FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF BACHELOR’S THESIS

Title:	Reservation System for Football Matches
Student:	Nikolai Kosariev
Supervisor:	Ing. Petra Pavlíčková, Ph.D.
Study Programme:	Informatics
Study Branch:	Information Systems and Management
Department:	Department of Software Engineering
Validity:	Until the end of summer semester 2018/19

Instructions

1) Analyze current reservation systems for football games and their advantages and disadvantages.
2) Based on the results of the analysis, choose the best option and use it for implementation.
3) On this base, design reservation system.
4) Implement the prototype which will allow the user of the web application to register on the site, create events or take part in existing ones.
5) Create the test case scenarios and test the prototype.
6) Implement, evaluate and suggest additional recommendations / developments.
7) Describe costs (of implementation and operation) and benefits of the reservation system.

References

Will be provided by the supervisor.

Ing. Michal Valenta, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague January 10, 2018

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Reservation System for Football Matches

Nikolai Kosariev

Department of Software Engineering
Supervisor: Ing. Petra Pavlíčková Phd.

May 14, 2018

Acknowledgements

I would like to thank the supervisor of the work, Ing. Petra Pavlíčková Phd. for help in writing of this thesis, valuable advice and suggestions for improvement. Also I would like to thank my whole family and all the people who are close to 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 that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on May 14, 2018 . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Kosariev, Nikolai. *Reservation System for Football Matches.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2018.

Abstrakt

Tato bakalářská práce je zaměřena na řešení problému organizace fotbalových utkání. Cílem práce je zanalyzovat současná řešení této problematiky, navrhnout jejich vylepšení, prozkoumat požadavky uživatelů a na základě předchozích šetření navrhnout nejoptimálnější řešení a následně implementovat a otestovat prototyp rezervačního systému. Samotný rezervační systém bude realizován jako webová aplikace postavena na Java frameworku Spring. Prototyp aplikace bude jakožto hlavní výstup této práce podporovat autentizaci jednotlivých uživatelů, vytvoření nových utkání, vyhledávání již existujících utkání a jejich zúčastnění.

Klíčová slova rezervační systém, koncept, fotbalové utkání pro amatery, management fotbalových zápasů, web, Java, Spring Framework

Abstract

This thesis is focusing on solving the problem of organizing football matches. The aim of the work is to analyze existing solutions of this problem, propose improvements, analyze user needs and, on the basis of previous studies, design the most optimal solution and subsequently implement and test the prototype of reservation system. This booking system is implemented using Java framework Spring. As the main result of this thesis, the prototype of the application will support the authentication of individual users, the creation of new matches, the search for existing ones and further participation in them.

Keywords Reservation system, concept, football matches for amateurs, management of football matches, website, Java, Spring Framework

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Football is the number one sport on the planet according to its popularity. FIFA members comprise of 208 national football federations, this is more than the number of countries that make up the United Nations. In addition to such a global official distribution, in almost every city on local football grounds amateurs play football almost every day, organizing matches and tournaments among themselves.

The main component of the football game is a team and not an individual player. The key process that determines whether a match happens or not is the organization of a match. For the match to take place, it is necessary to set a place, time and determine the level of the participants. It is not an easy task to organize a game without making a prior agreement between all of the participants. With the advent of the Internet, of course, all processes were simplified. However, based on my personal experience, I found out that today the process of organization occurs with the help of social networks or with the help of other means of online communication, which is not always convenient and, therefore, not the most optimal solution for the organization process.

Therefore, in the practical part of my work, I decided to focus on creating a reservation system that would simplify the process of organizing matches for all users participating in this process.

In the theoretical part of the work, I will focus in more detail on the problem of organization of football matches. Next, I will analyze all existing applications. Also, based on the analysis, the most optimal solution will be offered and a prototype reservation system will be created on its basis. SWOT analysis will also be carried out and the costs of creating and operating this reservation system will be calculated.

Chapter 2

The aim and motivation of the thesis

2.1 Aim

The first goal of this thesis is to describe the process of organizing football matches and formulate the problems that arise during this process.

The second objective is to analyze existing applications that are trying to solve the problem of organizing football matches and understand whether their functionality is able to fulfill all the requirements for organizing matches. Following the analysis and considering the requirements of potential users, will be suggested the most optimal solution for this problem.

The third goal, is to design of the reservation system and the implement and test its prototype based on the proposed solution. The prototype will allow the user to register, search for existing football matches and join them. Also, the user will be able to create new matches according to the specified criteria.

2.2 Motivation

The motivation behind choosing this topic was my personal experience of participating in such football matches for amateurs. Due to the lack of necessary full-fledged applications dealing with the organization of such matches, often the process of communication and finding the necessary participants gets very complicated, which leads to the fact that the scheduled matches are not organized and get cancelled. Also the benefit of the bachelor thesis will be for the author to extend the knowledge of web development tools such as Spring Framework and to practise building complex systems using them.

Chapter 3

The review part

3.1 Description of problem

A football match among amateurs is usually organised in two different ways:

1. If there is a group of players who are interested in participating in a football match.

2. If there is an individual player who wants to join an existing game but does not want to solve the issues related to the organization of the match.

In the first case there are 2 options for holding a football game among amateurs one-time play or holding several games for the tournament. In both cases, the game organizer needs to find the required number of players with matching skill level, arrange the date and time that will suit all the participants and afterwards find and reserve a local football field. If the field is private, an organiser needs to enquire about the price and available time slots and in case of availability of the field reserve it. Afterwards, an organiser needs to send the confirmation to all participants of the game about the date, time and location of the game as well as an additional cost necessary to cover the rent of the field if it is private. In both cases, it is often complicated to organize a match at a suitable place and time and notify everyone of the upcoming event. Another difficult task is to search for missing players for a particular match, while taking into account the skill level of already participating players.

We can also consider this situation from the point of view of an individual player who wants to join an existing game but does not want to solve the issues related to the organization of the match. In the case of an individual player, the problem may be searching for an already existing match (because such kind of matches are organized spontaneously and therefore are hard to find) or finding a match of players with a certain skill level.

3.2 The current state of the solution to this problem

Having searched and analyzed similar applications that attempt to solve the problem with the organization of football matches or tournaments, I found out the following information.

Each application has the most basic functions such as registration and authorization. Certainly, all applications have function of creating a team or a club and they have very rich and attractive functions for editing a profile of a team or club (such as adding a logo, a club location, motto or history). Also, almost every application has the above mentioned tools to create or edit the profile of each player from that team or club. Most applications have a functionality of organizing a team meeting for training or another event, and naturally you can specify meeting parameters such as location, event importance, date and time. In addition, there is a function of monitoring the attendance of an upcoming event: participants who will definitely come, will probably participate or can not attend this event. Some applications have additional functionalities such as communication on the site between the participants and the organizer as well as notifications and reminders about the upcoming event.

Therefore, all of the applications represent a good tool for existing teams to organise training sessions. However, they lack functionality for reserving the field for a training session or game, which can lead to clashes if another team decides to train at the same place and time.

All the aforementioned functionalities are very limited in a sense that they do not allow user to organise a live match with real players. Having such functionality would allow not only to create two competing teams, but also to hold a real game, save the result and collect statistics. Absence of such functionality in existing applications makes their use-case very limited and in some instances becomes a good reason for a user to stop using the application. Activity and further development of football teams after they have registered and arranged a couple of training sessions are not possible within existing application platforms. Functionality of existing applications does not allow users to do other useful actions including organisation of the live competition, where teams could apply their skills gained through their training sessions.

3.3 Existing solutions

Below, we will have a look at existing web applications, which deal with organising meetings for teams or like-minded people in a specific area. Each of those web applications has functionalities that were described in the previous section, therefore to avoid repetition I will not describe them again and will only point out the unique features that are present in the web application. Also web applications will be looked at from the point of view of design, structure, navigation, response time and other factors.

3.3.1 Conventional solutions

Conventional solutions described as part of this thesis are ways of organising football matches without using special web applications. Conventional process of organising a football match without using any web application takes place through the communication in social networks. One of the organisers or participants creates a common chat or group specifying all the information about upcoming event and then following a chat a common agreement is reached about conditions of the game. The most popular service without any doubt is Facebook, which is the largest social network in the world. There are two options to approach the problem of field reservation: one can contact a field manager directly or use a special online booking service, which is available in some cities. An example of such service is a czech web application [2] , which helps finding private sports fields for different kinds of sports in big cities in Czech Republic and book them. However, this web application only allows finding and booking the fields without additional features of gathering players into teams for a game.

3.3.2 Teamapp.com

Team App is a platform for sport teams and various social groups, which facilitates communication between all team members and which has synchronization with its mobile version of the application.

The web application has a simple design and an intuitive navigation around website, that speaks of a well-created UX Design. The main page is not overloaded with information, there is no advertising which results in a good performance and a normal response time. In addition to a functionality to create teams, there is a functionality to search for existing teams, view detailed information about each player, multimedia files and schedule of upcoming events of this team. Applications supports a little more than 100 languages. Mobile version of the application is available on both iOS and Android platforms.

3.3.3 Tymuj.cz

This is one of the most commonly used portals in the Czech Republic. It is used for organizing sports activities in different kinds of sports (it is possible to specify the type of sport for the team or plan physical activity and leisure). Based on the stats for January 2017 available on the site [3], the portal has 150 000 users. The website has a nice minimal design, intuitive navigation and in general a good UX Design. A small amount of advertising slightly reduces the response time. Portal only supports only Czech and English languages. Mobile apps are available on iOS and Android platforms.

3.3.4 Spoluhraci.cz

The web application positions itself as a communication server, which was created for communicating and planning a variety of events for any groups, clubs and organizations.

In addition to communication within his own team, a user has the ability to search for members with similar interests in existing communities. The web application has a very cluttered home page and a complicated navigation structure which poses as a problem for a user unfamiliar with this service and complicates the use of this application. There is a lot of advertising on the website. The design of the site is also not ideal, as colors and fonts are not quite appropriate and impact the content readability in a negative way. Also the application is available only in Czech and does not support mobile devices.

3.3.5 Meetup.com

This is a web application which represents a huge social platform that helps to organize or find an already existing event, meeting or seminar. The areas of interest that are available for searching and creating of the events are very diverse and include meetings for moms, dog walking, business seminars, city tours or museums and finally a joint fitness visit or the creation of a football match.

The site has a minimal design, a fairly simple navigation structure, almost no advertising and fast web page load time. It is available in several languages and has mobile applications on android and iOS platforms. You can search the existing events for free based on different criteria and join them. But if you want to be the organizer of an event, you need to purchase a paid account.

This web application solves the problem of finding existing sports events for further participation in them, however the event creation format in this web application is very generic and universal in order to create events of different kinds and therefore has insufficient functionality to organize football matches. Theoretically, you can gather people at the appointed time, but you can not solve the issue with the reservation of the football field and allocation of players to the teams.

3.3.6 Teamstuff.com

Web application for organizing events and managing existing teams. It was invented in Australia by a group of sports amateurs. It is translated into 14 languages. Among its distinctive features, it is worth noting that the application allows you to register your family in addition to registering a team or club and further plan and create a schedule of sports and any other activities for your family and manage the schedules of all members together, as the creators of the application believe that the family is an integral part of life each person.

It is worth noting that among all the listed applications, only this one has an online payment functionality, which allows you to pay for example judicial, seasonal fees, equipment, transportation, insurance and other necessary payments. The function is implemented with the help of a payment partner Stripe [4]. The application has a minimalistic design, well-chosen colors and fonts, which makes the process of using this resource very convenient. The navigation bar is conveniently located on the page. Response time is also normal in speed, the only censure is a very long recording process. Mobile apps are available on iOS and Android platforms.

3.3.7 Sportmap.kz

I managed to find one web application that has the necessary features to solve the described problem. This is a kazakh site. It can be described more as a large database of all sports facilities and sports grounds available in Kazakhstan. For example, there you can find a fitness club based on the search criteria (city, region, price) and learn more about this place, which additional services and infrastructure this institution offers. The application has a simple design and structure.

Also, unlike other applications, this site allows you to add not only a team or a club, but a playground for any sport, so that later it can be found and used for sports meetings. If you are the owner of a sports establishment or a playground, you can add this institution to a common database, so that later, customers or interested parties could find this institution with all the necessary information.

In addition, it is possible to organize a football match on the football fields available on the website, where it is possible to specify the number of players, to add participants or invite them, indicating the entrance fee for each participant. Afterwards user can contact the the owner of the field using the contact details available on the site and resolve the issues with the reservation.

The application is available only in Russian and does not a have mobile application . While using the application, I encountered several mistakes and failures, in particular when creating a football match. As a result, I received 404 page error, which indicates that the web application has not passed user acceptance testing [5]. Overall, I can say that this application performs those functions and tasks for organizing football matches that have been described in the problem description section. The application allows you to find the football match based on the location and price preferences and join it or create your own match and invite other players to join it. However, the issue with the field reservation will need to be resolved separately by the organizer by contacting the owner of the field, because the site does not provide such functionality.

It is worth mentioning also that when registering on the site 3 types of accounts are available: free “start”, which allows you to make and change basic information about the institution and maintaining statistics, the second paid “standard” that allows you to add to the basic information a hyperlink to the official site of the institution, a section with reviews is also available,

And the third is the more expensive “super” plan. Where, in addition to all the listed functions, you get help with the promotion of your institution, creating content and you institution will always appear at the top of the search. From this it is possible to conclude that this application has elements of an advertising and CMS platforms [6].

3.4 Chosen solution

In a web application that I’m going to develop, I decided that the functions that were described in the section 3.3, such as creating a separate club or team and adding related information about them (description and characteristics of individual team members, slogans, history, multimedia files and etc.), will not be implemented in this application, since its main goal is not to store the team as a static element with information, but primarily to organize sports activities, related to participation and competition in football matches or tournaments with the participation of this team. So the teams in my application will exist neither as an element that is conditionally registered and filled with related information, but as a dynamic element that is part of the process – such as a game or tournament, where this team will be a member and someone else’s opponent.

In this thesis a football match or just a match is an event in which two teams of football players meet on a specific football field and compete with each other. A tournament is a series of matches held between several teams that compete for a prize or a title.

Basic features The presence of such functions as new user registration and authorization are basic minimum functions. There will be a function for editing the profile of an individual user and the profile of the team. Communication between players of one team will also be available in the form of a group conversation. A user will be able to view statistics on the success and effectiveness of his team based on the results of matches played in order to track the success of the development of this team during a specific period of time. Also, will be available the rating of the team. The rating of the team is defined on the basis of the responses of the members of the opposing team. After the game, the participant of this match can authorize and rate a player of the opposing team.

Creating a team After authorizing, a user can create his own team by registering his team players for further participation in games or tournaments. Is it usually the coach of the team, who is dealing with the task of creating and registering an existing semi professional team playing for a specific club. In case of unofficial amateur team, the team captain can register it and in both cases the team creator gets special privileges to challenge another team to compete or to accept the challenge from another team.

Registering a football field Also, the manager of any private football field will be able to register his field in the application and make it available for bookings (having indicated and set up date and time available for bookings) and will have his user account, where he can view and manage the reservations. Registered participants will be able to register public football fields by indicating the necessary location information and making this field available for selection when creating a football match.

A match between amateur players The first category is the creation of a one-time football match for individual players, where you can invite any participants (they will be able to participate there after registration), or any user of the web application can find this match and after the authorization to participate in it. When creating a match, the number of players in the team has to be indicated, and if not all the places in both teams are filled with the right number of players, then the match is considered to be incomplete and remains available in the search with the status "available for participation" and authorized users can participate in it and, depending on the availability of places, select the team which they want to join. When the right number of participants for both teams is registered, the match will be considered to be organised and all participants will receive an email with all the necessary information about the upcoming match. After the match ends, any of the participants in this match can publish the results of this match in the application.

A match between the teams The second category of matches is matches between formed teams registered in the application. If the user is the organizer of the team, he can send an invitation to play the match to another registered team and in turn if the organizer of the team that received the offer for the game accepts the invitation then this match will be held.

A tournament There will also be a function for creating tournaments. Any team organizer will be able to create a tournament and invite 2^n (this is a necessary condition for the draw between the teams) and if the required number of teams have agreed, then the application itself will make a draw by distributing participants and their opponents in the standings. For the purpose of searching a free time for organizing a match the will be a function that allows a user to view all scheduled matches for a specific field on a specified date.

3.5 Development of the information system

The purpose of this thesis is to create a reservation system, which represents an information system. An information system is a system for collecting, storing, analyzing data and that provides information to different users.

To date, the main requirement for the information system is to function as a web application. This is related to the easier and more customary management as well as the lack of hassle associated with the installation of the application, synchronization of information, etc. In this thesis, I will use the terms web application and application interchangeably, since by definition a web application is considered to be a type of application [7].

Before proceeding with the implementation of this reservation system I will analyze existing applications with similar functionality and will determine to which extent they are able to solve the described problem. Based on the analysis, I will define the necessary functions for creating the necessary reservation system. Based on this, the functional requirements will be defined. After the requirements, I will create a domain model and wil design use cases, choose the technologies for implementing this web application and describe them.

Finally, I will begin to implement the prototype application. During the implementation of the application, I will be guided by the principles of object-oriented programming. At the end the prototype will be tested.

3.6 Chapter summary

This chapter described the process of organizing football matches and the problems regarding their organization. Most of the existing applications for the organization of these events have been described. An analysis was made of how those applications can solve the organization's problems and it was found out that to date there is no full-fledged web application that could solve the described problem of the organization. Based on the analysis, a version of the reservation system was proposed that would be able to solve the entire process of organizing football matches. The implementation of this reservation system will be implemented as a web application.

Chapter 4

Analysis and design

4.1 Functional requirements

In this section, I will review the functional requirements. Access to the functions of creating a new match, team or football field and joining the match will be available only after registration and authorization. An unauthorized user will only be able to view previously listed items.

- *Evidence of users*
  - The application should be able to persistently store users and information about them.
- *Evidence of football grounds*
  - The application must be able to persistently store football fields and information about them.
- *Evidence of team*
  - The application must be able to persistently store teams and information about them.
- *Evidence of match*
  - The application must be able to persistently store matches and information about them.
- *Search for existing one-time matches*
  - The application should allow users to search for existing football using the given parameters (location, skill level, date and time).
- *Participation in existing one-time matches*

  - The application should allow users to join a given football match depending on availability of free spaces.

- *Registration*
  - To use all the functions of the web application, the user must register. When registering, the user is required to provide mandatory data - username and password.

- *Editing user profile*
  - An authorized user can update his personal data in the profile editing section. Also, the user can add data such as name, surname, date of birth, city, professional level and his number in the team (the data is necessary if the user is interested in creating a team to participate in matches/tournaments).

- *Authorization*
  - To access most of the functions of the web application after registration, the user must be authorized.

- *Registration and managing a private sports ground*
  - After registration and authorization the owner of a private sport ground will be able to create a new location for football matches by indicating all the necessary information about his sports ground, and also specifying the number of the ownership certificate for the sports ground.

    After that the web application will check if the number of this certificate belongs to this person and in case of compliance a new location will be created and the owner of this field will have access to administer part this field (for example set the price, the number of players and the available time for using this field).

- *Registering a public football field*
  - Each authorized user in order to create a new match can register a new public football field if it is not already registered in the web application.

- *Tournament creation*
  - An authorized user, who is the creator of any team, can create a tournament by inviting 2^n teams registered in the application (including his own team) and specifying the appropriate football fields registered in the application. Further, if the required number (multiple of 2^n) of the teams confirms their participation in this

tournament, then the application will distribute the teams among themselves in the required order for the tournament.

- *Reservations Management*
  - An authorized user can manage their reservations. He will be able to view a list of all matches where he will participate and if necessary he will be able to cancel his participation in the match during the specified cancellation time.
- *Creation and management regular team*
  - The coach or captain of the team will be able to create a team and send out invitations to the participants of this team so that they register in the web application and confirm that they are players of this team. The creator of this team will have access to the administrator part to manage this team. And he can, if necessary, change participants, edit information about them (as time goes by, update the professional level of a particular participant) and will be able to assign additional team meetings for training or other events for the whole team.

    The creator of the team will be able to accept invitations from other teams to compete with them or invite other teams for a match.
- *Search for a match nearby*
  - An authorized user will be able to search and view all football matches available within a given radius for a subsequent participation in them.
- *Communication between players of one team in the form of a general conversation/chat*
  - Participants of the permanent registered team together with the team's creator (coach or captain of this team) will be able to communicate and negotiate in a general conversation/chat.
- *Statistics and rating*
  - For each registered team there will be a function for viewing the statistics and rating that will be formed based on the results of previously played matches and the ratings of other teams. Each player which participated in a match against any permanent team, after the end of the match will be able to rate this team.
- *Creating a new football match*

  - The application should allow users to create a new football match and indicate a football field, skill level, date and time. The first possibility when creating a match is the creation of a one-time game in which different single users who are not participants of any permanent registered team will participate. This type of match is intended for players who do not have a permanent team and they only want to play a one-time football game with similar players. The match will be gradually filled with authorized users of this application who registered in this web application in order to search for a football match and subsequently participation in it.
  - The creator of the match will indicate the number of seats for this match and accordingly there will be 2 teams with the same capacity of the players. It will be possible to view the existing members of each team and choose one of the two teams and become a member. At the beginning of the match, the match will have the status of newly-formed and when the match gets the required number of participants in both teams, the match will get the status of the formed and all participants will be notified that the match will be held. For the organization of the match there will be the opportunity to invite further participants who after the registration will be able to participate in this match.

- *A match between 2 permanent teams registered in the application.*
  - The second possibility when creating a match will be available as follows: the creator of the permanent team can create a new match by indicating all the necessary details for the match and invite the other team registered in this web application for a match.

4.2 Non-functional requirements

4.2.1 Availability

Expected system availability is 24x7 except cases of unforseen downtimes or irregular updates.

4.2.2 Safety

Application should process main authentication principles, maintain data consistency and user rights. Application should use technology, which will be used for encoding important data in the interest of user safety.

4.2.3 Reliability

Data entered by the user must retain its structure in the event of a power outage, and the application must be ready to eliminate collisions in case of incorrect user requests. In order to avoid these problems, the application should use the principle of transactions to eliminate logical data inconsistencies.

4.2.4 The following non-functional requirements are

- Web application
- Support in all major browsers
- Responsive design for display on different types of devices
- Framework Spring
- Database PostgreSQL

4.3 Organization of users

The following roles will be available in the application. This list of roles is ascending in a way that, each role, in addition to its rights, has the rights of the previous role.

Guest An anonymous website visitor, can view parts of the website with an open access (including the list of current matches), can register and authorise.

Authorised user Can create a new match and take part in an existing match.

Organizer This is a registered user who is the coach or captain of the team and after creating a team he will have the possibility to send a request to another team to compete with his team or to accept such a request, he will also be able to create a new tournament inviting teams there.

Administrator Administrator in this web application is usually the owner of a private football field or a proxy who has the authority to use this football field.

4.4 Software architecture

Software architecture is quite a broad concept and currently many definitions exist. Most of them are based on the principle of describing components and their interaction. Here is one of the definitions: *“Architecture is the fundamental organization of a system embodied in its components, their relationships to* *each other, and to the environment, and the principles guiding its design and evolution."* [8]

In my information system, I will use the typical Model-View-Controller (MVC) architecture. Model provides data and responds to controller commands, changing its state, View is responsible for displaying model data to the user, responding to model changes, Controller interprets actions user, notifying the model of the need for changes.

4.5 Design requirements

Design requirements are defined by the fact that the design and content of the web-application are the main components that a target user interacts with. The website should have a user-friendly interface. Perception of the information and subsequent evaluation of the web-application, which influences a user's decision to use this product is dependent on the quality and level of the web-application design. The visitor's decision cannot be directly influenced, only the quality design of the application can convince him. The design of a well developed web application should be defined by the following three points [9]:

- Adaptive
- Elementary
- Organized

4.5.1 Adaptive

Is a type of a web page design that ensures a correct display of the site on various devices. It dynamically adapts to the specified browser window size of the device and if necessary, the text size and width of the web page will vary according to the display size of the device. Also occurs optimization or complete elimination of the animation. First of all, it concerns mobile devices.

4.5.2 Elementary

When developing a web application, it is required to implement a simple design that will be understandable to any user. This criterion should satisfy the following points:

- Intuitive animations
- Using the minimum number of colors (preferably a maximum of 4)
- Do not have irrelevant items
- Have clearly defined boundaries of all elements

The most common mistake that occurs in web applications is the use of a rich color palette. Also often there are problems such as fuzzy boundaries of elements. And another very common problem is the overloading of a web page with a lot of animation. All of those factors combined together complicate the perception of information by the user and make it difficult for him to interact with the application. These design errors were present in some applications, which I described in the review part in existing solution. Design must be as minimal as possible. *"The minimalist philosophy centres on the idea that you must design around the content."* [10]

4.5.3 Organized

This property implies that the user can analyze the application at a glance and use it further without problems. To achieve this result, it is necessary to implement such rules as selecting one style and size of the text and using it throughout the application. One should also take into account the text of the headings, so that a user can easily distinguish between a title, a subtitle or plain text. Next, the text itself should be structured into paragraphs that must be short and accurate, conveying the basic meaning. And the last, but not the least important point is the recognition of links. Each link or button should be easily recognizable without first hovering the mouse over them.

4.6 Use cases

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous [11]. Use case diagram is located in the Appendix D.

UC1 - Participation in existing one-time match

- Brief Description
  - This use case describes the steps that will allow the user to participate in an existing one-time match.
- Actors
  - User

    - Web application

- Goals

    - This process will allow the user to participate in the selected existing one-off match.

- Preconditions

    1. The user must be authorized in the application.
    2. There must be a necessary match.
    3. In this match there is at least one free seat.
    4. User successfully performed actions from UC2.

- Post Conditions

    - The user will become a member of this match and he will be displayed on the reservations of this user and in case the match is formed the user will receive a notification by email.

- Basic Flow

    1. The user will perform the same actions as in UC2.
    2. Next, the user will choose the necessary match with at least one free place in the team which he wants to join.
    3. The user presses the "participate" button under the desired team.
    4. The application will check if the user has not already scheduled other matches for the selected time and, if successful, will add him to the list of participants in the match.

- Alternate Flows

    - There are no alternative scenarios.

- Exception Flows

    1. If the user wants to join the match that overlaps in time with a different match that has already been scheduled.
        - An error message will be shown to the user.

- If the user tries to join the team, which he has already join or if participating in on team he will try to join the opposing team, thereby participating in both teams.

  - An error message will be shown to the user.

UC2 - Searching for the existing one-time matches

- Brief Description
  - Use case describes the necessary steps to find an existing one-off match by the user. This use case will be used by other use cases to implement different functionality.
- Actors
  - User
  - Web application
- Goals
  - The purpose of this use case is to describe the process of searching for existing one-off matches by the user.
- Preconditions
  - There are no preconditions for this use case.
- Post Conditions
  - The application will display to the user all existing one-off football matches.
- Basic Flow
  1. The user clicks on the menu item “Search matches”.
  2. The application will ask a user to choose a city.
  3. Next, the user selects the city and the application displays all available football fields in the city.
  4. The user will choose the football field.
  5. Then user chooses a date
  6. The application will display all the actual existing matches held on this field for the selected date.
  7. The user will be able to view the necessary information about each match.
- Alternate Flows
  - Within this use case, there are no alternative scenarios.

- Exception Flows
  - This use case does not have exceptions.

UC3 - Creating new one-off match

- Brief Description
  - This use case describes the steps necessary to create a new one-off match.
- Actors
  - User
  - Web application
- Goals
  - In the application a new match will be created according to the set criteria and the match will be available for participation to other participants.
- Preconditions
  1. The user must be authorized in the application. For the given time there is no other match being held on the chosen football field.
  2. User successfully performed actions from UC2.
- Post Conditions
  - All data will be saved. The application will create a new match and it will be available for participation to other participants. The user who created this match automatically becomes a member of this match.
- Basic Flow
  1. The user will perform the same actions as in UC2.
  2. If user is not able to find a match based on the search criteria then he will be able to create his own.
  3. The user clicks on the “create new match”.
  4. The app will generate a form, which pre-populates certain fields based on the previously user-selected criteria (city, date and a football field).

5. Next, the user selects the number of seats for each team and the skill level of the players involved in this match, time of the match and, if necessary, if it is a private football field indicates the entrance fee for each participant.

- Alternate Flows
  - If at the beginning of the scenario, when a city or a football field is selected, the user does not find his city or his field in the list, he can click on the "add playground" button and add the required field. Action required for this are described in UC4.
- Exception Flows
  - This use case does not have exceptions.

UC4 - Adding new public playground

- Brief Description
  - This use case describes the steps for adding a new public football field by the user.
- Actors
  - User
  - Web application
- Goals
  - The user will add a new public football field not previously registered in the application for use in other football matches.
- Preconditions
  1. The user must be authorized in the application.
  2. Exactly same field does not exist in the application.
- Post Conditions
  - A new public sports field will be created and all data about it will be saved and it will be available for holding matches on it.
- Basic Flow
  1. The user clicks on the "add new playground".

2. The application will generate a new form.
3. Next, the user will need to enter city, address and name of the field and select the field coverage.
4. Next, the application will check the correctness of the entered data and if it passes validation will save the field making it available for selection when creating a match.

- Alternate Flows
  - This use case does not have alternative flows.
- Exception Flows
  1. User enters an existing address or name of the football field (which matches an existing one).
     - An error message will be shown to the user.

UC5 - Creating new permanent team

- Brief Description
  - This use case describes the process of creating a permanent team.
- Actors
  - User
  - Web application
- Goals
  - The goal of this process is to create a new permanent team for the user.
- Preconditions
  1. The user must be authorized in the application.
  2. This team has not been created earlier.
- Post Conditions
  - The application will create a new permanent team saving all the necessary data.
  - The user who created this team will receive special privileges to manage the team created by him (he can answer and accept invitations for a game with other teams or participate in the tournament).

   - Also, the user will have a separate administrator access where he can change information about the team and manage invitations for the competition.

- Basic Flow

   1. The user selects the menu item “add new team”.
   2. The application will generate a form to create a team.
   3. The user will enter the city of the team, its name, the number of participants and the professional level of the team.
   4. The application will check the correctness of the entered data and, if successful, create and save a new empty team.
   5. Next, the user will be asked to send invitations to the participants of this team to register in the application and confirm participation in this team as a regular player.

- Alternate Flows

   - This use case does not have alternative flows.

- Exception Flows

   1. When creating a team user enters information similar to an existing one.
      - An error message will be shown to the user.

UC6 - Registration

- Brief Description

   - Use case describes a new user registration process in the web-application.

- Actors

   - User
   - Web application

- Goals

   - The goal of the use case is to provide an opportunity to a new user to register in the application in order to subsequently use all of its core functions.

- Preconditions

  - There are no preconditions for this use case.

- Post Conditions
  - All the information entered in the web application will be persistently stored, a new user account will be created, a user will be automatically authorised and redirected to a home page.

- Basic Flow
  1. On the initial screen a user clicks on Register button.
  2. Application will display a registration form.
  3. A User enters user name, password, password confirmation and clicks on Submit button.
  4. Application controls if the entered data is valid.
  5. Application will create a new user account.
  6. Application will automatically authorise the user and will redirect him to a home page.

- Alternate Flows
  - There are no alternate flows for this use case.

- Exception Flows
  1. User enter a user name, which is too short or which already exists in the system.
     - An error message will be shown to the user.
  2. User enter a password, which is too short or enters an invalid value in the password confirmation field or enter a wrong password confirmation.
     - An error message will be shown to the user.

UC7 - Editing user profile

- Brief Description
  - In this use case, the process of changing user information is described.

- Actors
  - User
  - Web application

- Goals
  - The purpose of this process is to allow the user to change already existing information or add new necessary data that has not been provided during registration (name, last name, date of birth and current city).
- Preconditions
  - The user must be authorized in the application.
- Post Conditions
  - The newly added information updates the old data or is added to the database.
- Basic Flow
  1. The user clicks the “edit my profile”.
  2. The user will enter new or change the current data and press the “change” button.
  3. The application will check the correctness of the entered data and, if successful, update the information in the user profile.
- Alternate Flows
  - Within this use case, there are no alternative scenarios.
- Exception Flows
  1. The user will enter incorrect information in any of the given fields.
     - An error message will be shown to the user.

UC8 - Searching for a match nearby

- Brief Description
  - This use case will describe the process of finding matches near a user.
- Actors
  - User
  - Web application
- Goals

  - The user will see all available football matches nearby.

- Preconditions

  1. The user must be authorized in the application.
  2. The user profile must have his city specified or the device must have a gps enabled.

- Post Conditions

  - The user will be shown a list of all available matches nearby.

- Basic Flow

  1. The user clicks the "search match nearby".
  2. The application will check if the city is available for a search an in case of success, will search for matches in the given city.
  3. In case of success, the application will show the user a list of matches in his city.

- Alternate Flows

  - Within this use case, there are no alternative scenarios.

- Exception Flows

  - Within this use case there are no exceptions.

UC9 - Team management

- Brief Description

  - This use case describes the process of managing a permanent team.

- Actors

  - User
  - Web application

- Goals

  - The team's creator will manage invitation to play against other teams and will be able to send such invitations to other teams.

- Preconditions

   1. The user must be authorized in the application.
   2. The user must be the creator of this team.

- Post Conditions
   - In the administrator part of the application, where all invitations are displayed, the status of the invitation will be changed once user responds to it.
   - If the creator of the team accepts the invitation, all participants of his team will receive a notification that they will participate in this match.

- Basic Flow
   1. The user will enter the administrator part responsible for managing invitations from other teams.
   2. The user will choose a specific invitation for the match and will either accept it thereby agreeing to take part in the match / or will refuse to participate.
   3. He can perform the same actions with regards to participation in the tournament.

- Alternate Flows
   - Within this use case, there are no alternative scenarios.

- Exception Flows
   - Within this use case there are no exceptions.

UC10 - Creation new match between permanent teams

- Brief Description
   - This use case describes the process of creating a match between two permanent teams.

- Actors
   - User
   - Web application

- Goals
   - The goal is to create a new match between 2 permanent teams.

- Preconditions

  1. The user must be authorized in the application.
  2. The user must be the creator of the team.
  3. In a city where the user wants to create a match there must be at least one other regular team.

- Post Conditions

  - The application will create a new match between the permanent teams and all the necessary data will be saved.
  - The creator of opposing team will receive a notification that his team was invited to participate in this match.
  - Also, all participants of the team that invited for a challenge the other team will be notified about the upcoming match.

- Basic Flow

  1. The user presses the "create match" button and then selects the option "for permanent teams".
  2. The application will generate a form where the user can select a field in the city where his team is located, then he will select the opposing team from the available ones (in this list there will be teams from the same city and with equivalent number of players).
  3. Next, the user selects the date and time and presses the "create" button. The application will create a new match between the selected teams and will notify the creator of the opposing team.

- Alternate Flows

  - Within this use case, there are no alternative scenarios.

- Exception Flows

  - Within this use case there are no exceptions.

UC11 - Tournament creation

- Brief Description

  - Use case will describe the creation of the tournament.

- Actors

  - User
  - Web application

- Goals

  - The purpose of this function is to allow the user to create a tournament.

- Preconditions

  1. The user must be authorized in the application.
  2. The user must be the creator of any team.

- Post Conditions

  - A match will be created and all the information about it will be stored in the application, all the users of the team whose creator created the tournament will be notified about the fact that their team participates in the tournament, and all the creators of teams whose teams are invited to participate in the tournament will receive notification about this.

- Basic Flow

  1. The user clicks the "create tournament" button.
  2. The application will generate a form to fill in the data.
  3. The user will choose the city of his team.
  4. Next, he will select the required number of teams for the tournament so that the application can draw their lots. The number of teams needs to be a multiple of 2^n.
  5. Further, depending on the number of games that will be conducted with the specified number of teams, the user will need to specify the required number of football fields in the city.
  6. After the tournament is completed, the application will check the correctness of the entered data and will draw the teams' lot and send out a notice to all participants about their participation in the tournament.

- Alternate Flows

  - Within this use case, there are no alternative scenarios.

- Exception Flows

  - Within this use case there are no exceptions.

UC12 - Adding a new private playground

- Brief Description
  - Use case describes the process of adding a new private football field.
- Actors
  - User
  - Web application
- Goals
  - The goal is to add a new private football field for playing matches on it.
- Preconditions
  1. The user must be authorized in the application.
  2. The user must be the owner of this particular field (have the certificate of ownership number for this field).
- Post Conditions
  - A new private field will be added to the application and will become available for future matches and the user will be able to use the administrator part to manage this field.
- Basic Flow
  1. The user clicks the “add new private ground”.
  2. The application will generate a form for specifying the information.
  3. The user will input his data (for identification during verification), then the address, field name, coverage, time of this field and the certificate number.
  4. Next, the application will display a message that the data is being scanned and after validating all data, the user will be notified if the process has completed successfully (the field was added) or not.
- Alternate Flows
  - Within this use case, there are no alternative scenarios.

- Exception Flows
    1. The user will enter incorrect information about the field.
        - An error message will be shown to the user.
    2. The user does not own this particular field.
        - After checking by the application whether the user is actually the owner or not and in the latter case the field will not be added to the application and the user will receive a notification.

UC13 - Managing own playground

- Brief Description
    - Use case describes the possibilities of managing a private field by its owner.
- Actors
    - User
    - Web application
- Goals
    - The purpose of this process is to manage the private field (set the schedule and other characteristics to create a match) to allow other teams to hold matches on this field.
- Preconditions
    1. The user must be authorized in the application.
    2. The user must own this particular field.
    3. The field must be tested for belonging to this user.
- Post Conditions
    - There will be temporary sectors for this field for which it will be possible to book the match, the cost of the matches will be established, as well as other support services and their cost (changing rooms, showers, sports equipment rental).
- Basic Flow
    1. The user will enter the administrator part of the management of his private field.

2. First, he will enter basic information about the opening hours and the cost of holding a match.
3. After the user selects the time interval when he can be on the field and administer matches and press the "save".
4. The application will check the correctness of the entered data and then create temporary slots for the matches by dividing the user-entered interval by 1.5 hours. In case of success of validating all data, this field will become available for holding matches on it.

- Alternate Flows
  - Within this use case, there are no alternative scenarios.
- Exception Flows
  1. The user enters invalid values in the time or cost fields.
     - An error message will be shown to the user.

UC14 - Browsing reservations of user

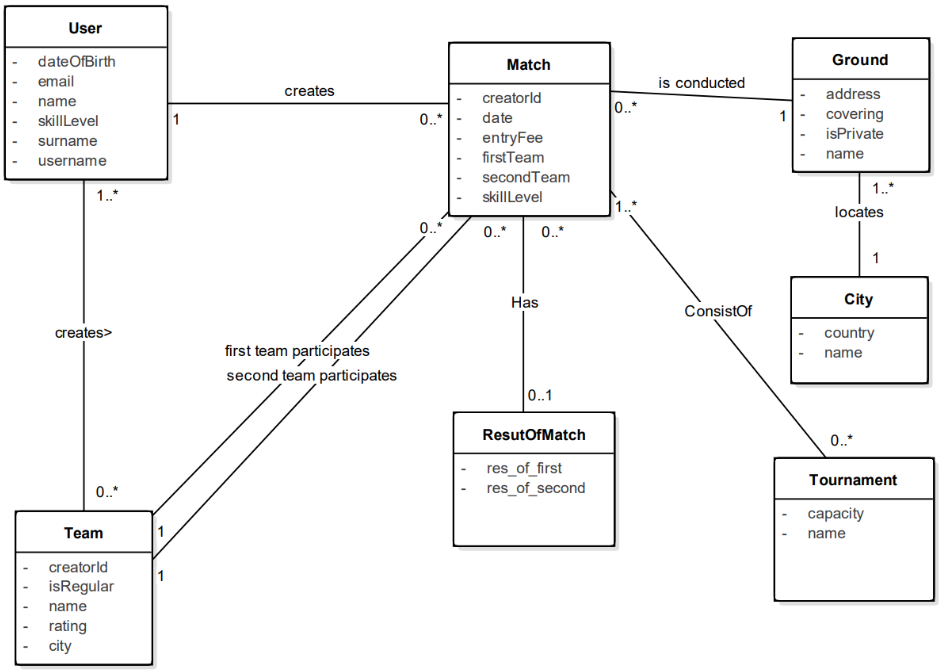
- Brief Description
  - Use case describes the function of viewing the user's reservations in user personal account.
- Actors
  - User
  - Web application
- Goals
  - The purpose of this process is to describe the possibility of viewing the user's reservations available to him.
- Preconditions
  1. The user must be authorized in the application.
  2. The user must successfully pass UC6.
- Post Conditions
  - The user will be shown all the booking matches in which he will participate and information about these matches.
- Basic Flow

1. The user will perform all actions in UC6, thus will have at leastone reservation for the future match.
2. The user selects the menu item “My reservations”.
3. The application will show the user a list of all the reservations in the matches where he will participate.

- Alternate Flows
  - Within this use case, there are no alternative scenarios.
- Exception Flows
  - Within this use case, exception scenarios are not expected.

4.7 Domain model

This section shows the domain model on which the further implementation will be based.



4.8 Chapter summary

In this chapter, based on the analysis of the previous chapter, the functional and non-functional requirements of the web application have been defined. The application will allow the user to create a new account with the help of registration, then the user will be able to search for existing one-time matches and take part in them by selecting the team. Also the user can create one-time matches himself. There will be the possibility of creating a regular team, for its further participation in contests with other regular teams or participation in tournaments. The application will be able to store and add football grounds in a particular city for their further use in the conduct of matches. Public sports grounds will be able to add to the application by any user.

The user will be able to have different roles in the application, depending on his activities. Accordingly, there will be an basic user – a participant in football matches of any kind, the creator of the team is a user with the ability to manage the team he created. The creator of the team will decide his team's participation in competitions with other teams or in tournaments. Also, the owner of the private field will be able to add a private field, which he owns, to the application after passing a certain procedure confirming the authenticity of the rights to own this object. For optimality and ease of implementation of this web application, the use cases of all the application functions were compiled and step by step.

Chapter 5

Implementation

5.1 Current scope development of application

Since as part of this thesis it is necessary to implement a prototype that will allow the user to register in the application, to create his match or find an existing one and take part in it, rather than create a full-fledged application, then to implement these functions at the prototype level, it will be enough to implement the following use case : UC1, UC2, UC3, UC6, UC14.

5.2 PostgreSQL

To manipulate data in this web application, I use the object-relational database management system – PostgreSQL, which supports a data model consisting of a collection of named relations, containing attributes of a specific type.

5.3 Choice of technology

Having compared all the pros and cons of the technologies that are used to create websites, I found out that Java or PHP is most often used for these purposes. To understand the comparison of Java and PHP we must go back in time to the 2000s, when Java brought with Servlets and Struts first concepts for Web applications. Struts framework is based on MVC pattern. Thereby java creating a new vector for the development of Web development technology and first php web framework based on MVC pattern appears only in 2011 [12]. *"Nowadays Java is the ultimate standard for big enterprise web development – banking, insurance, tickets, etc."* [13]

One of the most widely used web site among programmers [14] stackoverflow.com conducted a study between January and August 2017 [15] and collected data on the differences in programming technologies between rich and poor countries and the impact of these differences on the world industry development. A total of 64 countries were surveyed, which had at least five million visits to questions during this period of time. They noticed that requests related to Android, negatively correlated with the country's GDP per capita. So they decided to explore other languages and platforms. The amount of Android and PHP traffic is negatively correlated with the country's income, and Python, Java and R' correlation is positive. In each case, we can see exceptions (South Korea uses Android to a greater extent, and China uses Python), but the correlation is strong.

And at the same time, Java, C and C++ are the most popular programming languages from year to year [16].

I can not say with absolute certainty that the PHP language is a worse choice than Java for the purposes of web development. But, referring to the relevance of technology for the development of my application, I will choose Java, specifically Spring web framework.

5.4 Description of technologies

5.4.1 Java

"Write once Run anywhere." [17]

Java is a platform independent programming language. When a programmer writes a Java application, the compiled code (known as bytecode) runs on most operating systems (OS), including Windows, Linux and Mac OS. Java derives much of its syntax from the C and C++ programming languages, therefore Java is easy to learn. Also other advantages of Java is that it is object-oriented, robust and secure [18].

5.4.2 Spring

"Spring makes it easy to create Java enterprise applications. It provides everything you need to embrace the Java language in an enterprise environment." [19]

Spring is the most popular application development framework for enterprise Java. Millions of developers around the world use Spring Framework to create high performing, easily testable, and reusable code. The basic features of the Spring Framework can be used in developing any Java application.

Inversion of Control The fundamental principle underlying the work of the Spring Framework is Inversion of Control (Ioc). The technology that Spring is most identified with is the Dependency Injection (DI) flavor of Inversion of Control. The Inversion of Control (IoC) is a general concept, and it can be expressed in many different ways. Dependency Injection is merely one concrete example of Inversion of Control. When writing a complex Java application, application classes should be as independent as possible of other Java classes to increase the possibility to reuse these classes and to test them independently of other classes while unit testing. Dependency Injection helps in gluing these classes together and at the same time keeping them independent [20].

For a better understanding of this subject, I will give the following example. For example, we are developing a server component that must process the user registration form and send a message to the user when finished:

```
public class RegisterFormController {

    private static final String msg = "Hello, World!";

    public void send() {

        MailService mailService = new SmtpMailService();

        try {
            mailService.sendMessage(msg);
        }
        catch (MailServiceSendMessageException e) {
            e.printStackTrace();
        }
    }
}
```

Code snippet 1: Server component

The problem is in the object's initialization.

```
MailService mailService = new SmtpMailService();
```

In case we need IMAP email agent instead of SMPT – we will have to change the code.

The `RegisterFormController` class completely controls (composition) the process of creating an email client instance. It would be much more convenient and practical if the code that is responsible for creating instances of classes and establishing links between them is outside of the class, so that this class is not dealt with by this class, but by the user from the outside. This is the inversion mentioned. This approach was called Inversion of Control (IoC).

In the Spring Framework, this is realized in the following way. Spring has Spring contexts (also called Spring IoC containers), which are responsible for instantiating, configuring, and assembling beans. Bean in this case is an equivalent to an object.

This context contains all the declared objects and they are available for use (injection) in any class. One way to do this is to use the *@Autowired* annotation. Using this annotation, you do not have to worry about the best way to transfer to a class or bean a copy of another bean. The Spring framework itself finds the desired bean and substitutes its value in a property that is marked with the *@Autowired* annotation.

modules The Spring Framework is divided into modules. Applications can choose which modules they need. At the heart are the modules of the core container, including a configuration model and an aforementioned dependency injection mechanism. Beyond that, the Spring Framework provides foundational support for different application architectures, including messaging, transactional data and persistence, and web. It also includes the Servlet-based Spring MVC web framework.

To implement my web application and I use the following framework modules:

Spring MVC is a framework based on HTTP and servlets, providing many options for customization.

Spring Security is an authentication and authorization framework: a configurable toolkit for authentication and authorization processes that supports many popular and industry-standard protocols, tools, and practices.

To access the data and store it conveniently in the database, I use Spring Data JPA, which is designed to significantly improve the implementation of the data access layer. It introduces a new abstraction – Repository. This is the interface for accessing data.

5.4.3 Ajax

Ajax is a set of techniques for implementing web interfaces that allows making dynamic queries to the server without a visible reload of the web page: ajax sends a request to the server to retrieve the data and after receiving the data it dynamically inserts the received data into the current web page without an obvious refresh, updating only its specific part.

5.4.4 Bootstrap

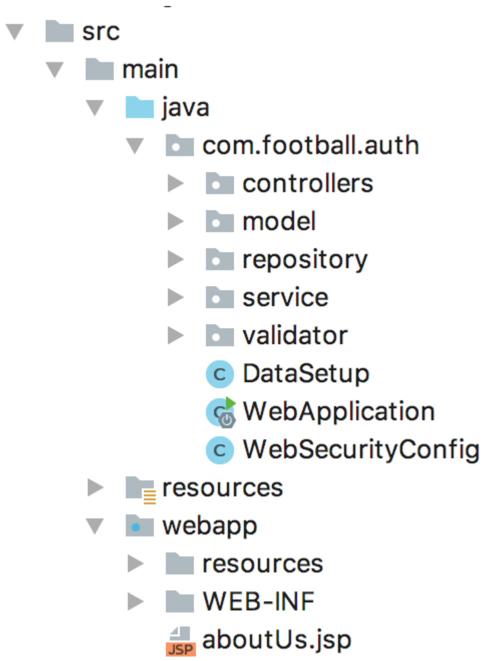
Bootstrap is a css and javascript framework. It contains ready-made styles and scripts that facilitate the layout of web pages and make them adaptive.

5.4.5 Jsp

Java Server Pages is one of the Java 2 Enterprise Edition technologies, which is an extension of the servlet technology to simplify the work with Web content.

JSP pages make it easy to split Web content into a static and dynamic part that allows multiple uses of previously defined components. JavaServer Pages technology allows to generate dynamic content of Web pages, such as HTML and XML files, which will be included in the Web application. JSP files are one way to implement dynamic page content on the server side. As part of the implementation of this web application, Java Server Pages is used to create web pages or create a View (one of the components of the MVC architecture), i.e. to displays all information provided by the controller.

5.5 Project structure



```
▼ src
  ▼ main
    ▼ java
      ▼ com.football.auth
        ▶ controllers
        ▶ model
        ▶ repository
        ▶ service
        ▶ validator
          DataSetup
          WebApplication
          WebSecurityConfig
    ▶ resources
    ▼ webapp
      ▶ resources
      ▶ WEB-INF
        aboutUs.jsp
```

5.5.1 Directory src/main/java

This directory contains all java classes with source code that implement the logic of the web application, the classes responsible for creating models and controllers contains here. I will describe the main classes and their purpose below in each subparagraph.

5.5.1.1 Controllers

This directory contains controllers that are responsible for the progress of the whole web application. I'll give an example of one of the controller's functions on Code snippet 2.

```
@RequestMapping(value = "/registration", method = RequestMethod.GET)
    public String registration(Model model) {

        model.addAttribute("userForm", new User());

        return "registration";
    }
```

Code snippet 2: Controller's function

The registration function has an *@RequestMapping* annotation, which can be accessed by the user from the browser, i.e. the user in the browser pressing the button "Register" sends a request to the address "/registration" and then the application's progress continues in this function, which after performing the necessary actions will send a reply to the browser in the form of a `registration.jsp` file with which will display a form with the registration fields to the user.

5.5.1.2 Model

In accordance with the principles of object-oriented programming and the principles of the relational database ORM will be implemented as follows.

The individual encapsulated entity specified in the Domain model will be defined using standard java classes with Hibernate Annotations that will create a basis for the schemas that Hibernate will load and map to the database as separate tables. Also in these classes, auxiliary getters and setters will be implemented and the relationships between these Objects. The following entities are defined:

City entity represents a city, and information about it – the name of the city and the country. An auxiliary fields is created for specifying the location of the field in a particular city.

Ground entity represents a football field on which the matches will be held, it contains the following information about the field: the name of the field, its address, the coverage material of the site and the city in which it is located.

Match entity is a football match and information about it – match name, date, entry fee, capacity, the field on which this match will be played, the first and second team participating in this match.

Team entity represents a team that will participate in the match, the team can be regular – have a constantly formed team of players and participate in tournaments or can be temporary, therefore be formed for the one-off match. This property is displayed using the parameter `is_regular`.

Entity also carries in itself information about the name of the team, its rating and members of this team.

User is the key entity of this web application. Contains information about username, name and surname.

The following classes will not map to the database, they only serve as helper objects for the operation of the application:

RequestParametrs A helper entity that stores the parameters received from the Ajax request in order to search for this information in the database.

AjaxResponseBody A helper entity that send a response to an Ajax request.

5.5.1.3 Repository

Next, the application uses classes that represent an abstraction for implementing the Spring Data JPA, which was mentioned earlier. These repositories are designed to retrieve information from a database for a particular entity. They include an inherited search mechanism for a given parameter that implements the Spring Framework. For example, the method of one of the repository:

```
City findByCityName (String cityName);
```

This function will return the city according to the specified search parameter `cityName`. This method works in such a way that you can specify any parameter of this object (city name, id, country) after the keywords in the name of the method `findBy` and based on the passed parameter the method will return the object that contains in itself this criterion of the query [21]. There is one repository for each entity listed in the Model section.

5.5.1.4 DataSetup.java

This file is designed to initialize and add data to the database before running the application. To prioritise execution of these operations, the method that performs the action in this class is indicated by the annotation *@PostConstruct* and *@Transactional*. In this class, I add test database data for the proper functioning of the application and the execution of operations that require data for their work.

5.5.1.5 WebSecurityConfigurerAdapter.java

This class is responsible for security configurations, determining which addresses require authorization when making requests.

5.5.1.6 WebApplication.java

The main class of the application from which the application is launched.

5.5.2 Directory src/main/resources

There are configuration files in this directory.

application.properties This file is required to configure individual project parameters, such as for example – which file format the application needs to look for to create a view, then also there are configurations for connecting the project to the database.

validation.properties this file contains the data necessary to validate the information entered by the user.

5.5.3 Directory src/main/webapp

Above I described 2 out of 3 components of the MVC architecture. The third component of the View remained to be described. This component uses templates to determine what the user sees. For these purposes I use .jsp files which, using different tags and attributes, make it easier for the programmer to link HTML code and the necessary data obtained with the previous two Model and Controller components. In total, this application contains 11 unique views that will be displayed as web pages. Some pages during the interaction with the user can be dynamically changed using javascript and AJAX requests. The latter mentioned files are in the subdirectory “js”.

5.5.4 Pom.xml

This is the main file describing the project, in this file the dependencies necessary for the functioning of the project are connected. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects.

5.6 How the application works

The basis of the application is the principle of the Spring Web MVC. All the logic of the Spring MVC work is built around the `DispatcherServlet`, which accepts and processes all HTTP requests (from the UI) and answers them. After receiving the HTTP request, `DispatcherServlet` accesses the `HandlerMapping` interface, which determines which Controller should be called, and then sends the request to the desired Controller. Controller receives the request and calls the appropriate service method based on GET or POST. The called method determines the data of the Model, based on a specific business logic and returns the name of the View to the `DispatcherServlet`.

Using the ViewResolver interface, DispatcherServlet determines which View to use based on the received name. After the View is created, DispatcherServlet sends the Model data as attributes to the View, which is ultimately displayed in the browser.

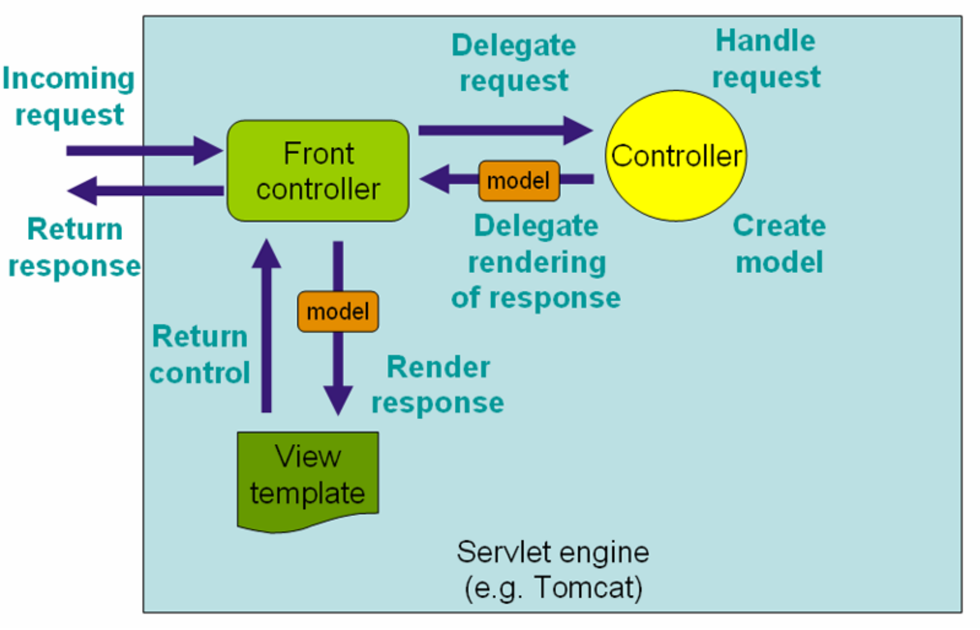


Figure 5.1: Front controller [1]

Another way of processing is possible: after the browser sends an HTTP request, the client (javascript) "intercepts" this request and then creates an instance of the XMLHttpRequest object. A connection is established, an Ajax request is generated, and it is sent to the application. Then the corresponding controller receives the request, processes it and sends a response in JSON format and afterwards the client (javascript) processes the received data and updates a certain part of the page inserting the received data.

5.7 Registration on the site

The new user is registered using the UserController class. It has 2 registration methods. After the user presses the "register" button, the application makes the redirection to "/registration" thereby sending an HTTP GET request. Control is transferred to the SendRegistrationForm method (see Code snippet 3).

```
@RequestMapping(value = "/registration", method = RequestMethod.GET)
    public String SendRegistrationForm(Model model) {
        model.addAttribute("userForm", new User());

        return "registration";
    }
```

Code snippet 3: Function for sending the registration form

This method returns a new form for populating it with the data and view `registration.jsp` for displaying the `registration` form.

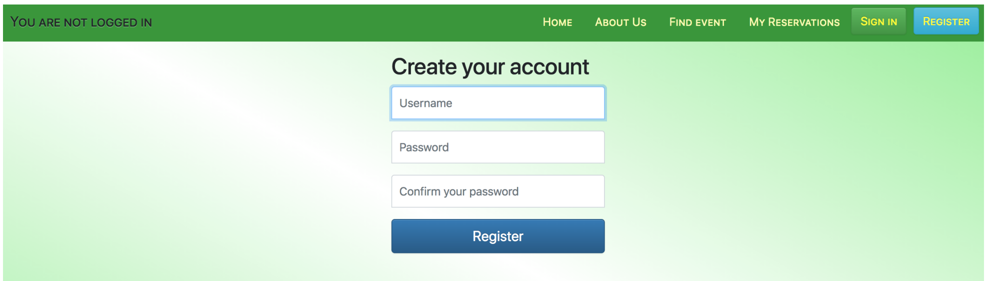


Figure 5.2: Registration form

After filling in the fields with data and clicking on the submit button, the browser sends the HTTP POST request to the controller and it gets into the `MakeRegistration` method (see Code snippet 4).

```
 @RequestMapping(value = "/registration", method = RequestMethod.POST)
    public String MakeRegistration(@ModelAttribute("userForm")
    User userForm, BindingResult bRes) {
        userValidator.validate(userForm, bRes);

        if (bRes.hasErrors()) {
            return "registration";
        }

        userService.save(userForm);

        securityService.autologin(userForm.getUsername(),
                        userForm.getPasswordConfirm());

        return "redirect:/";
    }
```

Code snippet 4: Function for making registration

Next, using the class userValidator, the received userForm is checked with the help of certain rules defined in the file `validation.properties`.

```
public void validate(Object o, Errors errors) {
       User user = (User) o;

       ValidationUtils.rejectIfEmptyOrWhitespace(errors,"username","NotEmpty");
       if (user.getUsername().length() < 5 ||
                       user.getUsername().length() > 32) {
           errors.rejectValue("username", "Size.userForm.username");
       }
       if (userService.findByUsername(user.getUsername()) != null) {
           errors.rejectValue("username", "Duplicate.userForm.username");
       }

       ValidationUtils.rejectIfEmptyOrWhitespace(errors,"password","NotEmpty");
       if (user.getPassword().length() < 8 ||
                       user.getPassword().length() > 32) {
           errors.rejectValue("password", "Size.userForm.password");
       }

       if(!user.getPasswordConfirm().equals(user.getPassword())) {
        errors.rejectValue("passwordConfirm", "Diff.userForm.passwordConfirm");
       }
   }
```

Code snippet 5: Validate function

Then the result of validation is returned back to the `MakeRegistration` method and in case of errors the view registration.jsp with the specific errors is sent again. If the validation was successful then the userForm using the save method of class `userService` saves the new user to the database.

```
@Override
   public void save(User user) {
       user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));
       userRepository.save(user);
   }
```

Code snippet 6: Save function

Next, the application automatically authorizes a new user and redirects him to the home page.

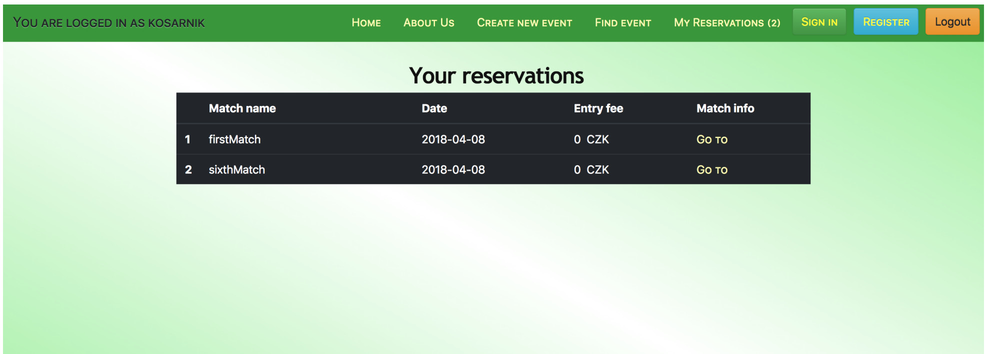
5.8 Participation in football match

The user is added to the team to participate in the existing match using the `SearchController` class. After the user opened the page with information about the match he needs and pressed the “Participate for this team” button HTTP GET Request is sent into the `addNewPlayer` method:

```
@RequestMapping(value="api/addPlayer")
        public ModelAndView addNewPlayer(@RequestParam Long id,
        @RequestParam String team) {

        ModelAndView mv = new ModelAndView("matchDetail");

        //get username of current user
        Authentication auth = SecurityContextHolder.getContext().
        getAuthentication();
        String name = auth.getName();
        User u= uRepo.findByUsername(name);

        Match match=matchRepo.findById(id);
        mv.addObject("entryFee", match.getEntryFee());
        mv.addObject("date", match.getDate());
        mv.addObject("mName", match.getMatchName());
        mv.addObject("team1", match.getTeam_A().getPlayers());
        mv.addObject("team2", match.getTeam_B().getPlayers());
        mv.addObject("id", id);

        if (alreadyParticipate(u,match)) {
            mv.addObject("You have participated this match already!");
            return mv;
        }

        return mv;
}
```

Code snippet 7: Function for adding a new player

According to the received match id, this match is searched for and retrieved from the database using the `MatchRepository` repository, then using the `alreadyParticipate` method, a check is made if the user does not already participate in this match for another team or tries to participate in the same team. After using the received *@RequestParam* attribute, the String team determines which team the user should be defined, this user is added to the corresponding team of the match using the addPlayer function of the Team class. Also, to view own bookings this match , is added using the `addMatch` method. After the invitation to this match, a number will appear in the top menu that displays the number of reservations of this user. Later in the section “My Reservations” authorized user can view all the necessary information about the matches in which he will participate.



Your reservations

	Match name	Date	Entry fee	Match info
1	firstMatch	2018-04-08	0 CZK	Go to
2	sixthMatch	2018-04-08	0 CZK	Go to

Figure 5.3: User's reservations

5.9 Football match creation

A new match is created using the NewMatchController. After the user clicks the "Create new event" button, application goes to sendMatchForm method, where a match object is created to be populated with the information from the user and, together with NewMatch.jsp, is sent to the user for display (see Appendix C). After filling in the form for the match and clicking submit button, form for a match falls into the method CreateMatch:

```
public String CreateMatch(@ModelAttribute("matchForm") Match matchForm) {
     matchService.save(matchForm);
     return "redirect:/findNewMatch";
}
```

Code snippet 8: Function for creating a new match

That method save a new match to the database using the save() method of the matchService class. In this method, also using the parameter groundNumber variable from the database, the desired field is found and is set as the game field of this match. This method also creates 2 teams and adds the current user as a member of the first team for the match he created. Afterwards the match becomes available for search and participation by other users.

```
public void save(Match match) {

    Ground ground = groundRepository.findById((long) match.
    getGroundNumber());
    match.setTheGround(ground);

    Team teamA = new Team(1,"TeamA");
    teamRepository.save(teamA);

    Team teamB = new Team(1,"TeamB");
    teamRepository.save(teamB);

    match.setTeam_A(teamA);
    match.setTeam_B(teamB);

    user.addMatch(match);
    setReservationCountGlobally(user.getMatches().size());

    matchRepository.save(match);
}
```

Code snippet 9: Save function for match

5.10 Economic and managerial summary

5.10.1 SWOT analysis

- Strengths
  - Well-conducted analysis.
  - Well-designed application architecture.
  - Easy modification for further change/improvement.
  - For implementation is using the modern and advanced framework.
- Weaknesses
  - A small project team.
  - Little experience in developing such applications.
  - Tight deadlines for the implementation of this project and its testing.
- Opportunities
  - Limited number of such applications in the market and in most cases they offer incomplete functionality.
  - Low maintenance cost.

  - Low advertising and promotion costs due to the popularity of football and a large number of people wishing to play.

- Threats
  - Project implementation delay.
  - The emergence of a more competitive application
  - Lack of opportunities (football fields) in small towns for organizations of this kind of sports events.
  - Low adoption of the application among the new users, which due to their habit will keep on using more familiar tools for organizing matches.

5.10.2 Benefits

The main advantage of this application for potential users of this web application is facilitating the process of organizing and searching for football matches. Another important advantage is that potential users of this web application will be able to find and make new acquaintances with like-minded people.

For owners of private fields this web application will be a good advertising platform for promoting their football fields.

Also the advantage of this web application is that it will help to avoid collisions when using public sports grounds, which do not have a timetable and no one knows at what time this field will be free or occupied. And thanks to this, the number of interested users of this web application is growing.

For example, if a match is organized through this application on any public field and team members come to the playground and other people who wish to play the game will say that they wanted to use the same filed at this time, then users of this application will be able to refer to this web application by answering that it can be used to see the schedule of organized matches for a particular field to avoid collisions. Also, thanks to this, the number of users of this web application is guaranteed to grow, which is a free advertising and certainly an economically-viable factor.

5.10.3 Costs

In this section I will describe the cost of implementing this project. First, the cost of implementing the application will be described in table 5.1, and then the cost of operation (work and functioning) will be described in table 5.2. The numbers stated below are estimated and approximate figures and by no means represent real data. To calculating the cost, I used man-hour unit mhr and hourly rate of 250 CZK, which corresponds to the salary of Java junior web developer.

Stage	Duration	Cost
Collection and analysis of requirements	10 mhr	2500 CZK
Further analysis and design	20 mhr	5000 CZK
Researching and familiarising with technology	50 mhr	12500 CZK
Implementation	60 mhr	15 000 CZK
Testing	12 mhr	3000 CZK
Installation and putting into operation	4 mhr	1000 CZK
Total	156 mhr	39 000 CZK

Table 5.1: Cost of implementing

The prices for web hosting and domain registration are taken from the hosting of Endora.cz.

Item	Price
License price for the application	15 000 CZK
Maintenance costs	1000 CZK for 1 year
Webhosting	250 CZK for 1 year
Domain registration	1250 CZK for 1 year
Total	1500 CZK for 1 year +15 000 CZK one time

Table 5.2: Cost of operation

5.11 Chapter Summary

To implement the designed application I have chosen Java programming language and Spring Framework. The web pages of the application are displayed using jsp files, bootstrap and ajax technology. The result of the implementation part is a fully functional prototype web application that has all the required functionality. Also the swot analysis, the cost of implementation and management are described. The benefits of this web application are described.

Chapter 6

Solution verification

6.1 Verification the implementation of individual functionality

The implementation contains all the use cases necessary to cover the functionality that was required in the task. The created prototype of the application allows a user to register, search for existing football matches by different criteria and join them. It is also possible to view all your reservations and create a new football match by the necessary criteria.

6.1.1 Registration

Registration of a new user on the website was tested by the author and several other users. To avoid basic errors, all participants were informed of the basic restrictions when entering information (the user name must not be shorter than 5 characters and must be unique, and the password is not shorter than 8 characters). Nevertheless, the author tested these restrictions and the application successfully coped with this, warning the user about these restrictions and the need to re-enter the correct information. As a result, observing the basic restrictions, no problems were found.

6.1.2 Searching and participation in the existing one-time matches.

The search and further participation in one-off matches was also tested by the author and other users. During the match search testing, the author found an error related to the location of some elements (submit button). This error was immediately eliminated and the entire match search script was run without any errors. Next, the user found a logical error related to the fact that the same user could participate in one match simultaneously for both teams or re-participate in the same team. The error was fixed and no further errors occurred in the course of the application's operation.

6.1.3 Creating new one-off match

The creation of a new match was also tested with the help of users. During testing, one unsecured error was detected when a user was able to create a match indicating a date in the past. The error was eliminated and the creation of the match works without problems.

6.2 Test scenarios

This test cases for testing the basic functions implemented in this prototype are located in appendix D. There will be 3 scenarios for: Registration, Participation in existing one-off match and creation of new one-off match.

Chapter 7

Evaluation and recommendations for further development

Due to a preliminary study of the Spring Framework, the overall implementation process went rapidly. The prototype was created almost completely in accordance with the projected concept in the chapter 4. The exception was the addition of an additional link between the user and the match. This arose from the need to take into account all the matches in which this user participates to provide him with this information. Due to the lack of time resources and the priority of creating a high-quality application framework for implementing the prototype, the most important functions were selected. Nevertheless, all the required basic functions were implemented. Thanks to the use of a simple 3-level architecture, the prototype can easily expand and implement the remaining functions that were described in the chapter 4. The process of further development will be described in this chapter.

7.1 Further developments

In this section I will describe the optimal in my opinion course of implementation of the remaining functions of the application.

Adding a public football field As the first and most necessary function, the author recommends the implementation of the addition of a new public football field. At the moment, several public fields are available in the application that are available for matches, but this will not be enough for the development of the application. This function is one of the key components of the process of creating and holding matches among amateurs.

Creating and managing a regular team The next important function is to create a regular team. it is an integral part of many of the following functions such as, organizing a match between two regular teams and holding a tournament. It is impossible without having regular teams to do any of the above actions. Also, the possibility of joining a permanent team will attract more interested users in this application.

Editing user profile The next important function which will allow to realize the individual needs of the user is the editing of the user's personal profile. During the registration only username of the uer represents individual information. For more accurate identification and recognition to other users, the user must be given the opportunity to add other personal data such as: profile photo, name, surname and skill level.

Creating a tournament and a match between regular teams The function that follows the idea from the second point is the creation of a tournament and a match between regular teams

Adding and managing private football fields To expand the choice of places for a match, the functions of adding and administering private football fields need to be implemented.

Maintaining statistics and creating a ranking of teams, Next, the function of collecting and processing the results of matches needs to be implemented. This is necessary to form a ranking of teams and individual players.

Find nearby matches The last function that can be implemented is finding a nearby match. This function is not key and, in principle, the whole process of creating and participating in matches in the case of an individual player will not be limited. But the presence of this function will add interactivity to the application and to some extent simplify the process of finding matches for the user, thereby causing greater interest in using this application.

7.2 Modification and development of the current solution

During the implementation, the author discovered that the use of certain technologies has become obsolete in recent years and new ones have replaced them. In the case of this application, it is the use of technology to create a View – one of the components of the used MVC architecture. From the very beginning, in applications developed using the Spring Framework, jsp files were used for this. But since the information technology industry is the fastest changing, today instead of the jsp standard is the use of the javascript framework – AngularJS [22].

7.3 View to future

To increase popularity, practicality and expediency of the application, other functions can be implemented. These may be functions that allow more accurate evaluation of individual players by the other criteria of interest (player speed, ball possession and level of interaction with the rest of the team's players). Despite the fact that this application was developed as a booking system for football matches, it can be changed into a booking system for any sports involving the formation of teams and competitions between them. With the favorable development and growth of popularity this application can grow into a large sports social network. In this social network, users will be able to register and based on the assessments of other users to form their own sports professional status, develop and find like-minded people.

Chapter 8

Conclusion

The purpose of the work was to assess the current state of the problem of organizing football matches, analyze existing applications dealing with this problem, design a full-fledged application that satisfies all the needs of the end user, followed by the implementation of the prototype and its testing. In the research part, the process of organizing football matches and the attendant problems was defined and described. It described the majority of existing applications dealing with this issue, their pros and cons, and also evaluated to which extent these applications can solve the described task. During the analysis, it was found that all applications can only partially solve the problem, and there is no full-fledged application capable of conducting the whole process of organizing a football match. Based on this analysis and studying the process of organizing matches, the most optimal solution was offered for a reservation system capable of fully satisfying the process of organizing football matches.

In the analytical part, all the functional and non-functional requirements for the booking system were defined, which more accurately determined the future functionality of the reservation system. Further on the basis of all the requirements, use cases were compiled. It was decided that the reservation system would be implemented as a web application. Using the standard methods of building web applications, the architecture of the web application was defined and the Domain model was created. In the implementation part, technologies which were used for prototype implementation were selected and described. The prototype allows the user to create football matches according to the given criteria, and also to search for existing matches and join them. The prototype was tested by potential users. the detected errors uncovered during testing were eliminated. In the end, the recommended process for further development and implementation of the prototype was described. The potential prospects for the development of this web application were described. All tasks were successfully completed.

Chapter 9

Installation guide

This manual describes deploy process of entire application.

Prerequisites

- JDK 1.7 or later
- Maven 3 or later
- PostgreSQL
- Eclipse Java EE Kepler or later
- Spring Tool Suite plugin for Eclipse

Stack

- Spring Security
- Spring Boot
- Spring Data JPA
- JSP

1. First, you need to configure the settings to connect to the postgresql database in the application.properties file. You must specify spring.datasource.url, and also username a password.
2. Next, right-click the project's root directory and select Run as -> Spring Boot App.
3. after the successful launch of the application, you must open the browser and go to http: // localhost: 8080/.

Bibliography

[1] Spring. The DispatcherServlet [online]. [viewed 2018-04-27]. Available from: https://docs.spring.io/spring/docs/3.0.0.M4/reference/html/ch15s02.html

[2] Jdemenato. JDEME NA TO - Online booking [online]. JDEMENATO.CZ, ©2010 - 2018 [viewed 2018-04-27]. Available from: https://jdemenato.cz/reservation/portal

[3] Tymuj. O nas [online]. [viewed 2018-04-27]. Available from: https://tymuj.cz/o-nas

[4] Stripe. The new standard in online payments[online]. [viewed 2018-04-27]. Available from: https://stripe.com/

[5] Swati, S. Acceptance Testing [online]. [viewed 2018-04-27]. Available from: http://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/

[6] Machuca, L. What Is a CMS? All You Need to Know About Content Management Systems[online]. [viewed 2018-04-27]. Available from: https://fitsmallbusiness.com/what-is-a-content-management-system-cms/

[7] Dictionary. webapp [online]. Dictionary.com, LLC. ©2018 [viewed 2018-04-27]. Available from: http://www.dictionary.com/browse/webapp?s=t

[8] Iso-architecture. Architecture [online]. [viewed 2018-04-27]. Available from: http://www.iso-architecture.org/ieee-1471/defining-architecture.html

[9] Krug, S. *Don't Make Me Think: A Common Sense Approach to Web Usability.* ISBN ISBN 978-0-321-96551-6, [cit. 2018-04-27].

[10] Creativebloq. 20 websites that use minimalism beautifully [online]. [viewed 2018-04-27]. Available from: https://www.creativebloq.com/web-design/25-websites-use-minimalism-91516685

[11] SearchSoftwareQuality. use case[online]. [viewed 2018-04-27]. Available from: https://searchsoftwarequality.techtarget.com/definition/use-case

[12] Laravel-news. Laravel Release Process [online]. [viewed 2018-04-27]. Available from: https://laravel-news.com/laravel-release-process

[13] Smartum. Java or PHP [online]. [viewed 2018-04-27]. Available from: https://smartum.pro/blog/java-or-php-what-to-choose-for-web-application/

[14] Intertech. 13 Top Developer Websites [online]. [viewed 2018-04-27]. Available from: https://www.intertech.com/Blog/13-top-developer-websites/

[15] Apptractor. poor and reach countries [online]. [viewed 2018-04-27]. Available from: https://apptractor.ru/info/analytics/bednyie-i-bogatyie-stranyi-ispolzuyut-raznyie-tehnologii-issledovanie-stack-overflow.html

[16] Dev. Python and java on the top[online]. [viewed 2018-04-27]. Available from: https://dev.by/lenta/main/top-3-programming-languages-vs-all-the-rest

[17] Java. Motto of Java [online]. [viewed 2018-04-27]. Available from: https://www.java.com/ru/

[18] Quora. Advantages of Java [online]. [viewed 2018-04-27]. Available from: https://www.quora.com/What-is-the-advantage-of-Java

[19] Spring. Spring Framework Overview [online]. [viewed 2018-04-27]. Available from: https://docs.spring.io/spring/docs/current/spring-framework-reference/overview.html

[20] Tutorialspoint. Spring Framework - Overview [online]. [viewed 2018-04-27]. Available from: https://www.tutorialspoint.com/spring/spring_overview.htm

[21] Spring. Spring JPA[online]. [viewed 2018-04-27]. Available from: https://projects.spring.io/spring-data-jpa/

[22] Quora. advantages of using AngularJS [online]. [viewed 2018-04-27]. Available from: https://www.quora.com/What-are-advantages-of-using-AngularJS-over-JSP-in-Spring-MVC-application

Appendix A

Acronyms

AJAX Asynchronous Javascript and XML

API application programming interface

CMS Content management system

FIFA International Federation of Association Football

GDP Gross domestic product

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

IMAP Internet Message Access Protocol

JPA Java Persistence API

JSP JavaServer Pages

JSON Javascript Object Notation

MVC Model, View, Controller

ORM Object-Relational Mapping

SWOT Strengths, Weaknesses, Opportunities, and Threats

UC use case

UI user interface

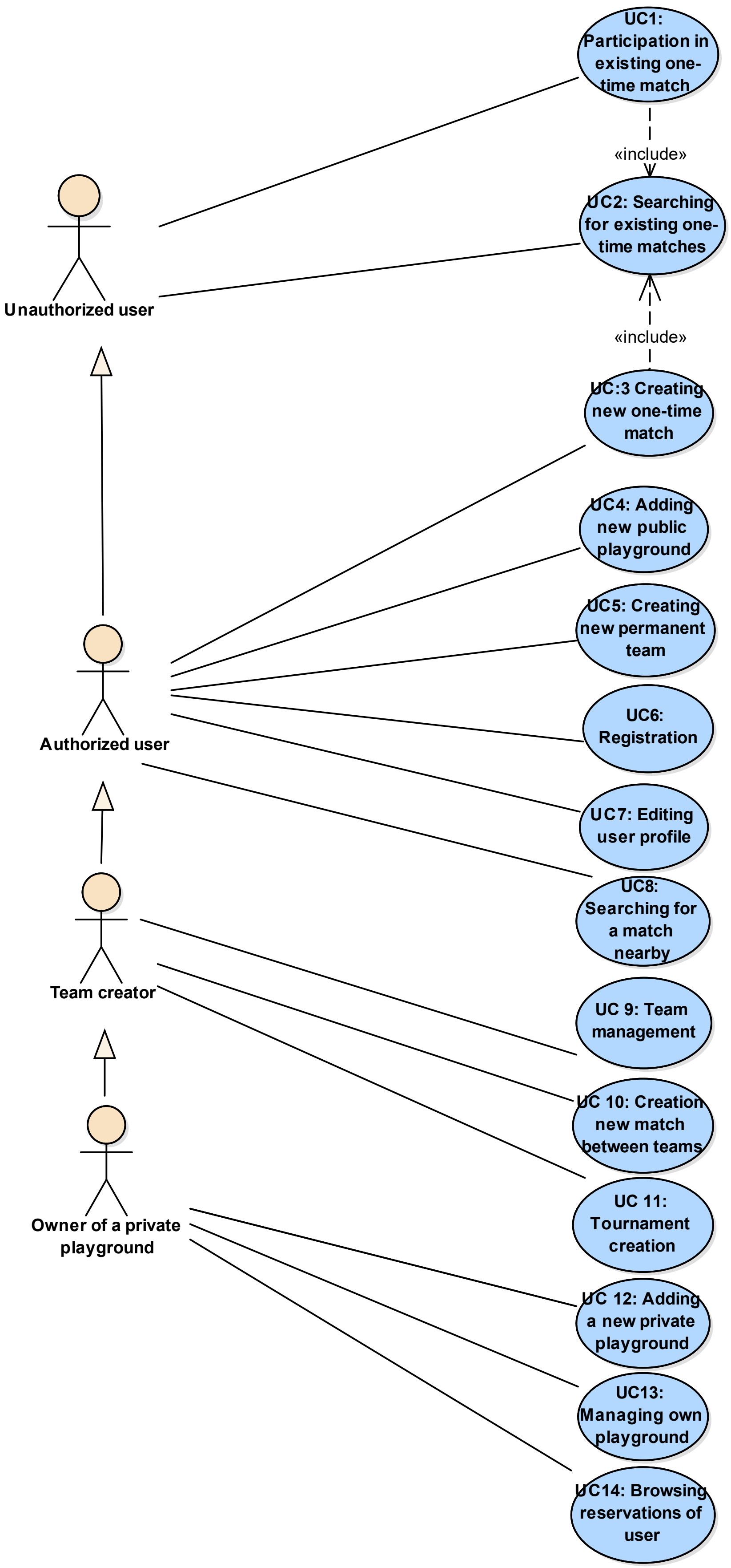
UX User experience

XHTML extensible hypertext markup language

XML eXtensible Markup Language

Appendix B

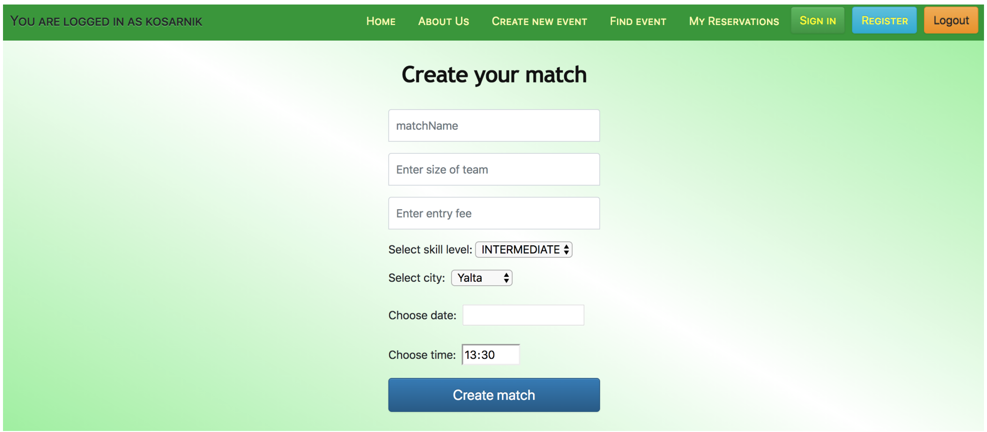
Use case diagram



UC1: Participation in existing one-time match
«include»
UC2: Searching for existing one-time matches
«include»
UC:3 Creating new one-time match
UC4: Adding new public playground
UC5: Creating new permanent team
UC6: Registration
UC7: Editing user profile
UC8: Searching for a match nearby
UC 9: Team management
UC 10: Creation new match between teams
UC 11: Tournament creation
UC 12: Adding a new private playground
UC13: Managing own playground
UC14: Browsing reservations of user
Unauthorized user
Authorized user
Team creator
Owner of a private playground

Appendix C

Illustrations of the application



You are logged in as kosarnik

Create your match

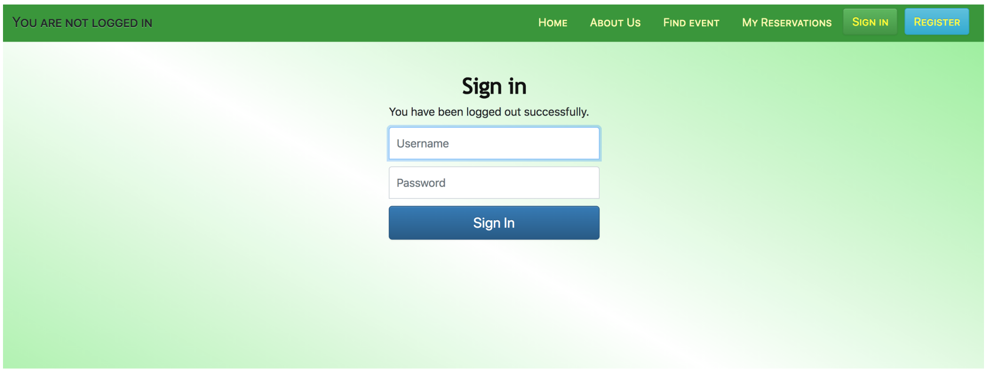
Select skill level: INTERMEDIATE

Select city: Yalta

Choose date:

Choose time: 13:30

Figure C.1: New match creation screen



Sign in

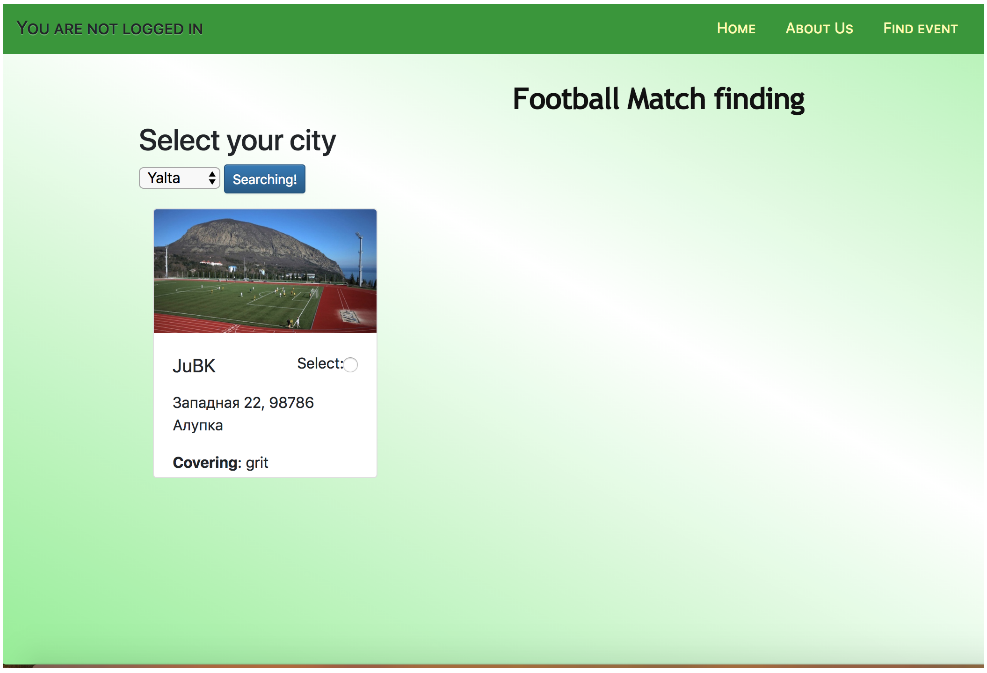
You have been logged out successfully.

Username

Password

Sign In

Figure C.2: Authorization screen



Football Match finding

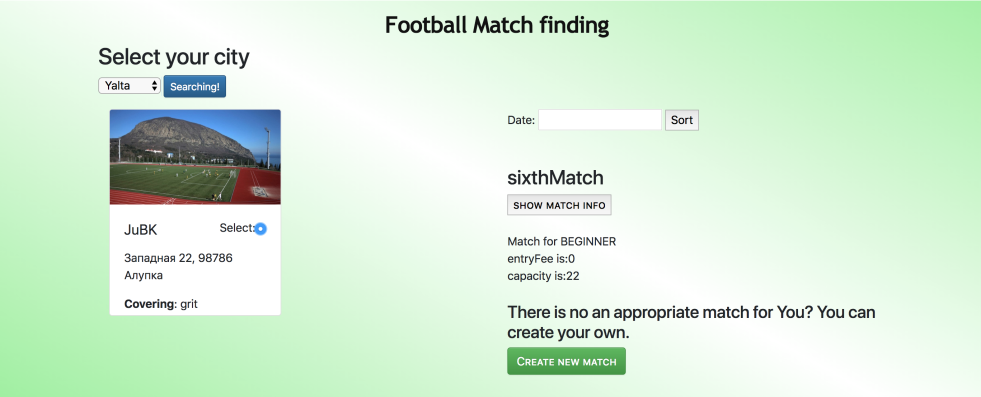
Select your city

JuBK

Западная 22, 98786
Алупка

Covering: grit

Figure C.3: City searching screen



Football Match finding

Select your city

Yalta

Date:

sixthMatch

Match for BEGINNER
entryFee is:0
capacity is:22

There is no an appropriate match for You? You can create your own.

JuBK

Западная 22, 98786
Алупка

Covering: grit

Figure C.4: Match searching screen

Appendix D

Test scenarios

Test Case Field	Description
Test Case ID	TC001
Test Case Summary	Checks that the registration functions properly.
Prerequisites	no prerequisites
Test Procedure	1. Click Register 2. Enter a unique username, password, and password confirmation 3. click the submit button
Test data	1. Username: testusername 2. Password: 12345678 3. Confirm password:12345678
Expected Result	1. A new user account will be created. 2. A message 'Please use between 5 and 32 characters.' Should be displayed if the username is not in this range 3. A message 'Try one with at least 8 characters.' If the password is shorter than 8 characters. 4. A message 'These passwords do not match.' If the password and confirmation do not match.
Actual result	1. If the specified quantity is valid, the result is as expected. 2. If the specified quantity is invalid, the expected message will display.
Status	Test was passed
Created and Executed By	Nikolai Kosariev
Date of Execution	29.04.2018
Test Environment	• OS: Mac OS • Browser: Chrome

Test Case Field	Description
Test Case ID	TC002
Test Case Summary	Verification that the user can take part in the chosen match.
Prerequisites	1. User authorized 2. This match exists
Test Procedure	1. The user will find the necessary match 2. presses the "show match info" 3. Selects the appropriate command and clicks "participate for this team"
Test data	For this test scenario there is no data
Expected Result	1. The user will become a member of the selected match and he will be displayed on his reservations 2. A message ‘You have participated this match already!’ should be displayed if the user is already participating in one of the teams of the match
Actual result	1. If the specified quantity is valid, the result is as expected. 2. If the specified quantity is invalid, the expected message will display.
Status	Test was passed
Created and Executed By	Nikolai Kosariev
Date of Execution	29.04.2018
Test Environment	● OS: Mac OS ● Browser: Chrome

Test Case Field	Description
Test Case ID	TC003
Test Case Summary	Checks if the user can create a new match.
Prerequisites	1. User authorized 2. The same match does not exist
Test Procedure	1. The user selects "Create new event". 2. Enter the required data 3. presses the "Create match" button
Test data	1. match name: testmatch 2. size of team: 11 3. entry fee: 180 4. skill level: beginner 5. city: Prague 6. ground: Sparta 7. date:29.04.2018 8. Time: 19:30
Expected Result	1. A new match will be created and it will be available for search. 2. A message ‘Please enter another name for this match.’should be displayed if the name of this match already exists
Actual result	1. If the specified quantity is valid, the result is as expected. 2. If the specified quantity is invalid, the expected message will display.
Status	Test was passed
Created and Executed By	Nikolai Kosariev
Date of Execution	29.04.2018
Test Environment	● OS: Mac OS ● Browser: Chrome

Appendix E

Contents of enclosed CD

```
readme.txt ........................ the file with CD contents description
├── exe ..................................... the directory with executables
├── src ....................................... the directory of source codes
│   ├── impl .......................................... implementation sources
│   └── thesis .............. the directory of LaTeX source codes of the thesis
└── text ........................................... the thesis text directory
    └── thesis.pdf ............................ the thesis text in PDF format
```