UNIVERSITIES OF ZAGREB, BELGRADE, ŽILINA

FACULTY OF ORGANISATION AND INFORMATICS,

FACULTY OF ORGANIZATIONAL SCIENCES,

FACULTY OF MANAGEMENT SCIENCE AND INFORMATICS

Schedule management application with PostgreSQL and JavaScript GUI

jcc project

Mentors: Students:

prof. Srđa Bjeladinović, PhD. Isidora Lazić 2022/3013

Prof. Dr. Sc. Markus Schatten Radovan Filický 5ZIS21

doc. Ing. Michal Kvet, PhD. Bruno Bašič

Domagoj Glumac

Tomislav Blaževič

Zasgreb, Beograd, Žilina, 2023.

Table of Contents

[Introduction 1](#_Toc125701739)

[1. ER diagram 2](#_Toc125701740)

[2. Application 4](#_Toc125701741)

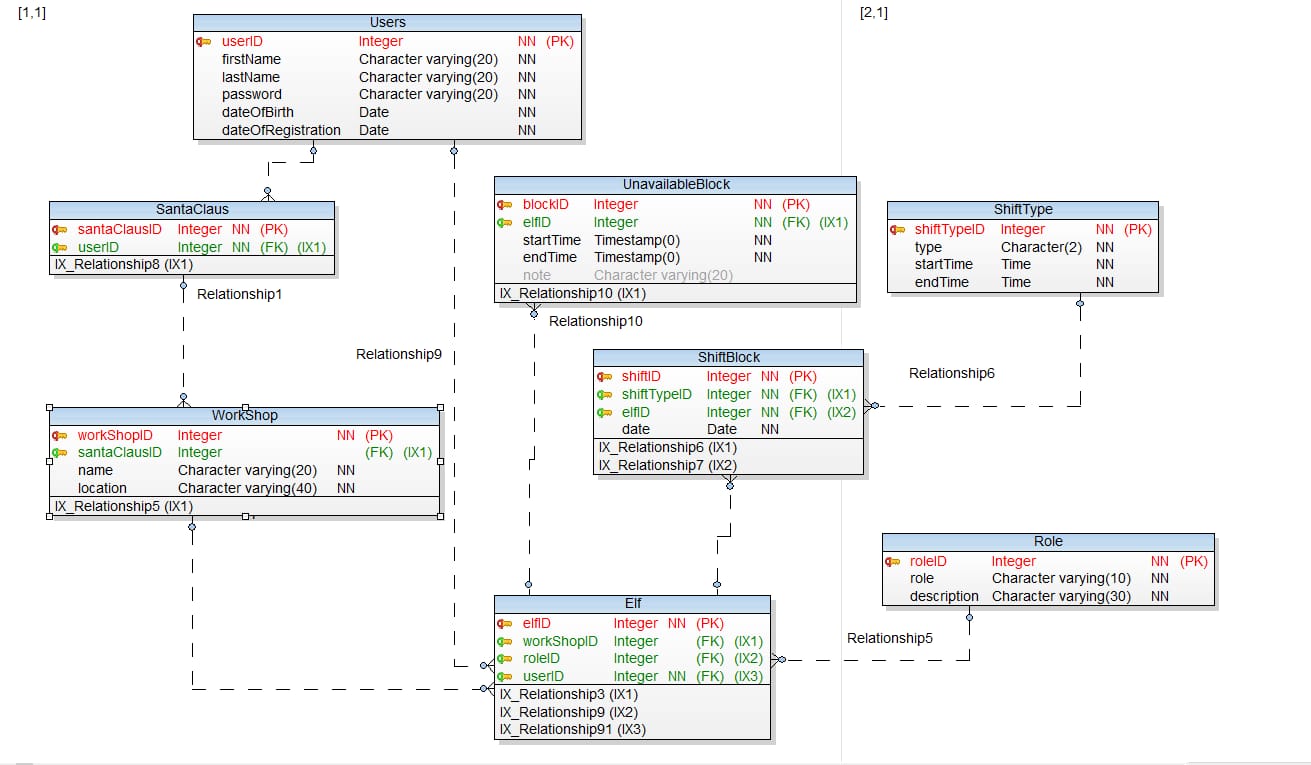
[3. Conclusion 18](#_Toc125701742)

[4. Appendix 19](#_Toc125701743)

# Introduction

This project documentation describes the work of our team on the topic of temporal databases called " Schedule management application with PostgreSQL and JavaScript GUI".

1. ER model (Database Schema)



Our schema consists of eight objects/tables. Users, which can be specialized into SantaClaus and Elf objects. They have the same attributes as Users, with the addition of attributes unique to each of them.

Elf object has additional ElfID, reference to the workshop he/she works in via WorkshopID, as well as RoleID, foreign key from Role table. SantaClaus manages one Workshop and has its own SantaClausID. Workshop has a name, a foreign key reference to the SantaClaus that manages it, name and location attributes.

A shift block is represented via ShiftBlock table, that has its ID, FK to the ShiftType table and Elf table (which elf works a given shift), as well as a date. Each elf has shift blocks when they're away/unavailable, and that is represented with the UnavailableBlock, which references the Elf table via FK.

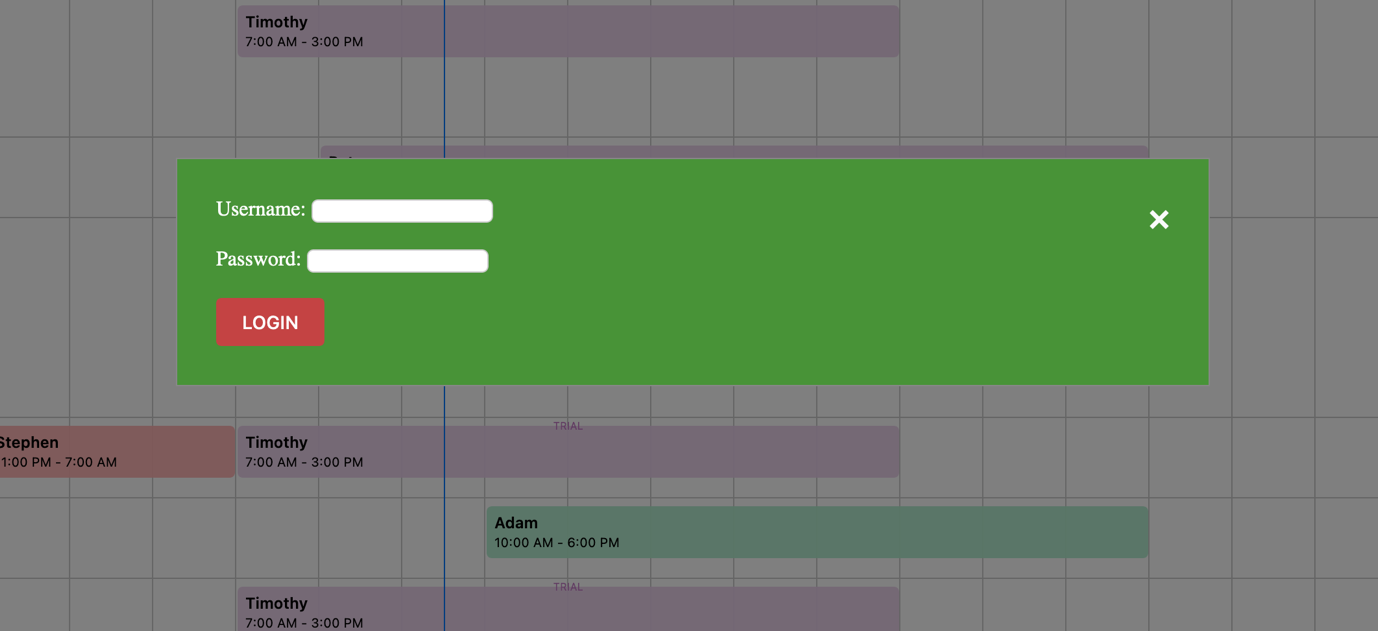
Role table includes its id, role name and description, while ShiftType has id, type name, shift start and end date.

# 2. Backend

For backend we have choosen .net API aplication which has defined endpoints for all needed operations to manage schedule for Santa's workshop. Aplication sends http requests to API which is connected to local database.

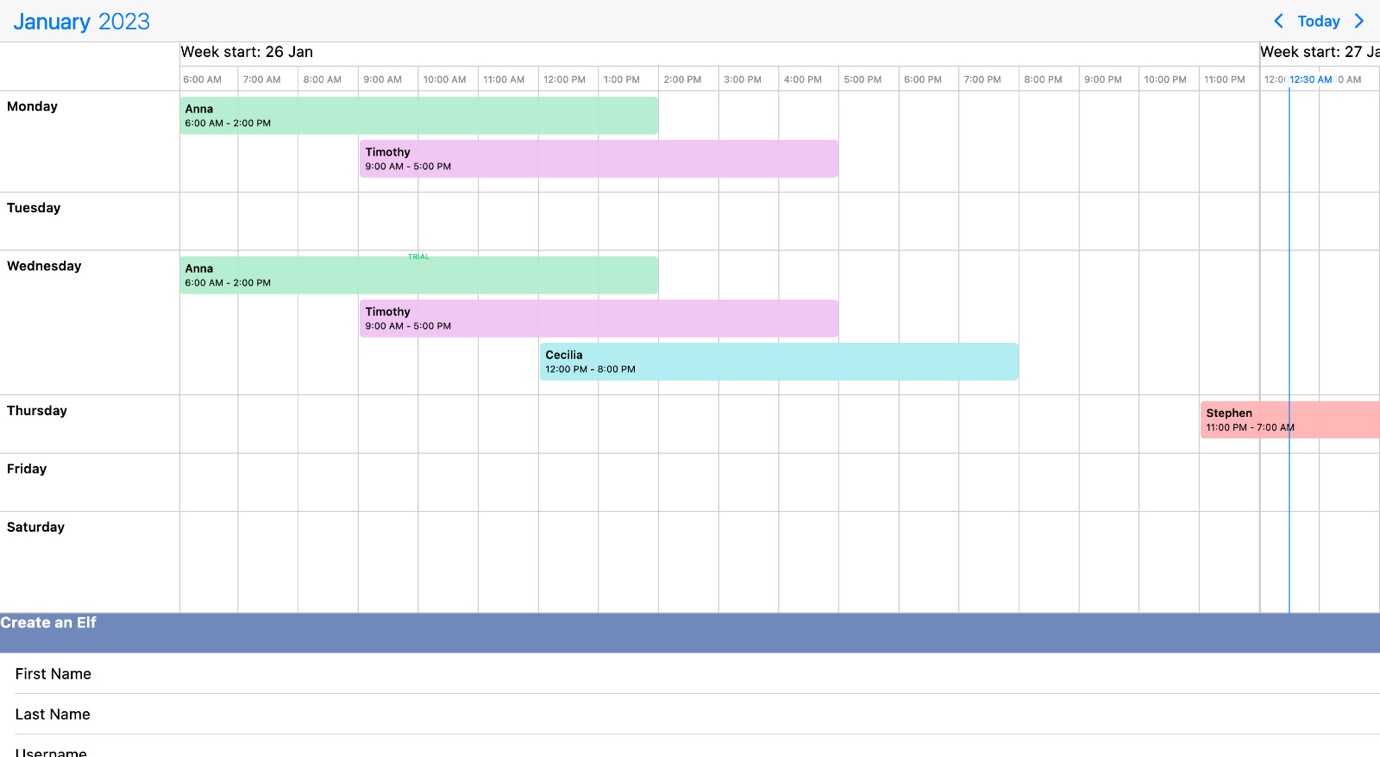
# 

# 3. Application



Picture 1 Login form

Login form has username and password fields which are of text input type. Upon entering credentials and clicking the Login button, those values are looked for in the Users table in the database, If a match is found, login form disappears and the schedule view is presented.



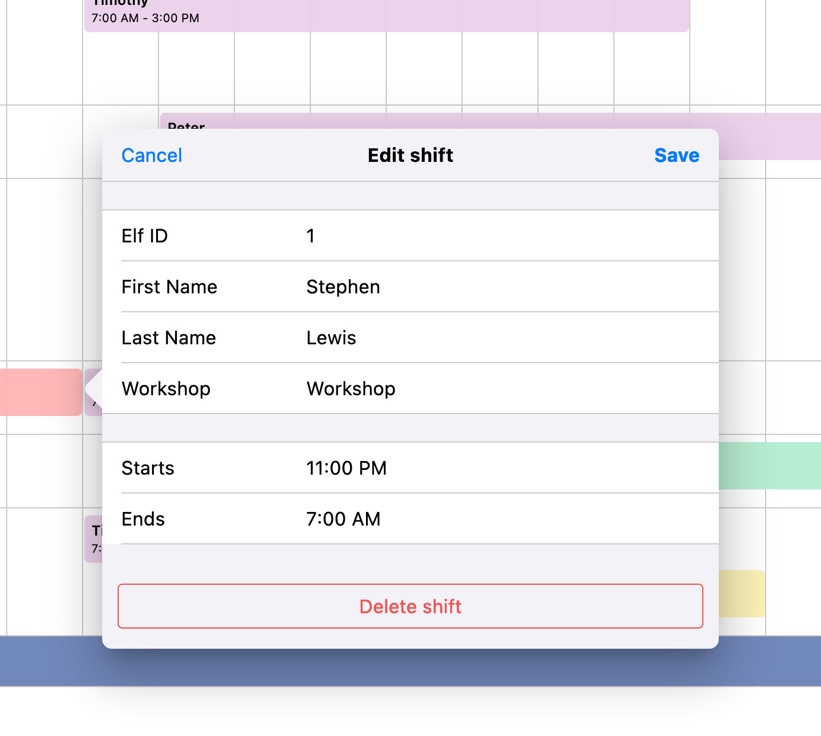
Picture 2 Schedule view

Schedule is designed to be user-friendly and interactive. On the left side there's a list of days of the week.

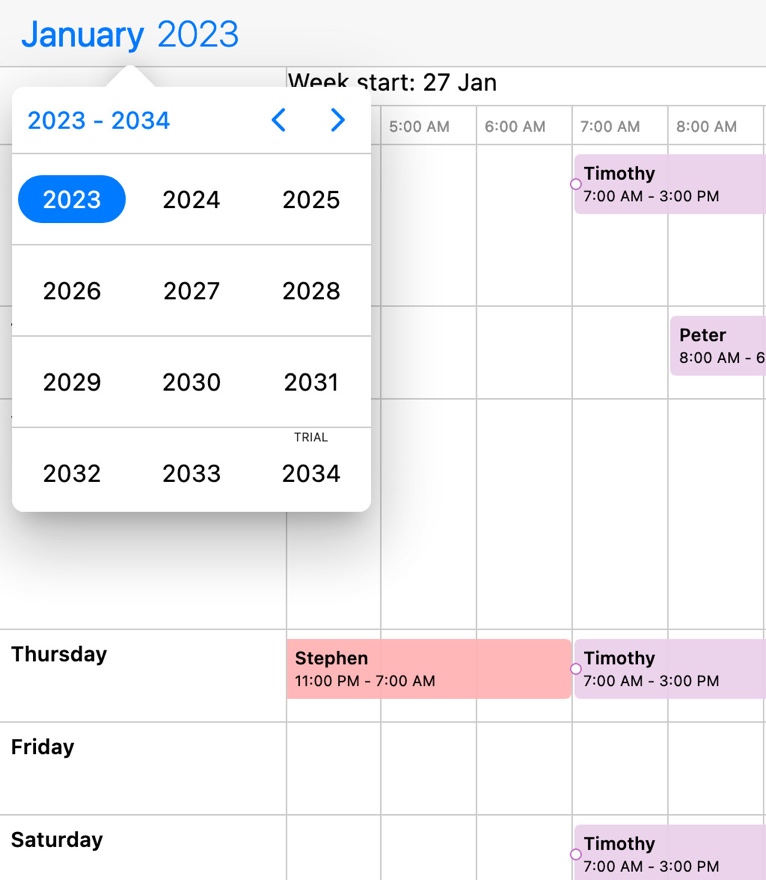
Upon clicking a schedule cell user will be promted to add a new shift. Each shift block is adjustable in length and postition across the schedule and colored differently for each day of the week. Clicking an existing shift block opens the edit shift form. There can be multiple elves working the same shift and the cell will expand to accomodate them.

At the top we can see blocks of time by hour, from 0:00AM until 23:00PM. In the top left corner, when we click on month and year label, a date picker will show up, which allows us to change the schedule view to the selected time interval.

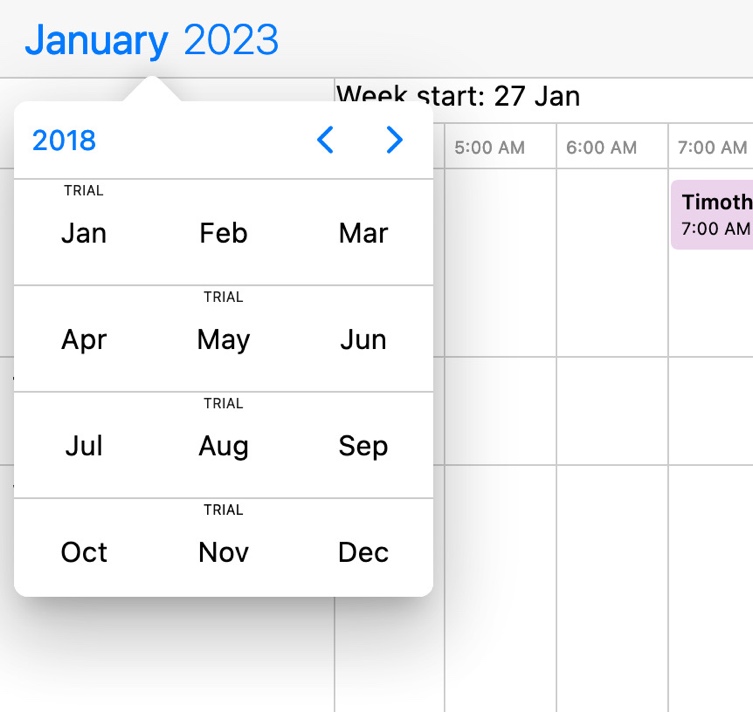
In the top right corner, there is a Today button which moves the view to the current date, and arrows which shift the schedule view one week before/after.



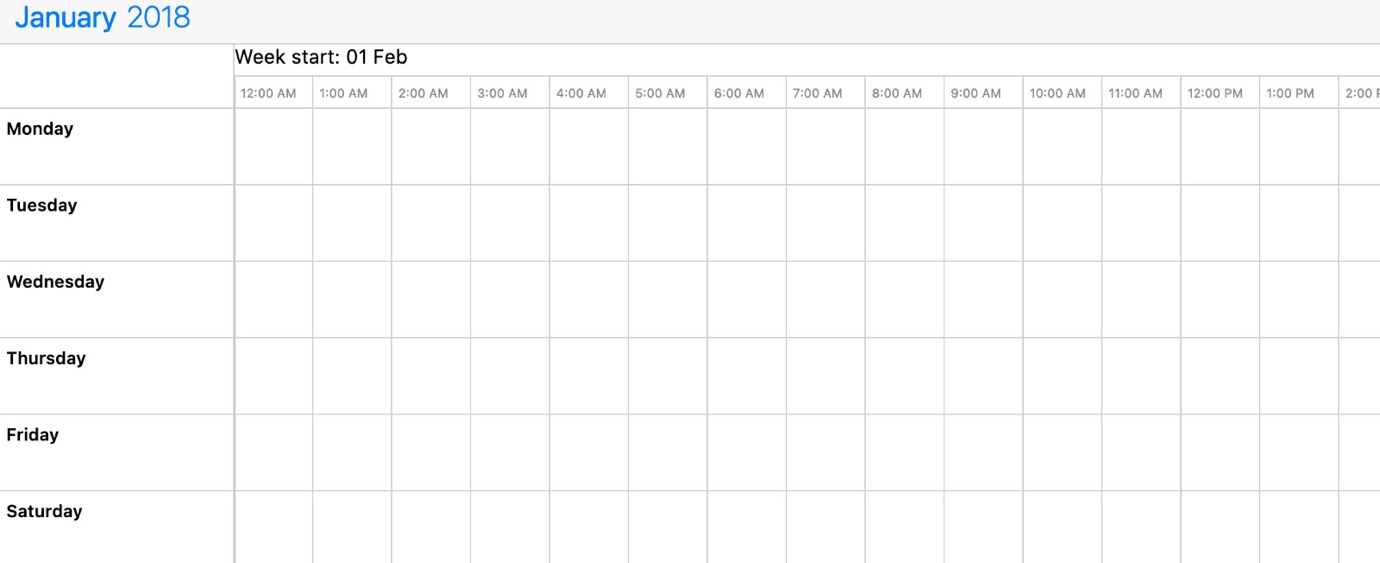
Picture 3 - Edit shift form



Picture 4 - Year picker



Picture 5 - Month picker



Picture 6 - Schedule changed to selected date

Below the schedule there's a form for adding a new elf. Firstname, lastnamem usrname, workshop and password are text fields, clicking on the date of birth field brings up a date picker.



Picture 7 - Create elf form

# 3. Conclusion

While working on this project, we got a unique opportunity to get a closer look into PostgreSQL and temporal databases. We also worked on a challenging but interesting UI design with advanced JavaScript and CSS, which further improved our skills both in SQL and Web development. We also got a better look at how databases are created from modeling to implementation, which greatly enriched our knowledge and skillset.

Last but not least, we got a chance to work with remote international teams which will surely be a very helpful add-on to our work experience.