been applied to the absence calendar system to ensure a comprehensive analysis of the key components and requirements needed for successful implementation. The FACTOR model provides a structured approach to identifying the system's functionality, the domain in which it operates, the conditions of development, the technologies involved, the objects that are central to the system, and the overall responsibilities. By using this model, we aim to clearly define how our system will function within the organizational context of KMD.

4.3.1 F - (Functionality)

The system will be used by employees to register absence and view absence for themselves and their teams. In addition to this basic functionality, employees must be able to specify the type of absence, such as vacation, sick leave, or other reasons, as well as indicate the duration of the absence. The system will also allow employees to create, modify, and delete teams, enabling efficient organization of team members according to department or project requirements. Team members can view each other's absence schedules to help avoid overlapping leave periods and ensure optimal staffing levels.

4.3.2 A - (Application domain)

The application falls under the domain of human resource management, specifically focusing on absence tracking and planning within an organization. It is designed to facilitate self-service absence management for employees while also providing oversight capabilities for managers. It supports both individual users who wish to manage their absences and managers who need to monitor the availability of their teams.

4.3.3 C - (Conditions)

The system is being developed by a team of bachelor software students in cooperation with KMD employees, who provide the necessary knowledge and information regarding company-specific requirements.

Additionally, the system must be versatile enough to accommodate users with varying levels of IT experience. It will be accessible from a variety of devices, including desktops, tablets, and mobile phones, allowing users to engage with the system in the most convenient manner possible.

4.3.4 T - (Technology)

The system will be developed as a web application, with a client-side interface that runs on each user's device and communicates with a server over a wireless network. The server will host the system's logic and maintain the database. The backend application will utilize a RESTful API, implemented in Java using the popular framework "spring boot", while the frontend will be developed using frameworks like React to ensure a responsive and user-friendly interface. Data concerning employees, absences, and teams will be managed through a relational database named PostgreSQL.

Maybe add that the network that the application runs on is a protected one. - TB

4.3.5 O - (Objects)

The main objects within the system are people, dates, types of absence, and teams.

Der skal tilføjes mere, men er ikke sikker på hvad jeg skal skrive... - Marc

4.3.6 R - (Responsibility)

The system is intended to ensure transparency in absence management across the organization. It provides a clear view of team members' absence schedules, helping to prevent scheduling conflicts and enabling better planning for upcoming work periods.

Der mangler en tail, men synes ikke at det gav mening at skrive på nuværende tidspunkt -

4.4 System Definitions

4.5 Requirements

5 Problem Domain Analysis

5.1 Classes of the System

5.2 Event

5.3 Behaviour Stucture

6 Application Domain Analysis

6.1 Use Cases

6.2 Functions

6.3 Interface

7 Design

7.1 Graphical User Interface

7.2 Architectural design

7.3 System Processes

7.4 Component design

7.5 Final Model Prototype

8 Implementation

9 Testing

9.1 Unit and Integration

9.2 User Tests

10 Reflection

11 Bibliography