WEB APPLICATION FOR THE MANAGEMENT OF REFEREE DELEGATIONS TO FOOTBALL MATCHES IN PRAGUE

Filip Pavlík

Bachelor's thesis
Faculty of Information Technology
Czech Technical University in Prague
Katedra softwarového inženýrství
Study program: Informatika
Specialisation: Softwarové inženýrství
Supervisor: Ing. Miroslav Balík, Ph.D.
June 10, 2025

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Web application for the management of referee delegations to football matches in Prague
Student:	Filip Pavlík
Supervisor:	Ing. Miroslav Balík, Ph.D.
Study program:	Informatics
Branch / specialization:	Software Engineering 2021
Department:	Department of Software Engineering
Validity:	until the end of summer semester 2025/2026

Instructions

The aim of this bachelor thesis is to analyze the processes involved in delegating referees to football matches within the city of Prague. These processes are carried out weekly during the competition period by individuals responsible for delegating, on the basis of information supplied by active football referees under the administration of PFS. The thesis will be conducted in collaboration with the delegation department of the Prague Football Association (PFS).
Based on the analysis, design and implement a web application, which simplifies selected processes.

Follow these steps:
1. Analyze the processes involved in creating referee delegations. Focus on processes that can be simplified or automated through the application. Define the key features the application should support.
2. Propose a solution and implement it using the .NET Core framework. Justify the choice of additional technologies.
3. Develop and implement a set of tests to verify the required functionality of the application.
4. Create documentation and user guide.
5. Evaluate the application's usability by gathering and incorporating feedback from users.

Electronically approved by Ing. Michal Valenta, Ph.D. on 5 February 2025 in Prague.

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Pavlík Filip. *Web application for the management of referee delegations to football matches in Prague.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2026.

First of all, I would like to thank my bachelor's thesis supervisors, Ing. Miroslav Balík, Ph.D., and Assist. dr. Sandi Gec, who pointed me in the right direction and provided a lot of valuable advice. Furthermore, I would like to thank the Prague Football Association for their great assistance in the early stages of this project, particularly for providing me with the idea and for sharing information related to the delegation process. I cannot forget my family, who believed in me throughout my studies and supported me.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

I declare that I have used AI tools during the preparation and writing of my thesis. I have verified the generated content. I confirm that I am aware that I am fully responsible for the content of the thesis.

In Praze on June 10, 2025

Abstract

Fill in the abstract of this thesis in English. Lorem ipsum dolor sit amet. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Cras pede libero, dapibus nec, pretium sit amet, tempor quis. Sed vel lectus. Donec odio tempus molestie, porttitor ut, iaculis quis, sem. Suspendisse sagittis ultrices augue.

Keywords enter, comma, separated, list, of, keywords, in, ENGLISH

Abstrakt

Fill in the abstract of this thesis in Czech. Lorem ipsum dolor sit amet. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Cras pede libero, dapibus nec, pretium sit amet, tempor quis. Sed vel lectus. Donec odio tempus molestie, porttitor ut, iaculis quis, sem. Suspendisse sagittis ultrices augue.

Klíčová slova enter, comma, separated, list, of, keywords, in, CZECH

Contents

List of Figures

List of Tables

List of Code listings

List of abbreviations

PFS	Prague Football Association

Chapter 1
Introduction

Football is the greatest sporting phenomenon on the planet. With billions of fans worldwide, it is a sport that captivates people from all walks of life and unites communities across continents. However, it is not only about clubs and players, but also about the equally important referees. Referees have always faced high demands, and making the right call, like awarding a decisive penalty in the 90th minute or ensuring fairness throughout the match, is an exceptionally difficult task.

As someone who has been a referee for several years, I know that the best learning happens through practice, by refereeing hundreds of matches. If you have ever wondered how the right referee is chosen for a particular match, who makes these decisions, and how the process can be improved, this thesis aims to provide the answers.

The primary objective is to develop a web application, in collaboration with the Prague Football Association, that facilitates a more efficient and higher-quality referee delegation process. This solution aims not only to streamline administration but also to ensure fairer and more transparent match appointments. At the end of this thesis, I will present my vision for the future development of this application and its potential contribution to the broader football community.

Chapter 2
Goals

The main goal is to develop a fully functional web application hosted on the domain www.rozhodcipraha.cz. The long-term objective is for the application to be actively used, regularly enhanced with new features, and eventually expanded to support other football associations.

To achieve this, several smaller objectives must be met. The most important is a thorough and detailed analysis of the existing processes, with a focus on identifying those that can be streamlined. The second objective is to design the web application to meet the specified requirements using .NET technologies, with an emphasis on maintainability, sustainability, and extensibility. The use of any additional technologies should be well justified.

A prototype of the application that meets all requirements will then be created. This prototype must be properly documented and tested. Finally, the outcomes of this thesis should be evaluated, and directions for future development should be outlined.

Chapter 3 Prague Football Association

This thesis was conducted in collaboration with the Prague Football Association (PFS), more specifically with Commission of Referees, which played a key role in shaping the project and provided information on the weekly match delegation process. The following chapter describes the PFS as an organization, detailing its competencies, organizational subcomponents and detailed description of current form of delegation process.

3.1 Organization and activities

PFS manages football competitions in the capital of Czech Republic in all age categories,is an organizational component of the Football Association of the Czech Republic with its own legal subjectivity and a special statute. Unlike other regions, it is the only component that manages regional and district competitions together.[1] In other regions, we always find a regional and district management component. PFS is also the largest regional football association in the Czech Republic.The association is divided into many various components, whose are responsible for its service. As an example, can be mentioned Executive Committee, already mentioned Commission of eRferees, Disciplinary commission, Sports Technical Commission and Economic Commission. [1]

The initiative came from Mr. Sýkora, a member of the Commission of Referees responsible for referee education and primarily for match delegations. Match delegations are officially controlled by two members of the commission and assisted by four others, mostly younger individuals still operating as referees under the PFS.

The main problem, according to Mr. Sýkora , is the tedious and complicated process of delegating every week. The various pieces of information needed for delegation are scattered across different shared documents, so the

[1] As a result, they delegate more matches than other district associations.

inconvenience stems from the lack of a compact application that brings all the information together.

3.2 Current form of delegation process

Referees submit excuses or time unavailability for game days by sending an email to the designated Commission of Referees's email address. The excuse must be submitted at least 14 days prior to the requested day off. Otherwise, it is considered late and must also be reported by phone to the person in charge. For such late excuses, the Commission of Referees reserves the right to impose a monetary penalty on the referees concerned.

The delegation process primarily uses an Excel file called ”obsadenie.xlsx”, which is divided into six sheets. Referees are categorized into these sheets based on their current level — that is, the league in which they primarily officiate. Within each sheet, referees are organized in rows (i.e., one referee per row), with their information listed in this order:

- **Notes about referee-** may include club vetoes, time preferences (e.g., a maximum of 2 matches per day), and any other special remarks.
- **The current residence of the referee**
- **Pfs or age-** note whether the referee is active under pfs or the age of the referee for level M (youngsters)
- **Name and surname**
- **Time slots for match assign-** pre-prepared time slots of game dates divided into parts of the day (e.g. first column Saturday 7.6.2025 in the morning, next column Saturday 7.6.2025 in the afternoon)

In addition to this file, there is a list of matches file called ”seznam-zapasu.xslx” for the game weekend downloaded from the FAČR system in Excel format to know which matches have to be completed. There is also an auxiliary file for recording previous delegations called ”odohrane.xlsx”, where information from played matches is manually entered to track how many times a referee was assigned to matches of a given team. This information is recorded only for the top two men's competitions under the scope of the PFS, in the format of referee and assistant referee.

3.2.1 BPMN diagram of the process

The diagram 3.1 illustrates the time sequence of subprocesses using BPMN notation. The diagram shows the amount of manual rewriting and repetition. By automating some processes, we could speed up whole process.

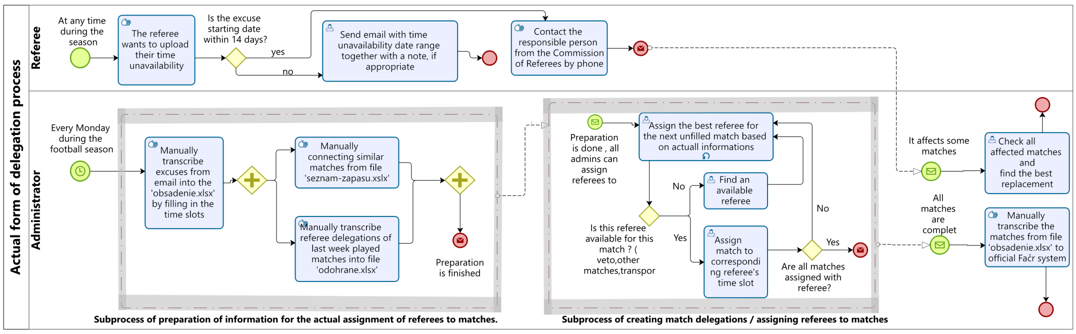

Actual form of delegation process
Referee
Administrator
At any time during the season
The referee wants to upload their time unavailability
Is the excuse starting date within 14 days?
yes
no
Send email with time unavailability date range together with a note, if appropriate
Contact the responsible person from the Commission of Referees by phone
Every Monday during the football season
Manually transcribe excuses from email into the 'obsadenie.xlsx' by filling in the time slots
Manually connecting similar matches from file 'seznam-zapasu.xslx'
Manually transcribe referee delegations of last week played matches into file 'odohrane.xlsx'
Preparation is finished
Subprocess of preparation of information for the actual assignment of referees to matches.
Preparation is done , all admins can assign referees to
Assign the best referee for the next unfilled match based on actuall informations
No
Find an available referee
Is this referee available for this match ? (veto,other matches,transpor
Yes
Assign match to correspondi ng referee's time slot
No
Yes
Are all matches assigned with referee?
Subprocess of creating match delegations / assigning referees to matches
It affects some matches
Check all affected matches and find the best replacement
All matches are complet
Manually transcribe the matches from file 'obsadenie.xlsx' to official Fačr system

Figure 3.1 BPMN diagram showing the delegation process and its subprocesses , created in Bizagi Modeler. [2]

3.2.2 Problematic parts

Problematic areas that could potentially be optimized in terms of time or used to improve the quality of referee delegations:

- **1. Manual transcription** of excuses from the email inbox into the file ”obsadenie.xlsx”.
- **2. Manual recording** of already played matches into the file ”odohrane.xlsx” , small range of controlled competitions.
- **3. The manual linking** of consecutive matches, which sometimes also requires checking the game times of individual matches, as not all are played for 90 minutes.
- **4. Manual transcription** of already delegated matches from application into the official fačr system.
- **5. When assigning matches** to referees, we don’t specify the exact match time—only the part of the day (morning or afternoon). This creates a problem, as we need to check for possible time overlaps when assigning additional matches.
- **6. Transport between matches** is related to fifth point. When assigning referees to matches, we also need to consider the transfer time between matches. Currently, this is done manually, roughly based on the locations of the game fields.
- **7. For each referee we delegate** , it is necessary to manually check whether they have a veto against any of the teams playing in the match.
- **8. The current system is unable to effectively control** the number of matches assigned to each referee, which results in significant disparities in match assignments over the weekend.

Chapter 4

Analysis

Requirements gathering and analysis are considered key steps in software engineering for successful application development. This chapter discusses the compilation of functional and non-functional requirements based on sessions with the client. It is important to communicate and understand the intent of the system—preferably with the people for whom the software is being developed. [**Wiegers2013Software**].

The chapter then outlines the basic use cases, along with descriptions of users and their privileges. It includes activity diagrams for some of the more complex use cases, providing greater detail on the processes within the application and its interactions with external services. Finally, the chapter presents a survey of similar applications, including their history and contributions to sports associations.

4.1 Application requirements

The major objective of the requirements engineering is defining the purpose of a propose system and outlining its external behavior. [3]. This section will specify the functional and non-functional requirements that the application must meet. For this bachelor's thesis, the requirements were established through communication with the individuals responsible for assigning matches, based on their preferences.

4.1.1 Functional requirements

Login part:

(F1) User registration and email confirmation during registration

Only referees and admins will be able to register for the application, i.e. registration is possible only from specific email addresses. We will ensure user authentication by following email confirmation.

(F2) User login and password reset option

Before redirecting after successful login, the user selects which part of the application they wish to access (e.g., admin or referee). A user may have multiple roles assigned simultaneously, as some referees can also be involved in the delegation process. During authorization, the system verifies whether the user holds the required role for the selected part. Access is granted only if the user is authorized for the chosen section.

Referee part:

(F3) Submission of excuses and vehicle information by referees

The referee must be able to create an excuse and specify the availability of a vehicle for a selected period of time. They should also be able to add a note to the excuse to assist the delegation department in their decision-making. Additionally, the referee must have the ability to view a preview of their submitted excuses and vehicle availability.

Admin part:

(F4) Viewing the overview of matches

Filtering by time and sorting matches according to various criteria, displaying detailed information about each match including the currently delegated referees, tracking who last modified the match and when, and allowing matches to be locked and played.

(F5) Uploading scheduled and played matches from a file

Files generated by the internal FACR system in .xlsx format record matches played and track how many times this season a referee served as referee or assistant referee for a given team.

(F6) Uploading current game data

The admin must be able to specify the current start of the game weekend. Based on this date, the time availability of all referees is displayed, and each referee's schedule is graphically represented in their personnel card.

(F7) Uploading referee information from a file

Referees are sorted according to the competitions listed in the nomination list. Their contact information (email, phone number, age) is obtained from .xlsx files edited by PFS during and before the season.

(F8) Automatic grouping of consecutive matches and manual grouping option

For automatic connection, matches must satisfy the defined time criteria to be considered connectable. Manual grouping allows setting pre-match and post-match for individual matches. Since the same referees are often delegated to consecutive matches, the graphical representation supports the delegation process.

(F9) Calculation of travel time between matches by car or bus

Whenever a referee is delegated to a match, the application must check the preceding and following matches. If they meet the specified time criteria and the exact locations of the football fields are provided, the route between the matches must be calculated, by car or bus, depending on the referee's vehicle availability at the time of the match. If the referee cannot reach the next match in time, the admin should receive a warning indicating the number of minutes by which the referee would be late.

(F10) Adding new referees and updating referee information

The admin must be able to add new referees to the system and edit existing referee information. This includes entering basic personal details (name, email, phone), FACR ID, league level, and general vehicle availability. The system should validate required fields and ensure data consistency across all parts of the application.

(F11) Calculation of points for referees based on the selected match

The application must use a point-based algorithm to support and accelerate the delegation process. Each referee is scored based on factors such as previous match history, travel distance, and recent assignment frequency. Referees who are vetoed by the team included in the match or have exceeded a threshold of delegations for the same team are excluded from the candidate pool. The referee with the highest remaining score should be delegated to the match.

(F12) Assigning and removing referees from matches

The application must check referee time availability for a given match based on submitted excuses and existing delegations of referee. If the referee is unavailable for that match, the system must display a warning to the administrator. However, the administrator may still proceed with the delegation despite the warning.

(F13) View referee overview

It must include current excuses, vetoes, referee information, and the referee's schedule relative to the date of the current match. It must support adding a veto to a referee, either for an entire club or for a specific team within a competition.

(F14) Downloading an Excel file with current matches and their delegations

It must be possible to export an .xlsx file in the same format as that generated by the internal FAČR system. This file will be used for manually uploading delegated referees to the official FAČR system.

4.1.2 Non-functional requirements

(N1) Availability as a web application and develop with ASP.NET core

The application must be available as a web application at www.rozhodcipraha.cz and app must be developed using ASP.NET core based framework.

(N2) Responsive design for login and referee part

It is expected that referees will most often enter excuses via mobile devices. Therefore, it is essential to have a user interface that is simple and fully responsive and optimized for mobile use, ensuring a smooth and user-friendly experience.

(N3) Real-time match updates in the match overview

To prevent conflicts during the delegation process in scenarios where multiple administrators assign referees to the same match concurrently.

(N4) Observance of the Principle of Least Privilege

Since the databases contain sensitive information, for the security of the application, each component must have access only to the databases it requires, and no others.

4.1.3 Use case specification models

Use case analysis focuses on outlining the interactions between the system and its actors. The use cases presented below reflect the application's behavior as specified in the functional requirements from the preceding section. They provide a user-centric overview of the primary processes within the application, organized into logical segments to support further design and development. Use case diagram see Figure

4.1.3.1 User roles

Users are assigned roles that determine their access rights. These roles will be used as a basis for describing the use cases.

- **1. Unauthenticated user-** A user who has not signed into the application and therefore lacks access to any features, able only to access the login and registration forms.
- **2. Referee-** A member of the PFS or another association serving as a referee, who is logged into the application. They do not have access to the delegation process and can only submit their excuses and vehicle preferences.
- **3. Admin-** The logged-in user, usually a member of the PFS Referee Committee, operates the admin section of the application, where they often perform routine tasks such as assigning matches, adjusting match details, and other related actions.
- **4. Main admin-** The logged-in user with the highest privileges. They can perform routine tasks and, in addition, have the privileges to load new matches into the application and upload played ones.

4.1.3.2 List of use cases

UC1 - Registration

Description: Creating a user account using the registration form. **Role:** Unauthenticated user **Basic path:**

1. The user opens the main page of the application , i.e. according to N1 www.rozhodcipraha.cz and clicks the ”Register” button on the top right.

2. The user fills in the registration form with their details and submits the information by clicking ”Register”.

3. If the data is incorrect, the user receives an error message. If the data is correct, the application checks whether the email address is on the list of allowed addresses. If the address is not present, the user sees an error message and is not allowed to register with that email address.

4. Upon successful registration, the user is shown a pop-up window informing them that a confirmation link has been sent to the email address provided during registration.

5. The user verifies their account by clicking the link in the email. This completes the registration process successfully.

UC2 - Login

Description: Allows a user with an already registered account to log in to a selected part of the application. **Role:** Unauthenticated user **Basic path:**

1. The user opens the main page of the application (i.e., as per N1, www.rozhodcipraha.cz) and clicks the ”Login” button in the top-right corner.

2. The user fills in the login form with their credentials, selects the part of the application they want to access, and submits the information by clicking the ”Login” button.

3. If the credentials are incorrect, the user receives a ”Login failed” error message. If the credentials are correct, the application generates a user token and redirects the user to the selected part of the application, where the system verifies if the user has permission to access it.

4. If the token is valid and access is granted, the desired section is displayed and the login is successful. If token validation fails, the application redirects the user back to the login page.

UC3 – Enter excuses

Description: Allows the referee to enter future excuses (time unavailability). **Role:** Referee **Basic path:**

1. The referee clicks the ”Zadat novou omluvu” button in the Referee section.

2. The application displays an entry form where the referee fills in the time range of the excuse and, if necessary, adds a note to clarify the circumstances.

3. The excuse is added to a temporary list by clicking the ”Přidat” button. The application checks the validity of the entered data, especially whether the time range is specified. If the required information is missing, the excuse will not be added and an error message will be shown.

4. If the user wants to enter another excuse, they continue with step 2. If they want to save the list of already entered excuses, they click the ”Send” button, which submits the excuses to the system.

UC4 – Assign the referee to a match or remove the referee from a match

Description: Assign any referee to any match with a suitability check, and remove any already assigned referee from their match. **Role:** Admin,MainAdmin
Basic path:

1. If the admin wants to remove a referee from a match, they simply click the input field associated with the referee in the match. The application then removes the button from the match panel and updates its color on the referee tab panel according to the new time availability.

2. If the admin wants to assign a referee to a match, they first click the referee’s button in the league tab panel or elsewhere on the screen.

3. The application colors this button blue. Then, the user can choose which match they would like to assign the selected referee to by clicking the corresponding input field.

4. The referee’s time and veto availability are evaluated with regard to the match. If the referee is available, a button is added to the input field, and the button in the league tab panel is visually updated to reflect the new availability.

5. If the referee misses a match due to transfers between matches, the application will display a warning showing the number of minutes by which the referee would be late. If this is caused by the referee having overlapping matches or an absence at that time, a warning will be displayed indicating that the referee is not suitable for the match due to unavailability. Alternatively, if the referee has a veto on at least one of the teams involved, a warning will be displayed stating that the referee is not suitable for the match due to the veto.

6. The warning can be overridden by the admin by clicking the button next to the warning, and the referee will then be assigned.

UC5 – Manually link follow-up matches

Description: Connect matches that were not automatically connected by the application (e.g., matches played on different fields within the same sport complex). **Role:** Admin,MainAdmin **Basic path:**

1. The admin usually sorts the matches by game field, in any order, by clicking the corresponding box next to ”Seřadit dle”.
2. The application sorts the matches and displays them in the correct order, or shows an error if the sorting was unsuccessful.
3. The admin goes through the matches one by one. If they find any matches that should be connected but aren't, they record the ID of the previous match.
4. Each match panel has two slots for linking connected matches. The admin fills the first slot of the second match with the recorded ID and clicks the ”Uložit” button on that match panel.
5. If the connection was successful, the application will show the user a success message; if not, an error message is displayed.
6. After reloading, the match panels will be colored the same, and the admin can navigate from one to the other by double-clicking the ID, which behaves like a link.

UC6 – Add new referee to the application

Description: During the football season, many people are interested in serving as referees under PFS, so the admin adds new referee to the application. **Role:** Admin, MainAdmin **Basic path:**

1. Clicks on the ”Přidat nového rozhodčího” button on the top bar of the default page.
2. The application displays a form for adding a new referee, containing various input fields.
3. The admin fills in these input fields based on information provided by the new referee, then assigns them to one of the available leagues.
4. The input fields are validated. If the data is invalid, the admin will be prompted to correct it. If the data is valid, the application checks whether a referee with the same name or email already exists in the system; if so, an error message is displayed. If the data is unique, a success message is shown.
5. After reloading the page, the referee's button is added to the relevant league referee group.

UC7 – Add veto for specific referee

Description: Some referees are not allowed to referee matches for certain teams or even entire clubs, so admin assigns a veto to referee. **Role:** Admin,MainAdmin **Basic path:**

1. The admin right-clicks the button of the referee to which he wants to assign a veto.

2. The referee overview is displayed, and at the bottom, previously added veto entries appear along with input fields for adding a new one.

3. After typing three characters in the ”Team name” input field, the application will display all teams that contain the entered string. The admin can then select one of the suggested teams. Next, the admin specifies whether the veto should apply to the entire club by selecting the ”all” option, or only to the selected team, and chooses the name of the competition in which the team participates.

4. Add a note in the ”Note” input field and save it using the blue plus button at the end of the line. The application will display a success message if the note is saved successfully. After the page is reloaded, the veto is displayed in the referee’s report.

UC8 – Uploading previous delegations of already played matches

Description: Sometimes, certain matches may not be parsed from the file containing played matches. The admin manually adds delegations and mark the match as played. **Role:** Admin, MainAdmin **Basic path:**

1. The admin fills in the date range next to ”Zadejte rozmezí filtrace” and clicks the ”Filtrovat” button. This action retrieves the matches that took place within the specified date range.

2. The admin identifies matches that are not marked as played and assigns referees based on an official source, such as the FAČR. Then, the ”Odehrát” button is clicked.

3. After this action, the application responds with a success or error message. The changes become visible after the page reloads.

4.1.4 Activity diagrams

I provide two activity diagrams that follow the Assign the referee to a match or remove the referee from a match use case and the Add new referee to the application use case.

The first diagram 4.1 illustrates the straightforward validation of the data entered by the admin, including checks for the uniqueness of the email address, name, and FACR ID. After successful validation, a new entity is created, which triggers communication between two independent parts of the application. This call from the admin part is not routed through the Nginx; instead, the login service is accessed directly via its IP address on the shared Docker network. This solution was necessary to allow the addition of an email address to the system during the registration of a new referee. Once the email address is added, the referee is able to register in the system.

In the second diagram 4.2, we can observe the progression of what is presumably the most frequently used functionality in the application. Since this functionality is expected to be used often, it makes sense to implement a shared cache of matches on the web server, especially given that the application will run on a single server instance [4].

When checking the delegation, all time-relevant matches to which the referee is assigned, as well as all relevant submitted excuses, are taken into account. If information about field locations is available, the referee's means of transport at the time of each match is determined, and the travel time between matches is automatically calculated using an API call to an external service. Since the number of matches per day can be high, accounting for transfers is also a critical part of the delegation process.

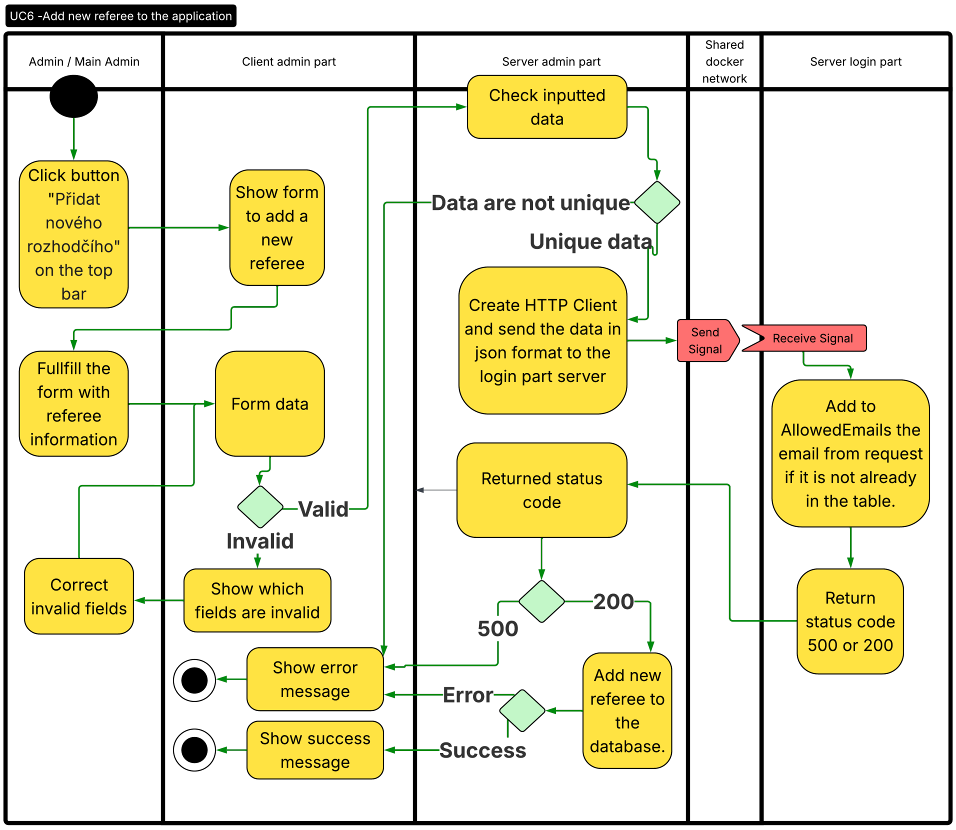
Once a referee is assigned to a match, these changes must be reflected to the user in near real time, using a library that supports server-to-client updates in order to avoid collisions.

4.2 Existing solutions

There is a limited amount of publicly available information regarding similar solutions. In most cases, football associations are too small to warrant the implementation of a systematic delegation process via specialised software. For such organisations, these solutions prove impractical, as the number of weekly match assignments is minimal, and the use of software would likely prolong rather than streamline the process.

In contrast, larger city football associations—such as Prague football association—often rely on systems developed by national football federations (e.g., DFBnet). However, these systems are typically undocumented or not publicly disclosed. In other cases, associations may use customised software developed specifically for their needs (e. g., RefAssist), with vendors opting not to publish comprehensive functional documentation to protect proprietary knowledge.

A further alternative, as noted in the preceding section, is manual delegation—using spreadsheets and documents without the aid of a dedicated software solution.


UC6 -Add new referee to the application
Admin / Main Admin
Client admin part
Server admin part
Shared docker network
Server login part
Click button "Přidat nového rozhodčího" on the top bar
Show form to add a new referee
Fullfill the form with referee information
Form data
Valid
Invalid
Show which fields are invalid
Correct invalid fields
Check inputted data
Data are not unique
Unique data
Create HTTP Client and send the data in json format to the login part server
Send Signal
Receive Signal
Add to AllowedEmails the email from request if it is not already in the table.
Return status code 500 or 200
Returned status code
200
500
Add new referee to the database.
Error
Success
Show error message
Show success message

■ **Figure 4.1** UML activity diagram illustrating the adding new referee to the application, created in Lucidchart. [5]

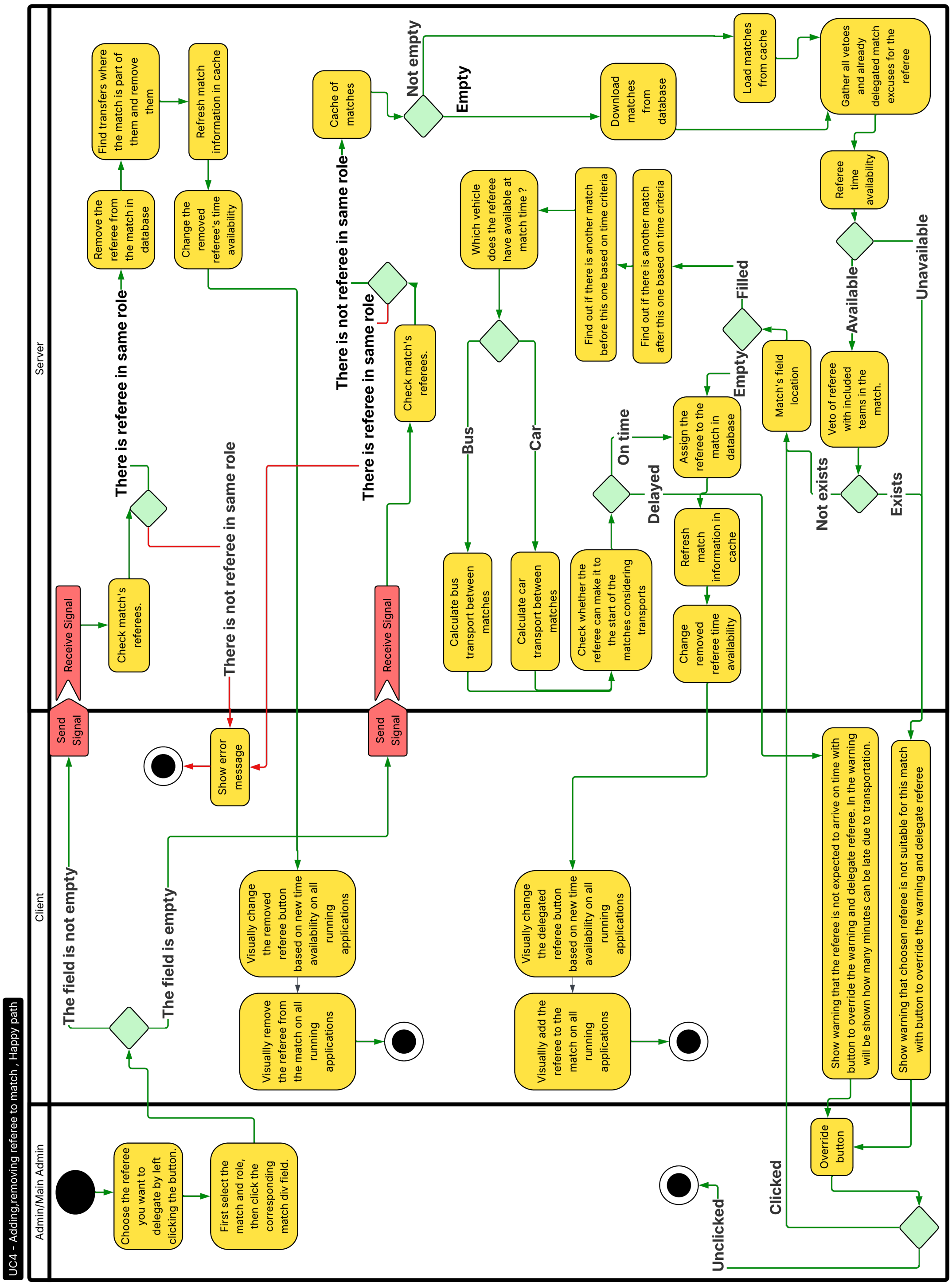

UC4 - Adding,removing referee to match , Happy path
Admin/Main Admin
Client
Server
Choose the referee you want to delegate by left clicking the button.
First select the match and role, then click the corresponding match div field.
The field is not empty
The field is empty
Send Signal
Receive Signal
Check match's referees.
There is referee in same role
There is not referee in same role
Show error message
Remove the referee from the match in database
Find transfers where the match is part of them and remove them
Refresh match information in cache
Change the removed referee's time availability
Visually change the removed referee button based on new time availability on all running applications
Visually remove the referee from the match on all running applications
Send Signal
Receive Signal
Check match's referees.
There is not referee in same role
There is referee in same role
Cache of matches
Not empty
Empty
Download matches from database
Load matches from cache
Gather all vetoes and already delegated match excuses for the referee
Referee time availability
Available
Unavailable
Veto of referee with included teams in the match.
Not exists
Exists
Match's field location
Filled
Empty
Find out if there is another match after this one based on time criteria
Find out if there is another match before this one based on time criteria
Which vehicle does the referee have available at match time ?
Bus
Car
Calculate bus transport between matches
Calculate car transport between matches
Check whether the referee can make it to the start of the matches considering transports
On time
Delayed
Assign the referee to the match in database
Refresh match information in cache
Change removed referee time availability
Visually change the delegated referee button based on new time availability on all running applications
Visually add the referee to the match on all running applications
Show warning that the referee is not expected to arrive on time with button to override the warning and delegate referee. In the warning will be shown how many minutes can be late due to transportation.
Show warning that choosen referee is not suitable for this match with button to override the warning and delegate referee
Override button
Clicked
Unclicked

Figure 4.2 UML activity diagram illustrating the assignment and removal of a referee from a match, created in Lucidchart. [5]

4.2.1 RefAssist

The founders of RefAssist started their own complex planning consulting agency in 2015. A request from the Royal Belgian Football Association for referee planning prompted the idea to build a standalone software application. [6]

According to the chairman of the Antwerp Arbitration Office, approximately 650 matches are scheduled each week, with around 550 referees available. Assigning referees to all matches presents a significant challenge due to numerous constraints: certain referees cannot be assigned to specific clubs due to past incidents, and repeated appointments to the same club must be avoided to ensure fairness and neutrality.

Previously, this complex planning task was carried out entirely manually, often requiring up to two full working days per week. With the implementation of the application, the scheduling process has been dramatically streamlined. The algorithm assigns referees to matches within 30 minutes, after which only minor manual adjustments are necessary.

The development team collected and analyzed existing planning heuristics — previously managed through manual navigation of multiple Excel files — and incorporated them into the digital system.After just one season of using the application, referee travel allowances were reduced by 18.8%, reflecting increased planning efficiency. [**refAssist_planning_flames**]

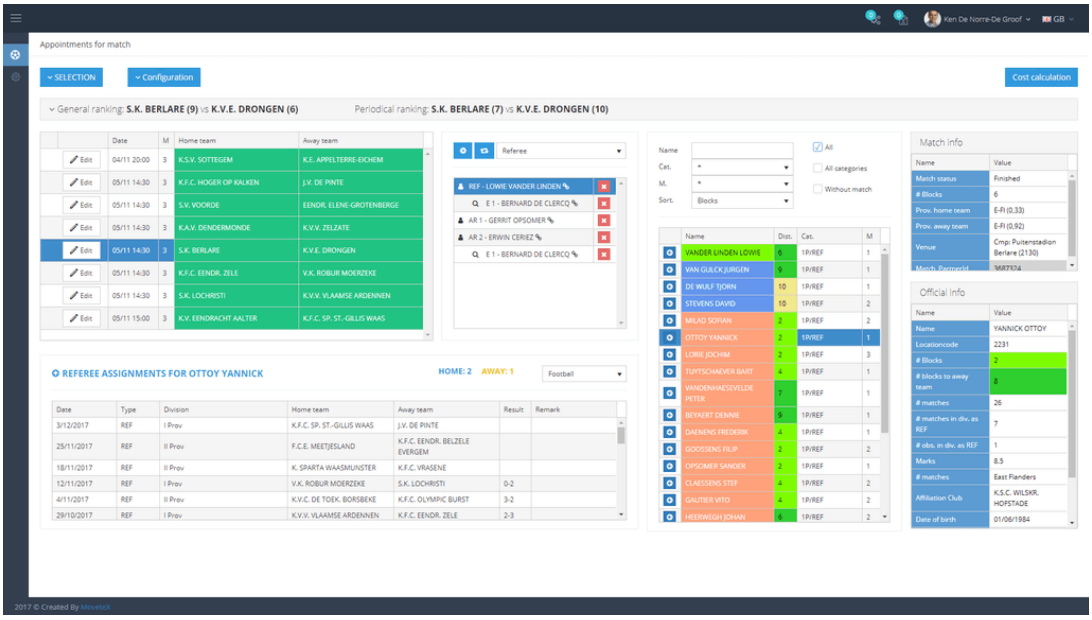
- The current team responsible for development of the application comprises 14 members, of whom 7 are software developers. [6]
- Collaboration with the Belgian Football Association encompasses 4,500 referees and approximately 300,000 scheduled matches per season
- According to Belgian appointment officers, the appointment process became 80% more time-efficient. In this case, transport expenses were rationalized drastically by 25% to 30%. [**refAssist_planning_belgium**]
- The application is utilized by the Australian Football League, the governing body for Australian football — a sport akin to rugby — and supports a network of 15,000 referees.

Due to the lack of information about the implementation of the given functionalities, we can only evaluate the appearance of the UI.

On the screen (shown in Figure 4.3), we can see that the application already includes some elements of data visualization—for example, when displaying the distance from the match. However, in my opinion, this is not sufficient, and the main screen is not very clear.

4.2.2 DFBnet

The software is developed and operated by DFB GmbH, a wholly owned subsidiary of the German Football Association (DFB), and is therefore part of the



Appointments for match

General ranking: **S.K. BERLARE (9)** vs **K.V.E. DRONGEN (6)** Periodical ranking: **S.K. BERLARE (7)** vs **K.V.E. DRONGEN (10)**

	Date	M	Home team	Away team
Edit	04/11 20:00	3	K.S.V. SOTTEGEM	K.E. APPELTERRE-EICHEM
Edit	05/11 14:30	3	K.F.C. HOGER OP KALKEN	J.V. DE PINTE
Edit	05/11 14:30	3	S.V. VOORDE	EENDR. ELENE-GROTENBERGE
Edit	05/11 14:30	3	K.A.V. DENDERMONDE	K.V.V. ZELZATE
Edit	05/11 14:30	3	S.K. BERLARE	K.V.E. DRONGEN
Edit	05/11 14:30	3	K.F.C. EENDR. ZELE	V.K. ROBUR MOERZEKE
Edit	05/11 14:30	3	S.K. LOCHRISTI	K.V.V. VLAAMSE ARDENNEN
Edit	05/11 15:00	3	K.V. EENDRACHT AALTER	K.F.C. SP. ST.-GILLIS WAAS

REF - LOWIE VANDER LINDEN
E 1 - BERNARD DE CLERCQ
AR 1 - GERRIT OPSOMER
AR 2 - ERWIN CERIEZ
E 1 - BERNARD DE CLERCQ

Name
Cat. *
M. *
Sort. Blocks
☑ All
□ All categories
□ Without match

	Name	Dist.	Cat.	M
⊕	VANDER LINDEN LOWIE	6	1P/REF	1
⊕	VAN GULCK JURGEN	9	1P/REF	1
⊕	DE WULF TJORN	10	1P/REF	1
⊕	STEVENS DAVID	10	1P/REF	2
⊕	MILAD SOFIAN	2	1P/REF	2
⊕	OTTOY YANNICK	2	1P/REF	1
⊕	LORIE JOCHIM	2	1P/REF	3
⊕	TUYTSCHAEVER BART	4	1P/REF	1
⊕	VANDENHAESEVELDE PETER	7	1P/REF	1
⊕	BEYAERT DENNIE	9	1P/REF	1
⊕	DAENENS FREDERIK	4	1P/REF	1
⊕	GOOSSENS FILIP	2	1P/REF	2
⊕	OPSOMER SANDER	2	1P/REF	1
⊕	CLAESSENS STEF	4	1P/REF	2
⊕	GAUTIER VITO	4	1P/REF	2
⊕	HEERWEGH JOHAN	6	1P/REF	2

Match info

Name	Value
Match status	Finished
# Blocks	6
Prov. home team	E-FI (0,33)
Prov. away team	E-FI (0,92)
Venue	Cmp: Puitenstadion Berlare (2130)
Match PartnerId	3687324

Official info

Name	Value
Name	YANNICK OTTOY
Locationcode	2231
# Blocks	2
# blocks to away team	8
# matches	26
# matches in div. as REF	7
# obs. in div. as REF	1
Marks	8.5
# matches	East Flanders
Affiliation Club	K.S.C. WILSKR. HOFSTADE
Date of birth	01/06/1984

REFEREE ASSIGNMENTS FOR OTTOY YANNICK HOME: 2 AWAY: 1

Date	Type	Division	Home team	Away team	Result	Remark
3/12/2017	REF	I Prov	K.F.C. SP. ST.-GILLIS WAAS	J.V. DE PINTE		
25/11/2017	REF	II Prov	F.C.E. MEETJESLAND	K.F.C. EENDR. BELZELE EVERGEM		
18/11/2017	REF	II Prov	K. SPARTA WAASMUNSTER	K.F.C. VRASENE		
12/11/2017	REF	I Prov	V.K. ROBUR MOERZEKE	S.K. LOCHRISTI	0-2	
4/11/2017	REF	II Prov	K.V.C. DE TOEK. BORSBEKE	K.F.C. OLYMPIC BURST	3-2	
29/10/2017	REF	I Prov	K.V.V. VLAAMSE ARDENNEN	K.F.C. EENDR. ZELE	2-3	

2017 © Created By MoveteX

Figure 4.3 RefAssist interface displaying the functionality for assigning referees to matches. [7]

main official system.All software modules grouped under the umbrella brand DFBnet have access to a central database. The DFBnet modules are developed using the Java EE programming platform and are provided to authorized users as decentralized web applications. [8]

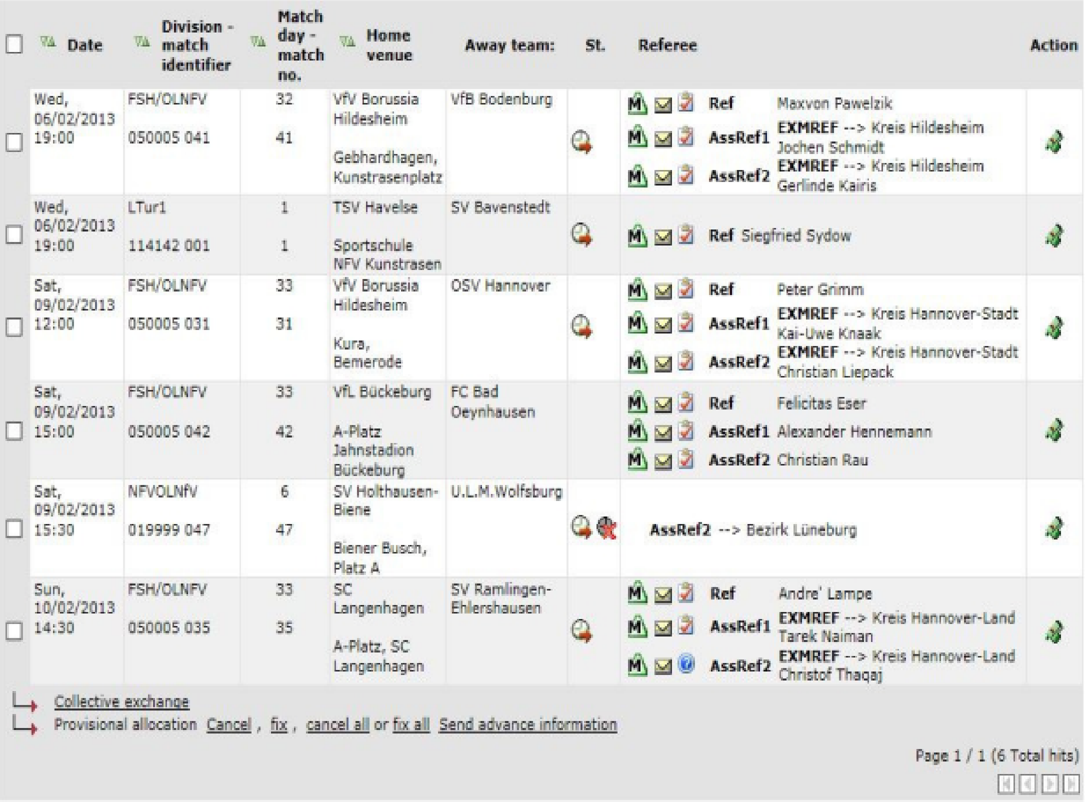
The referee appointment module assigns referees to matches based on availability and qualifications. The intelligent assignment, mainly used in amateur leagues, distributes referees from a pool to fixtures over multiple runs. Only referees who meet all match requirements and live closest to the venue are considered. [8]

- It is used by the DFB and Bundesliga as well as the 26 state and regional associations.
- Is the basis for 78,000 referees.
- Handles 70,000 matches per weekend.

The biggest advantage is the direct access to relevant data from the central information system, which eliminates the need for manual data imports—such as uploading scheduled and played matches every week—thereby significantly speeding up processes. I also appreciate the system's ability to assign referees to specific competitions, which is an essential step towards enabling automatic delegation. The automatic delegation feature is based on parameters that can be prioritized or limited by maximum values. These parameters include the distance from the referee's home to the match location, the total number of assigned matches, and the number of days since the referee last refereed a match involving one of the teams. This differs from my current approach, where only the total number of such matches is tracked.

However, I see the system's robustness as a minor disadvantage(shown in Figure 4.4). To me, as a layman, it appears opaque and lacks graphical indicators that could support efficient and high-quality decision-making. [9]. Another disadvantage is the handling of "problem clubs." The system only allows a referee to veto an entire club (i.e., all its teams in all competitions); it does not support vetoing individual teams.

According to the information I found, the system has not undergone major changes since its launch in 2012—only minor functionalities have been added. [10]



□	Date	Division - match identifier	Match day - match no.	Home venue	Away team:	St.	Referee	Action
□	Wed, 06/02/2013 19:00	FSH/OLNFV 050005 041	32 41	VfV Borussia Hildesheim Gebhardhagen, Kunstrasenplatz	VfB Bodenburg		Ref Maxvon Pawelzik AssRef1 EXMREF --> Kreis Hildesheim Jochen Schmidt AssRef2 EXMREF --> Kreis Hildesheim Gerlinde Kairis	
□	Wed, 06/02/2013 19:00	LTur1 114142 001	1 1	TSV Havelse Sportschule NFV Kunstrasen	SV Bavenstedt		Ref Siegfried Sydow	
□	Sat, 09/02/2013 12:00	FSH/OLNFV 050005 031	33 31	VfV Borussia Hildesheim Kura, Bemerode	OSV Hannover		Ref Peter Grimm AssRef1 EXMREF --> Kreis Hannover-Stadt Kai-Uwe Knaak AssRef2 EXMREF --> Kreis Hannover-Stadt Christian Liepack	
□	Sat, 09/02/2013 15:00	FSH/OLNFV 050005 042	33 42	VfL Bückeburg A-Platz Jahnstadion Bückeburg	FC Bad Oeynhausen		Ref Felicitas Eser AssRef1 Alexander Hennemann AssRef2 Christian Rau	
□	Sat, 09/02/2013 15:30	NFVOLNfV 019999 047	6 47	SV Holthausen-Biene Biener Busch, Platz A	U.L.M.Wolfsburg		AssRef2 --> Bezirk Lüneburg	
□	Sun, 10/02/2013 14:30	FSH/OLNFV 050005 035	33 35	SC Langenhagen A-Platz, SC Langenhagen	SV Ramlingen-Ehlershausen		Ref Andre' Lampe AssRef1 EXMREF --> Kreis Hannover-Land Tarek Naiman AssRef2 EXMREF --> Kreis Hannover-Land Christof Thaqaj	

Collective exchange

Provisional allocation Cancel , fix , cancel all or fix all Send advance information

Page 1 / 1 (6 Total hits)

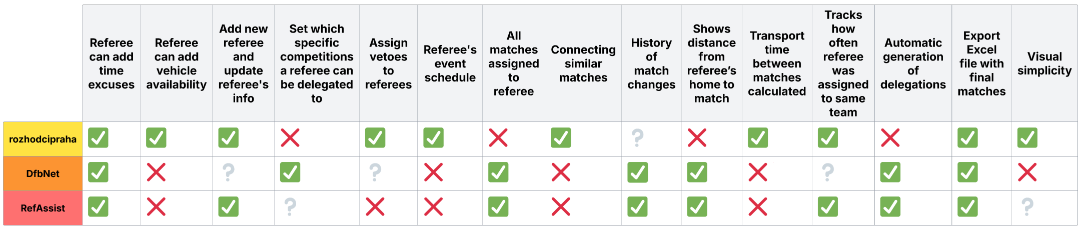
Figure 4.4 DFBnet interface displaying the functionality for assigning referees to matches. [10]

4.2.3 Functionalities of the mentioned solutions

First, a list of fourteen important functionalities was compiled based on the current functional requirements and potential future developments of the application. Next, each application was examined to determine whether it includes these functionalities. This overview is shown in the below Figure 4.5.The question mark in the overview means that it is uncertain whether the solution includes the functionality—this could be due to missing information, inability to locate it, or ambiguity in whether the functionality is fully implemented.

The examined functionalities are as follows:

- Referee can add time excuses.
- Referee can add vehicle availability.
- Add new referee and update referee's info.
- Set which specific competitions a referee can be delegated to.
- Assign vetoes to referees.
- Referee's event schedule.
- All matches assigned to referee.
- Connecting similar matches.
- History of match changes.
- Shows distance from referee's home to match.
- Transport time between matches calculated.
- Tracks how often referee was assigned to same team.
- Automatic generation of delegations.
- Export Excel file with final matches.



	Referee can add time excuses	Referee can add vehicle availability	Add new referee and update referee's info	Set which specific competitions a referee can be delegated to	Assign vetoes to referees	Referee's event schedule	All matches assigned to referee	Connecting similar matches	History of match changes	Shows distance from referee's home to match	Transport time between matches calculated	Tracks how often referee was assigned to same team	Automatic generation of delegations	Export Excel file with final matches	Visual simplicity
rozhodcipraha	✓	✓	✓	✗	✓	✓	✗	✓	?	✗	✓	✓	✗	✓	✓
DfbNet	✓	✗	?	✓	?	✗	✓	✗	✓	✓	✗	?	✓	✓	✗
RefAssist	✓	✗	✓	?	✗	✗	✓	✗	✓	✓	✗	✓	✓	✓	?

Figure 4.5 Overview of the functionalities of all applications , created in Lucidchart. [5]

Chapter 5
Design

This chapter provides an overview of the technologies used, the external services required for the proper functioning of the application, its architecture, and a description of the deployment on the hosted server.

5.1 Technologies

5.1.1 Razor

Razor is a markup syntax for embedding .NET based code into webpages. The Razor syntax consists of Razor markup, C#, and HTML. Nowadays, other technology called Razor Pages technology is gaining popularity because it combines the model and controller logic within the Razor Page itself [11]. However, I chose to follow the MVC architecture and use Razor syntax in the views because MVC has been established for a long time, I have experience with it, it is proven, and there is extensive documentation available. Splitting the code into Models, Views, and Controllers makes more sense to me for this application. That said, I am open to using Razor Pages in future projects.

5.1.2 jQuery

jQuery is a fast, small JavaScript library that simplifies HTML document traversal, manipulation, event handling, animation, and Ajax. Its easy-to-use API works across many browsers [12]. I chose jQuery for my project because it allows me to write cleaner and shorter code for AJAX calls and other tasks, and it supports method chaining, which can be difficult to implement otherwise. I excluded other JavaScript frameworks because, for developing a smaller, graphically simpler application, jQuery is sufficient—it is proven and requires less maintenance in the future. [13]

5.1.3 C#

It is an object-oriented programming language created by Microsoft that runs on the .NET Framework. C# has roots from the C family, and the language is close to other popular languages like C++ and Java.

5.1.4 ASP.NET Core

ASP.NET Core is a powerful, open-source, and cross-platform framework designed for building modern web applications. It supports development and deployment on Windows, macOS, and Linux. This feature will be leveraged when developing on Windows operating system and deploying to a production server running Linux. An additional advantage is the integration with Azure Key Vault, as both technologies are developed by Microsoft, resulting in a more seamless and secure configuration experience. [14] The application uses ASP.NET Core 8.0, which is a Long-Term Support (LTS) version and will be supported until November 2026.

5.1.5 AJAX

AJAX is the technique of exchanging data with a server and updating parts of a web page without reloading the entire page. jQuery provides several methods to handle AJAX functionality. Using jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server via both HTTP GET and POST. I choose AJAX because it is fully supported within jQuery and it still works well. [15]

5.1.6 SignalR

ASP.NET SignalR is a library that simplifies the process of adding real-time web functionality to applications. Real-time web functionality is the ability of server-side code to share content with connected clients immediately when it becomes available, instead of waiting for the client to request new data, as is the case with AJAX. My application falls under the peer-to-peer collaboration category because users are working on the same dataset and can see changes to it immediately. [16]

5.2 Architecture of the Application

The application is divided into three functional modules. The first is the login module, which handles user registration and authentication. The second module is intended for referees, allowing them to enter relevant match data. The third and main module is for administrators, where the system assigns referees to matches, supports delegation planning, and collects all available data within the system.

Each of these modules is based on the ASP.NET Core MVC (Model-View-Controller) architecture, which has been widely adopted as a design for web application. [17]They also incorporate elements of a multi-layer architecture (three-layer structure) as repository and service:

- ASP.NET Core routes the request to the appropriate controller.
- The controller retrieves the necessary data via a repository and calls service that encapsulates business logic.
- The controller then passes the retrieved data to the view.
- The view uses Razor syntax to generate the resulting HTML, which is returned to the user's browser.

5.3 Database

Relational databases are used in all these modules, because it is the best choice if database handle relatively small amounts of data with a predictable structure and size. [18]

5.3.1 PostgreSQL

PostgreSQL is currently among the most widely used database technologies.[19] One of its key strengths lies in its extensive support for various data types, including primitive types, structured types, geometric data, JSON, XML, and even user-defined custom types. This flexibility provides a advantage over many other database management systems. Moreover, PostgreSQL has demonstrated high performance in handling fundamental operations, particularly with datasets containing up to 100,000 records.[20]

5.3.2 Entity Framework and LINQ

In this project, the database schema will be designed and implemented prior to the development of the application layer. The Scaffold-DbContext command, provided by Entity Framework Core—an Object-Relational Mapper (ORM)—will then be used to reverse-engineer the database schema into the application as strongly typed data models and a corresponding database context class. This database-first approach is intended to facilitate the use of Language Integrated Query (LINQ) for type-safe, readable, and maintainable data access.

5.4 Database models

Domain models were created based on the functional requirements from section 4.1.1 and the nature of the data to be stored, taking into account the modular structure of the application. The figure 5.1 illustrates the login database ,figure 5.2 referee database, and 5.3 admin database.

In the following text, I will describe the selected important entities and define their relationships.

Login part: In the login part, most of the tables are generated using the ASP.NET Core Identity framework, whose structure and contents are publicly documented and accessible, for example, in [21].

5.4.1 AllowedEmailAddresses

This is the only custom table I created in this part. It is used to verify if a user has the right to register—i.e., it is essential for authenticating users. The table includes the following columns:

- Id
- Email — the user's email address
- Role — the role assigned to the user upon registration, which grants access to specific parts of the application: 0 = Main Admin and 1 = Admin (access only to the admin section) ,2 = Referee (access only to the referee section)

Referee part:

5.4.2 Referees

This table keeps up-to-date information about the referee:

- **referee_id** — unique referee identifier
- **user_id** — unique identifier linking to the ASP.NET Identity system
- **facr_id** — official referee ID from the Football Association of the Czech Republic
- **name** — referee's first name
- **surname** — referee's last name
- **email** — registration and contact email address
- **league** — league level where the referee primary referees matches

- **age** — referee's age
- **ofs** — indicates if the referee works under PFS (true) or not (false)
- **note** — additional comments or remarks, i.e., special referee requirements
- **prague_zone** — official place of residence
- **actuall_prague_zone** — current place of residence
- **car_availability** — whether the referee has access to a car in general
- **timestamp_change** — tracks when the record was last modified

5.4.3 Excuses

This table manages referee unavailability periods:

- **excuse_id** — unique excuse identifier
- **referee_id** — foreign key, links to the Referees table
- **date_from** — start date of unavailability
- **time_from** — start time of unavailability
- **date_to** — end date of unavailability
- **time_to** — end time of unavailability
- **datetime_added** — when the excuse was submitted
- **note** — additional note to clarify or elaborate the excuse details, making it easier for the admins to understand
- **reason** — reason provided if submitted after the 14-day deadline

The table **VehicleSlots** is very similar and boolean field **has_car_in_the_slot** indicates whether the referee has a car at the given time(true) or not(false).

The relationships between tables are maintained through foreign key constraints, where both `Excuses` and `VehicleSlots` tables reference the `Referees` table through the `referee_id` field, establishing a one-to-many relationship pattern.

Admin part:

5.4.4 Matches

This table contains comprehensive match information:

- **match_id** — unique match identifier, format exactly as in FACR official systems
- **competition_id** — competition of the match due to ie.match length , foreign key to Competitions table
- **field_id** — field where the match takes place ,foreign key to Fields table
- **home_team_id** — identifier for the home team, format exactly as in FACR official systems
- **away_team_id** — identifier for the visiting team, format exactly as in FACR official systems
- **match_date** — scheduled date of the match
- **match_time** — scheduled kickoff time
- **pre_match** — if there is a preceding pre-match, this is its match ID
- **post_match** — if there is a following post-match, this is its match ID
- **referee_id** — assigned referee identifier
- **ref1_id** — first assistant referee identifier
- **ref2_id** — second assistant referee identifier
- **note** — additional match notes or comments
- **already_played** — whether the match has already been played
- **locked** — indicates if the match is locked and should not be changed
- **last_changed_by** — admin who made the last modification
- **last_changed** — timestamp of the last modification

Many-to-Many Relationship (Teams Matches): The relationship between teams and matches is implemented through the `TeamsMatches` junction table with the following structure:

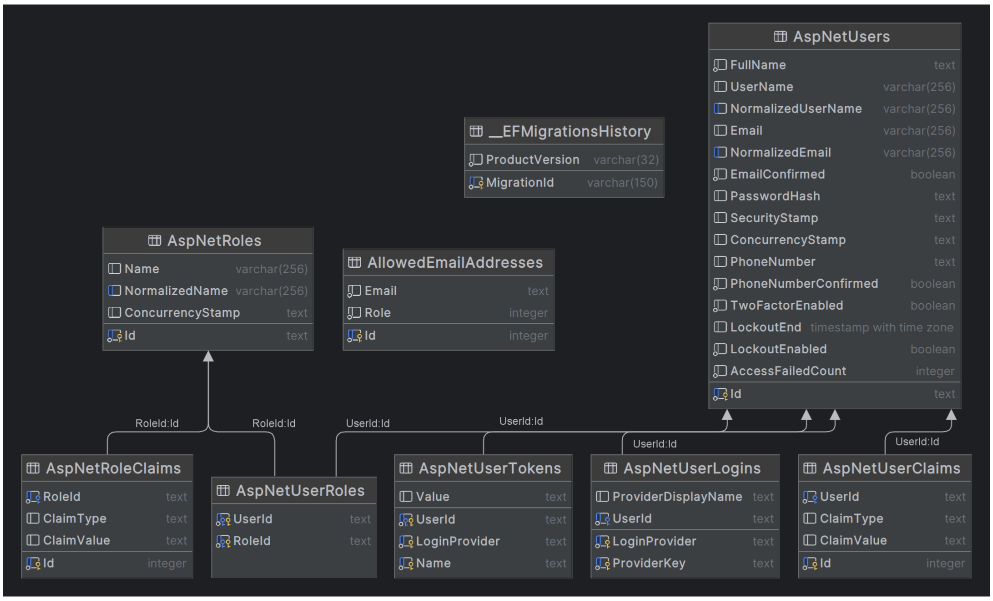
- **team_id** — foreign key referencing Teams table
- **match_id** — foreign key referencing Matches table
- **competition_id** — foreign key referencing Competitions table

- **field_id** — foreign key referencing Fields table

The composite primary key consists of all four fields: `(team_id, match_id, competition_id, field_id)`. This many-to-many relationship allows each team to participate in multiple matches across different competitions and fields, while each match can involve multiple teams.

Other Key Relationships:

- `Matches` → `Competitions` (many-to-one): Each match belongs to a specific competition via `competition_id`
- `Matches` → `Fields` (many-to-one): Each match is played at a specific field via `field_id`
- `Vetoes` → `Teams` and `Competitions` (many-to-one): Manages referee restrictions for delegations to specific team-competition combinations
- `Transfers`: Handles referee transportation between matches with references to previous and future matches
- `FilesPreviousDelegations`: Stores information about uploaded files with played matches
- `StartingGameDates`: Stores current game date



AspNetUsers	
FullName	text
UserName	varchar(256)
NormalizedUserName	varchar(256)
Email	varchar(256)
NormalizedEmail	varchar(256)
EmailConfirmed	boolean
PasswordHash	text
SecurityStamp	text
ConcurrencyStamp	text
PhoneNumber	text
PhoneNumberConfirmed	boolean
TwoFactorEnabled	boolean
LockoutEnd	timestamp with time zone
LockoutEnabled	boolean
AccessFailedCount	integer
Id	text

__EFMigrationsHistory	
ProductVersion	varchar(32)
MigrationId	varchar(150)

AspNetRoles	
Name	varchar(256)
NormalizedName	varchar(256)
ConcurrencyStamp	text
Id	text

AllowedEmailAddresses	
Email	text
Role	integer
Id	integer

AspNetRoleClaims	
RoleId	text
ClaimType	text
ClaimValue	text
Id	integer

AspNetUserRoles	
UserId	text
RoleId	text

AspNetUserTokens	
Value	text
UserId	text
LoginProvider	text
Name	text

AspNetUserLogins	
ProviderDisplayName	text
UserId	text
LoginProvider	text
ProviderKey	text

AspNetUserClaims	
UserId	text
ClaimType	text
ClaimValue	text
Id	integer

■ **Figure 5.1** Relational database model of login part, created in Datagrip. [22]

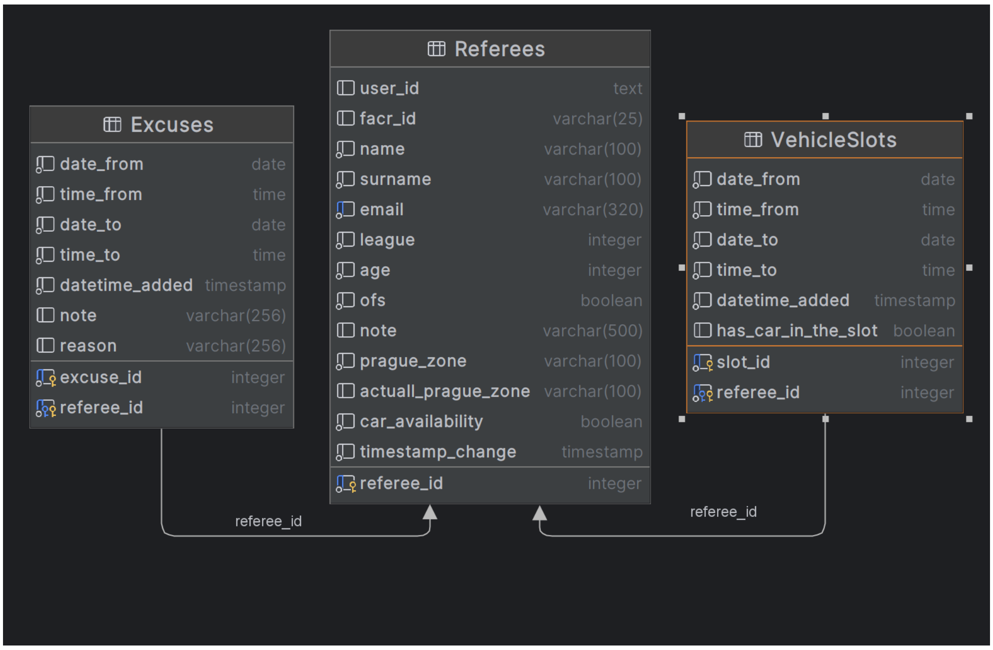


Figure 5.2 Relational database model of referee part, created in Datagrip. [22]

5.5 External services

5.5.1 SendGrid

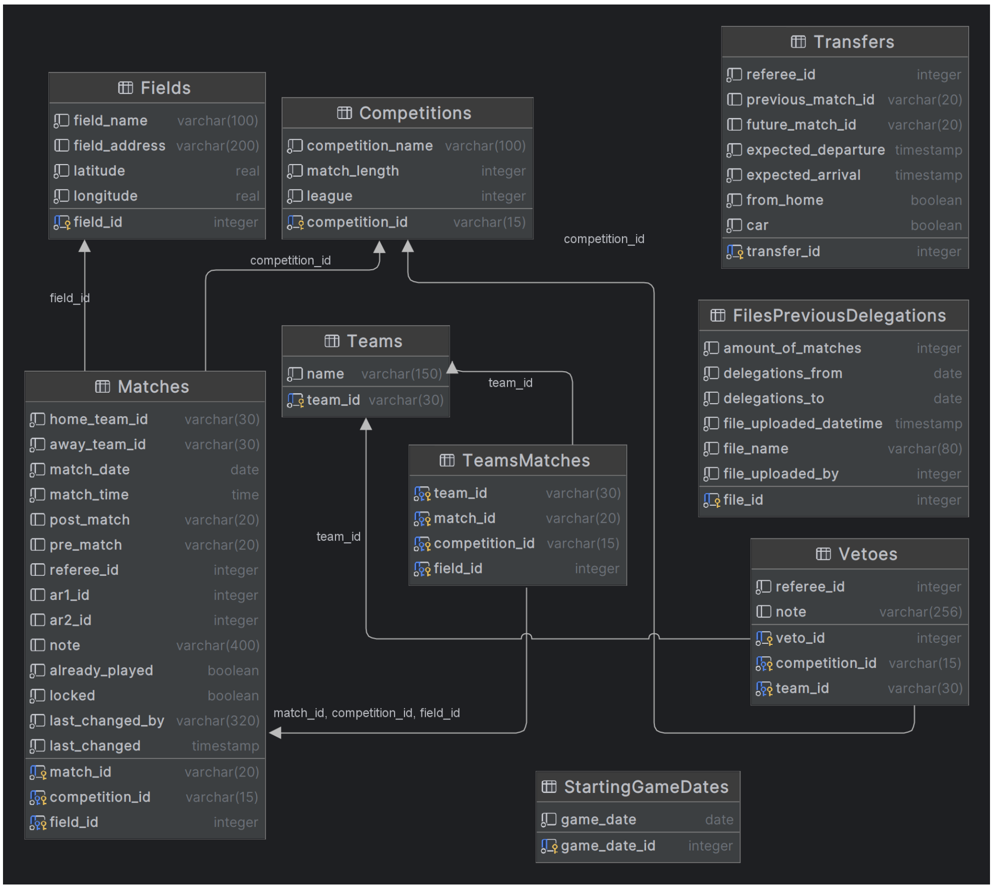
SendGrid provides a cloud-based service that assists businesses with email delivery. The service manages various types of email including shipping notifications, friend requests, sign-up confirmations.In the free plan, you are allowed to send a maximum of 100 emails per day, which is sufficient for this application. [23]Integration with .NET is also straightforward.

5.5.2 Azure Key Vault

It is necessary to securely store database passwords and API keys for external services; therefore, I chose to use Azure Key Vault, specifically its Secrets Management feature. As a service developed by Microsoft, it offers seamless integration with .NET projects. Additionally, access to stored secrets can be restricted to specific allowed IP addresses, enhancing security by preventing unauthorized access to the vault even in the event of leaked API keys. [24]

5.5.3 REST API Mapy.com

The Mapy.cz REST API is an interface provided by Seznam.cz that allows developers to access various features of the Mapy.cz map services via HTTP



Transfers

referee_id	integer
previous_match_id	varchar(20)
future_match_id	varchar(20)
expected_departure	timestamp
expected_arrival	timestamp
from_home	boolean
car	boolean
transfer_id	integer

Fields

field_name	varchar(100)
field_address	varchar(200)
latitude	real
longitude	real
field_id	integer

Competitions

competition_name	varchar(100)
match_length	integer
league	integer
competition_id	varchar(15)

FilesPreviousDelegations

amount_of_matches	integer
delegations_from	date
delegations_to	date
file_uploaded_datetime	timestamp
file_name	varchar(80)
file_uploaded_by	integer
file_id	integer

Teams

name	varchar(150)
team_id	varchar(30)

Matches

home_team_id	varchar(30)
away_team_id	varchar(30)
match_date	date
match_time	time
post_match	varchar(20)
pre_match	varchar(20)
referee_id	integer
ar1_id	integer
ar2_id	integer
note	varchar(400)
already_played	boolean
locked	boolean
last_changed_by	varchar(320)
last_changed	timestamp
match_id	varchar(20)
competition_id	varchar(15)
field_id	integer

TeamsMatches

team_id	varchar(30)
match_id	varchar(20)
competition_id	varchar(15)
field_id	integer

Vetoes

referee_id	integer
note	varchar(256)
veto_id	integer
competition_id	varchar(15)
team_id	varchar(30)

StartingGameDates

game_date	date
game_date_id	integer

Figure 5.3 Relational database model of admin part, created in Datagrip. [22]

requests, following REST API principles. The API supports route planning between two points for various types, including short car routes, fast car routes (with or without traffic), walking, and cycling. [25] Each route planning request costs 4 credits, and developers receive 250,000 free credits per month, allowing for up to 62,500 route searches — which is more than sufficient for our app. For the purpose of this app, I will use the car route planning type with traffic enabled. [26] Since the API does not support public transport, I will need to use a different service for that functionality.

5.5.4 Directions API

This robust and reliable service is part of the Google Maps Platform and enables route planning with support for various transportation methods (e.g., driving, walking, biking, and public transit). Since I already have a service for car navigation, I will only use the public transit mode. A major advantage of this service is the ability to specify a departure time, allowing the route to be calculated based on actual transit schedules. I will use this feature in my application because matches are usually played on weekends, when public transport runs less frequently. Without setting the correct departure time, the routes might be inaccurate. The free plan allows up to 10,000 API calls per month, which is sufficient for our application. [27]

5.6 Deployment of the application

The application will be deployed behind an Nginx server, which is to be configured as a reverse proxy and equipped with API gateway functionalities, including request authorization [28]. Access will be restricted exclusively to the HTTPS protocol by enforcing redirection of all HTTP traffic within the Nginx configuration. Secure communication will be achieved using TLS encryption with certificates provided by Let's Encrypt, a widely recognized certificate authority currently securing over 600 million websites. [29]

The application will be protected by the Nginx ModSecurity WAF, a Web Application Firewall (WAF) integrated with the Nginx web server using the ModSecurity module. It is used to protect web applications from application-layer attacks that could lead to system abuse or compromise. ModSecurity enables the analysis and filtering of HTTP traffic using defined rules, thus increasing the application's security without requiring changes to its source code. [30]

The .NET Core applications and databases will be deployed using Docker, as containerized services. For security reasons, the login component of the application will run in a container with reduced privileges (i.e., not as the root user). [31]

After successful authentication in the login module, a jwt token containing user role information will be generated and forwarded to an internal validation endpoint on referee or admin part, which is accessible only via Nginx. This endpoint verifies whether the user has the necessary permissions to access a specific part of the application. Upon successful validation, the Nginx redirects the user to the appropriate containerized component.

A shared emails network will be used to manage and update the list of email addresses that are permitted to register.

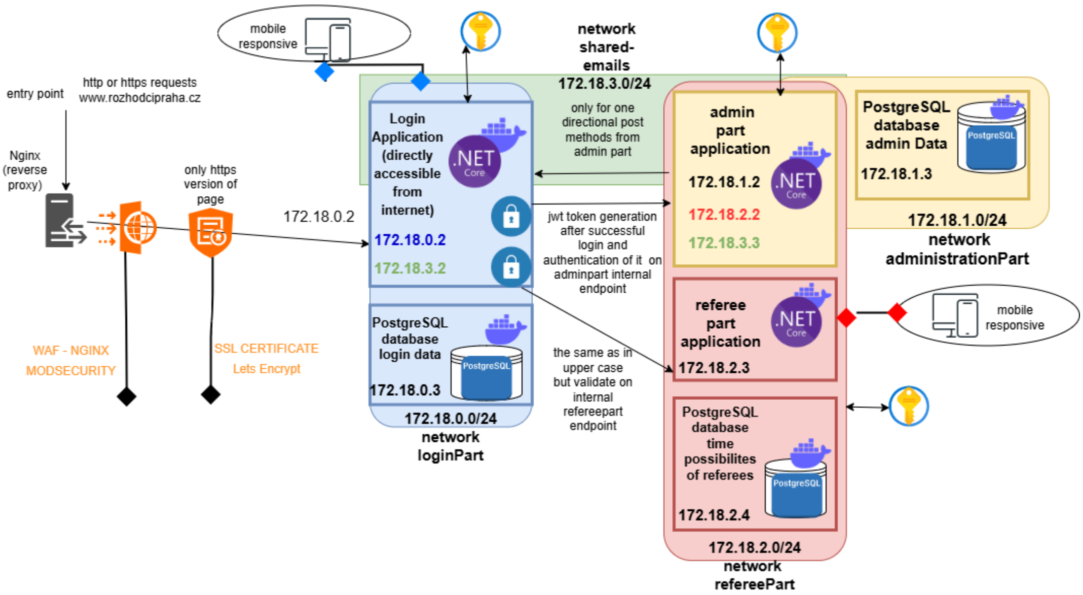

mobile responsive
entry point
http or https requests
www.rozhodcipraha.cz
Nginx (reverse proxy)
only https version of page
172.18.0.2
WAF - NGINX MODSECURITY
SSL CERTIFICATE
Lets Encrypt
Login Application (directly accessible from internet)
.NET Core
172.18.0.2
172.18.3.2
PostgreSQL database login data
PostgreSQL
172.18.0.3
172.18.0.0/24
network loginPart
network shared-emails
172.18.3.0/24
only for one directional post methods from admin part
jwt token generation after successful login and authentication of it on adminpart internal endpoint
the same as in upper case but validate on internal refereepart endpoint
admin part application
172.18.1.2
172.18.2.2
172.18.3.3
PostgreSQL database admin Data
172.18.1.3
172.18.1.0/24
network administrationPart
referee part application
172.18.2.3
mobile responsive
PostgreSQL database time possibilites of referees
172.18.2.4
172.18.2.0/24
network refereePart

Figure 5.4 Description of the deployment on the production server,created in drawio. [32]

5.6.1 Docker

Imagine being able to package an application together with all its dependencies and run it seamlessly across various development, testing, and production environments [33]. This is precisely the goal of the open-source Docker project. Among all containerization solutions, Docker has emerged as the market leader, offering a comprehensive toolset for packaging and delivering software [34]. Additionally, Docker supports secure management of sensitive data such as API keys and passwords through Docker Secrets, a feature that will be used to store my API key for accessing Azure Key Vault [35]. The application will be built using a Dockerfile and deployed using Docker Compose.

5.6.2 Nginx

is a high-performance, open-source web server and reverse proxy designed to handle a large number of simultaneous connections efficiently. Nginx is known for its speed, stability, and low resource requirements. As a reverse proxy, it can provide backend protection by serving as an intermediary that routes client requests to appropriate backend servers and can be configured with various security features.” The core concepts in your description are correct, but the wording could be more precise, particularly around the ”open” terminology and the backend protection capabilities. [36]

Chapter 6

Implementation

This chapter details the implementation process and includes example parts of the source code.

6.1 Project structure

This thesis was developed using Visual Studio Community 2022, which is available under a free license. Source code versioning was primarily managed using the faculty GitLab. However, due to my thesis supervisor from Slovenia not having an account there, GitHub was also used.

The project uses the default directory structure of the .NET Core framework projects with a few minor modifications. For example, additional folders such as Services, Data, and Hubs have been included.

6.1.1 Repository patern

6.1.2 Service patern

6.1.3 SignalR hubs

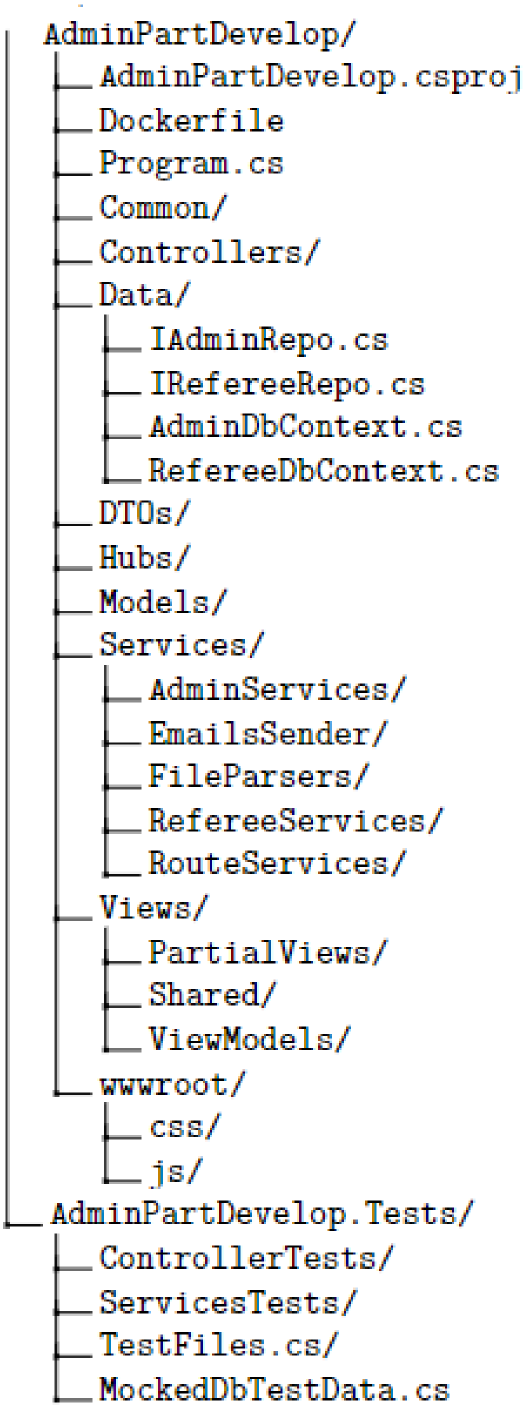
6.2 Dependency Injection

6.3 Data transfer objects(DTO)

6.4 Authentication and authorization

6.5 Nginx

6.6 Building and deploying the application



```
AdminPartDevelop/
├── AdminPartDevelop.csproj
├── Dockerfile
├── Program.cs
├── Common/
├── Controllers/
├── Data/
│   ├── IAdminRepo.cs
│   ├── IRefereeRepo.cs
│   ├── AdminDbContext.cs
│   └── RefereeDbContext.cs
├── DTOs/
├── Hubs/
├── Models/
├── Services/
│   ├── AdminServices/
│   ├── EmailsSender/
│   ├── FileParsers/
│   ├── RefereeServices/
│   └── RouteServices/
├── Views/
│   ├── PartialViews/
│   ├── Shared/
│   └── ViewModels/
└── wwwroot/
    ├── css/
    └── js/
AdminPartDevelop.Tests/
├── ControllerTests/
├── ServicesTests/
├── TestFiles.cs/
└── MockedDbTestData.cs
```

Figure 6.1 Directory structure of the admin part in the develop version

Chapter 7

Testing

Chapter 8
Evaluation

Chapter 9

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Curabitur sagittis hendrerit ante. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Cras pede libero, dapibus nec, pretium sit amet, tempor quis. Sed vel lectus. Donec odio tempus molestie, porttitor ut, iaculis quis, sem. Cras pede libero, dapibus nec, pretium sit amet, tempor quis. Sed vel lectus.

Appendix A
Nějaká příloha

Sem přijde to, co nepatří do hlavní části.

Bibliography

1. PRAŽSKÝ FOTBALOVÝ SVAZ. *O Pražském fotbalovém svazu*. 2025. Available also from: `https://www.fotbalpraha.cz/pfs-text/1-o-prazskem-fotbalovem-svazu`. Accessed: 2025-06-05.
2. MODELER, Bizagi. *Bizagi Modeler for Windows*. 2025. Available also from: `https://bizagi-process-modeler.en.softonic.com/`. Accessed: 2025-06-07.
3. ZHOU, Jun Ying. *Functional requirements and non-functional requirements: a survey*. 2004. PhD thesis. Concordia University.
4. MICROSOFT. *Cache in-memory in ASP.NET Core*. 2024. Available also from: `https://learn.microsoft.com/en-us/aspnet/core/performance/caching/memory?view=aspnetcore-9.0`. Accessed: 2025-06-05.
5. LUCIDCHART. *LucidChart*. 2025. Available also from: `https://www.lucidchart.com/pages`. Accessed: 2025-06-07.
6. REFASSIST. *We simplify your referee management*. 2025. Available also from: `https://refassist.com/en`. Accessed: 2025-06-05.
7. REFASSIST. *How Football Flanders freed up 85%*. 2019. Available also from: `https://refassist.com/en/case/how-football-flanders-freed-up-85-of-its-planning-time-with-automatic-and-manual-planning-tools/3`. Accessed: 2025-06-05.
8. DFBNET. *DFBnet Referee Appointment User Manua*. 2012. Available also from: `https://portal.dfbnet.org/fileadmin/content/downloads/DFBnet_Verein/Manual_Referee_Appointment.pdf`. Accessed: 2025-06-04.
9. PATEL, Het. The Role of Data Visualization in Decision Making: Case of D-mart. 2025.
10. DFBNET. *Freigabemitteilungen für Schiedsrichter*. 2023. Available also from: `https://portal.dfbnet.org/de/service/freigabemitteilungen/schiedsrichter.html`. Accessed: 2025-06-04.

11. ANARSOLUTIONS. *Razor Pages vs. MVC – Which one is better for your project?* 2024. Available also from: `https://anarsolutions.com/razor-pages-vs-mvc-which-one-is-better-for-your-project/`. Accessed: 2025-06-05.

12. JQUERY FOUNDATION. *jQuery - The Write Less, Do More, JavaScript Library.* 2025. Available also from: `https://jquery.com/`. Accessed: 2025-06-07.

13. TRAN, Lac. Javascript vs jQuery and React in web development. 2020.

14. TYAGI, DR POORNIMA. Enhancing Web Application Performance: ASP .NET Core MVC And Azure Solutions. [N.d.].

15. W3SCHOOLS. *jQuery AJAX Methods.* 2024. Available also from: `https://www.w3schools.com/JQuery/jquery_ref_ajax.asp`. Accessed: 2025-06-07.

16. NAYYERI, Keyvan; WHITE, Darren. *Pro ASP. NET SignalR: Real-Time Communication in. NET with SignalR 2.1.* Apress, 2014.

17. WIKIPEDIA. *Model–-view–-controller.* 2025. Available also from: `https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller`. Accessed: 2025-06-05.

18. AWS, Amazon. *What's the Difference Between Relational and Non-relational Databases?* 2024. Available also from: `https://aws.amazon.com/compare/the-difference-between-relational-and-non-relational-databases/`. Accessed: 2025-06-07.

19. STACKOVERFLOW. *2024 Developer survey.* 2024. Available also from: `https://survey.stackoverflow.co/2024/technology#most-popular-technologies-database`. Accessed: 2025-06-07.

20. CHOINA, Marcin; SKUBLEWSKA-PASZKOWSKA, Maria. Performance analysis of relational databases MySQL, PostgreSQL and Oracle using Doctrine libraries. *Journal of Computer Sciences Institute.* 2022, vol. 24, pp. 250–257. Available also from: `https://ph.pollub.pl/index.php/jcsi/article/view/3000`.

21. ROUT, Pranaya. *ASP.NET Core Identity Tables.* Dot Net Tutorials, 2024. Available also from: `https://dotnettutorials.net/lesson/asp-net-core-identity-tables/`. Accessed: 2025-06-07.

22. JETBRAINS. *DataGrip.* 2025. Available also from: `https://www.jetbrains.com/datagrip/`. Accessed: 2025-06-07.

23. SENDGRID. *SendGrid pricing.* 2025. Available also from: `https://sendgrid.com/en-us/pricing`. Accessed: 2025-06-07.

24. MICROSOFT. *Network security for Azure Key Vault.* 2025. Available also from: `https://learn.microsoft.com/en-us/azure/key-vault/general/network-security`. Accessed: 2025-06-07.

25. MAPY.CZ. *API pro Plánování*. 2025. Available also from: `https://developer.mapy.com/cs/rest-api/funkce/planovani/`. Accessed: 2025-06-07.

26. MAPY.CZ. *Cenová kalkulačka*. 2025. Available also from: `https://developer.mapy.com/cs/cena/cenova-kalkulacka/`. Accessed: 2025-06-07.

27. GOOGLE. *Google Maps Platform billing and pricing overview*. 2025. Available also from: `https://developers.google.com/maps/billing-and-pricing/overview#directions`. Accessed: 2025-06-07.

28. HUOTARI, Hermanni. NGINX as an API gateway. 2025.

29. ENCRYPT, Let's. *Encryption for Everybody*. 2025. Available also from: `https://letsencrypt.org/`. Accessed: 2025-06-07.

30. F5. *NGINX ModSecurity WAF*. 2024. Available also from: `https://www.f5.com/go/product/nginx-modules/nginx-waf`. Accessed: 2025-06-07.

31. BUI, Thanh. Analysis of Docker Security. *CoRR*. 2015, vol. abs/1501.02967. Available from arXiv: `1501.02967`.

32. DRAWIO. *Drawio*. 2025. Available also from: `https://www.drawio.com/`. Accessed: 2025-06-07.

33. MERKEL, Dirk et al. Docker: lightweight linux containers for consistent development and deployment. *Linux j.* 2014, vol. 239, no. 2, p. 2.

34. COMBE, Theo; MARTIN, Antony; DI PIETRO, Roberto. To Docker or Not to Docker: A Security Perspective. *IEEE Cloud Computing*. 2016, vol. 3, no. 5, pp. 54–62. Available from DOI: `10.1109/MCC.2016.100`.

35. THIYAGARAJAN, Gogulakrishnan; NAYAK, Prabhudarshi. Docker under Siege: Securing Containers in the Modern Era. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2025, vol. 11, pp. 3674–3719. Available from DOI: `10.32628/CSEIT25112773`.

36. WEBGLOBE. *What is nginx*. 2025. Available also from: `https://www.webglobe.cz/poradna/co-je-nginx`. Accessed: 2025-06-07.

Obsah příloh

```
/
├── readme.txt .............................. stručný popis obsahu média
├── exe ..................... adresář se spustitelnou formou implementace
├── src
│   ├── impl ................................ zdrojové kódy implementace
│   └── thesis ................... zdrojová forma práce ve formátu LaTeX
└── text .................................................... text práce
    └── thesis.pdf ......................... text práce ve formátu PDF
```