b UNIVERSITÄT BERN

b

U

ClubAdmin

Implementing a Sports Club Event Manager

Bachelor Thesis

Dominik Fankhauser from Trub BE

Philosophisch-naturwissenschaftlichen Fakultät der Universität Bern

29. October 2017

Prof. Dr. Oscar Nierstrasz Claudio Corrodi

Software Composition Group Institut für Informatik University of Bern, Switzerland

Abstract

Today, no existing club management software offers clubs a possibility to organize the human resources at their events in an adequate way. Available applications do not provide organizing committees with enough functionality to ensure that tasks and responsibilities are distributed fairly between a club's members over the course of a full season. Instead, clubs often rely on techniques such as e-mail and spreadsheets to keep track of the tasks at an event or the member collaboration throughout a season.

The use of these simple techniques and processes increases the complexity for the responsible event managers. Their work needs more communication, takes more time, and is more error prone. On the other hand, club members need to keep in mind when and where they have to help at upcoming events as they do not have a way to look up this information except for contacting the responsible person directly.

In this thesis we describe the web application *ClubAdmin* which has been developed for the floorball club *Unihockeyteam Eggiwil*¹. Aside from member, sponsor, and open issue management modules, our system provides an event management module which is accessible to the board (as the event organizer) and to the club members. The club can manage all aspects of event management such that existing cumbersome workflows are fully replaced.

¹www.uht-eggiwil.ch

Contents

1	Intr	oductio	n		1
	1.1	Motiva	ation		. 1
	1.2	Goal .			. 2
	1.3	Outlin	e		. 3
2	Rela	ated Wo	rk		4
	2.1	Fairgat	te		. 4
	2.2	Clubde	esk		. 5
	2.3	Weblir	ıg		. 5
	2.4	Conclu	usion		. 5
3	Reg	uiremei	nts		7
	3.1	Board	requireme	ents	. 7
		3.1.1	Member	management	. 7
			3.1.1.1	Add a member	. 8
			3.1.1.2	Member overview	. 9
			3.1.1.3	View and update a member	. 9
			3.1.1.4	Delete a member	. 9
		3.1.2	Sponsor	management	. 9
			3.1.2.1	Add a sponsor	. 9
			3.1.2.2	Delete a sponsor	. 10
		3.1.3	Event m	anagement	. 10
			3.1.3.1	The season	. 11
			3.1.3.2	The event	. 12
			3.1.3.3	The helper service	. 13
		3.1.4	Open iss	sue management	. 14
	3.2	Memb	er require	ments	. 15
		3.2.1	Event m	anagement requirements	. 15
	3.3	Genera	al requiren	nents	. 16
		3.3.1	Authenti	ication	. 16

CONTENTS

4	Solu	tion	17
	4.1	Technologies and architecture	17
		4.1.1 MVC (Model-view-controller)	18
		4.1.2 MVC in the context of Spring MVC	19
		4.1.3 Spring Security	23
	4.2	User level	25
		4.2.1 Main page and navigation	25
		4.2.2 Member management	25
		4.2.3 Sponsor management	28
		4.2.4 Open issue management	28
		4.2.5 Event management	28
		4.2.5.1 Manage events as a board member	28
		4.2.5.2 Manage personal helper tasks as a club member	34
		4.2.6 Feedback	36
5	Vali	lation	38
	5.1	First meeting: 16th May 2017	38
	5.2	Second meeting: 26th June 2017	39
	5.3	Final evaluation of the board part	40
	5.4	Testing the member part (20th August 2017)	40
	C		
6	- Con	Plusion and Future Work	42
6	Con	Lessons learned 4	42 43
6	Con 6.1	Lessons learned 4 Future work 4	42 43 44
6	6.1 6.2	Clusion and Future Work 4 Lessons learned 4 Future work 4 6.2.1 E-mail connection	42 43 44 44
6	6.1 6.2	Clusion and Future Work 4 Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4	42 43 44 44 44
6	6.1 6.2	Clusion and Future Work 2 Lessons learned 2 Future work 2 6.2.1 E-mail connection 2 6.2.2 Team management module 2	42 43 44 44 44
6 A	6.1 6.2 Anle	Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4	42 43 44 44 44 4 5
6 A	Con 6.1 6.2 Anle A.1	Clusion and Future Work 2 Lessons learned 2 Future work 2 6.2.1 E-mail connection 2 6.2.2 Team management module 2 itung zu wissenschaftlichen Arbeiten 2 Preparation 2	42 43 44 44 44 45 46
6 A	6.1 6.2 Anle A.1	Clusion and Future Work 2 Lessons learned 2 Future work 2 6.2.1 E-mail connection 2 6.2.2 Team management module 2 itung zu wissenschaftlichen Arbeiten 2 Preparation 2 A.1.1 Record the teams 2	42 43 44 44 44 45 46 46
6 A	6.1 6.2 Anle A.1	Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4	42 43 44 44 44 45 46 46 46
6 A	Con 6.1 6.2 Anle A.1 A.2	Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4 A.1.2 Record the sponsor types 4	42 43 44 44 44 45 46 46 48 48
6 A	Anle A.1 A.2 A.3	Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4 A.1.2 Record the sponsor types 4 Manage the events 4	42 43 44 44 44 45 46 46 46 48 48 48
6 A	Anle A.1 A.2 A.3	Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4 A.1.2 Record the sponsor types 4 Add the club members 4 A.3.1 Add a season 4	42 43 44 44 44 45 46 46 48 48 48 48 52
6 A	Anle A.1 A.2 A.3	Lusion and Future Work 2 Lessons learned 2 Future work 2 6.2.1 E-mail connection 2 6.2.2 Team management module 2 itung zu wissenschaftlichen Arbeiten 2 Preparation 2 A.1.1 Record the teams 2 Add the club members 2 Manage the events 2 A.3.1 Add a season 2 A.3.2 Manage an event 2	42 43 44 44 44 4 44 4 46 46 48 48 48 52 53
6 A	Anle A.1 A.2 A.3	Lusion and Future Work 2 Lessons learned 4 Future work 2 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4 Add the club members 4 Manage the events 4 A.3.2 Manage an event 4	42 43 44 44 44 45 46 46 48 48 52 53 53
6 A	Anle A.1 A.2 A.3	Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4 A.1.2 Record the sponsor types 4 Add the club members 4 A.3.1 Add a season 4 A.3.2 Manage an event 4 A.3.2.1 Add the event 4 A.3.2.2 Manage helper services as a member 4	42 43 44 44 44 45 46 46 48 48 48 52 53 53 55
6 A	Anle A.1 A.2 A.3	Clusion and Future Work 2 Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4 A.1.2 Record the sponsor types 4 Add the club members 4 Manage the events 4 A.3.1 Add a season 4 A.3.2 Manage an event 4 A.3.2.1 Add the event 4 A.3.2.3 Close the event 4	42 43 44 44 44 45 46 46 48 48 48 52 53 56 57
6 A	Anle A.1 A.2 A.3	Clusion and Future Work 2 Lessons learned 4 Future work 4 6.2.1 E-mail connection 4 6.2.2 Team management module 4 itung zu wissenschaftlichen Arbeiten 4 Preparation 4 A.1.1 Record the teams 4 A.1.2 Record the sponsor types 4 Add the club members 4 A.3.1 Add a season 4 A.3.2 Manage an event 4 A.3.2.1 Add the event 4 A.3.2.3 Close the event 4 A.3.3 Get hours overview 4	42 43 44 44 44 45 46 46 48 48 53 53 56 57 58
6 A	Anle A.1 A.2 A.3	Clusion and Future Work 2 Lessons learned 4 Future work 2 6.2.1 E-mail connection 2 6.2.2 Team management module 2 itung zu wissenschaftlichen Arbeiten 2 Preparation 2 A.1.1 Record the teams 2 Add the club members 2 Add the club members 2 A.3.1 Add a season 2 A.3.2 Manage an event 2 A.3.2.1 Add the event 2 A.3.2.3 Close the event 2 A.3.3 Get hours overview 2	42 43 44 44 45 46 48 48 48 48 53 55 56 57 58 54

iii

CONTENTS

A.4 Conclusion	65
----------------	----

Introduction

The *Unihockeyteam Eggiwil* is a floorball club located in the canton of Bern. The club has over 200 members, around 40 sponsors and competes with eleven teams in different leagues in the Swiss championship. A board of seven persons leads the club and is responsible for the entire organization.

Most of the challenges the board members face during their work are very typical for a club: keep an overview of the members, manage the club's sponsors, and organize different events (e.g., season matches, sponsor events, and internal activities). Prior to this work, most of these tasks were managed with manual processes like mailing Excel sheets back and forth or simple face-to-face communication. Using these techniques leads to a lot of coordination issues and administrative overhead.

The goal of this work is to replace and simplify the current processes with an online tool for both club and board members. The solution provides facilities to manage members, sponsors, and events in a single, shared place. In the remainder of this chapter, we motivate our work (section 1.1), state the thesis goal (section 1.2), and outline the structure of the thesis (section 1.3).

1.1 Motivation

The tasks and responsibilities of the board encompass a wide range of more or less complex activities. Each board member is in charge of a subset of these tasks. The roles in focus in this thesis are as follows.

The cashier has the responsibility over the member register and updates it once

changes emerge.

The *president* handles everything that is related to the sponsors. This includes updating the sponsor register, negotiating contracts and maintaining the contact with the sponsors.

Last but not least, the two *organizers* manage all the events. One of them is responsible for the organization of all the club's matches while the other one manages all the remaining events such as lottos, markets, and sponsor events.

All events require several persons (from the active members) helping with different jobs like refereeing, cashing in at the bistro, controlling lotto cards, and so on. The constitution says that every member has to help a certain number of hours during a season by doing some of these services. If they do not reach the (not publicly communicated) goal, they can be sanctioned or even suspended. So the two event organizers have to keep an overview of all the hours helped at the events. On the other hand, the different teams have to assist at specific events and split up the services between the team members. Individual club members have to keep track of the services they applied for and of the total hours served so far in a season.

Today, the board mainly uses Microsoft Excel files to manage all the previously mentioned tasks and does not apply any synchronization solution (like $Dropbox^1$). This leads to abstruse situations like this: Assume that the secretary has to write a letter to all club members to inform them about the upcoming match schedule. So first of all, the secretary writes an e-mail to the cashier and asks for the current member register. The cashier replies that the register is currently being updated by the president. So the secretary has to write another e-mail to the president and wait for the next answer. After some days, the current version finally finds its way into the secretary's inbox so that the letter can be sent to all designated receivers.

Another common situation is the following one: At the end of the season, the two event managers meet to determine the total number of hours helped per member and to decide which members have to pay a fee. To answer these questions, they have to collect reports from events (possibly from other people), verify the validity, aggregate the results, and calculate the totals for each member.

Both described situations are unnecessarily complex in spite of all the technical aids, but they still occur very often in the environment of the Unihockeyteam Eggiwil.

1.2 Goal

The goal of this thesis is to replace the existing, Excel-based workflow with a system that covers all requirements and avoids shortcomings of the current workflow. To our knowledge, no existing software solution covers all required features in a combined and

¹https://www.dropbox.com

adequate way. Consequently, a custom solution is necessary.

We developed a web application that allows the board members to manage all the data about members, sponsors, and events in one place. Additionally, we offered access to the system for all the club members so that they can manage and get an overview of their services online without getting insight in the administration part of the board members. This reduced the administrative overhead for the individuals and improved the collaboration between the board and the club members as well.

During the entire duration of the project, we worked closely with our two customers Jakob Limacher, the current president of the Unihockeyteam Eggiwil, and Christoph Wüthrich, a computer scientist and former secretary of the Unihockeyteam Eggiwil, who knows the board's administrative processes well. The latter already tried to find a software solution that fits the requirements, without success. To ensure that the solution also suits the needs of all other stakeholders (board and club members) and is useful to them, we involved them in the development process as well.

1.3 Outline

This thesis encompasses the following chapters:

- **Chapter 2 Related Work:** Gives an overview of existing software solutions in the club management domain and explains why none meets our customer requirements.
- **Chapter 3 Requirements:** Introduces and explains the concrete problem domains and gives a detailed view of the customer requirements.
- **Chapter 4 Solution:** On the one hand, this chapter explains our solution in terms of the architecture and the technical implementation. On the other hand, we show how the system works and looks at user level.
- **Chapter 5 Validation:** Describes the way we validated our solution against the customer requirements together with the customer. Gives insight into how the system changed during the development process due to change requests from the customer.
- **Chapter 6 Conclusion and Future Work:** Recapitulates the project and shows some possible system extensions in the future.

2 Related Work

Today, a wide range of club management applications exists. Before we started our work, we took a look at these systems to make sure none fits the customer requirements already. We found many interesting solutions which provide some functionality to manage a sports club.

We reviewed several applications and found three systems that match some of the customer's requirements. However, none of those covers all requirements. We are briefly introducing these most promising systems in the following sections and explain, for each one, why it cannot be used for the desired purpose.

2.1 Fairgate

Fairgate¹ is a web portal that is used by over 300 sports clubs in Switzerland². It offers member and sponsor management modules that fit most of our needs. As a nice-to-have feature, our customer could manage the billing, the different teams and his internet appearance with this application. Fairgate seems to be very intuitive and user friendly, but has big disadvantages as well. It does not support event management in any sense. Additionally, the software is quite expensive and modules like the sponsor management have to be paid extra. Therefore, Fairgate is not suitable as a solution for the Unihockeyteam Eggiwil.

¹http://fairgate.ch/, accessed 26 February 2017 ²http://www.fairgate.ch/kunden

2.2 Clubdesk

Clubdesk³ is in many ways very similar to Fairgate except for one point: it supports event management. The event management module works similar to Microsoft Outlook, where one can invite people to events, get a calendar overview and see who accepted the invitations. But Clubdesk does not allow users to organize and assign roles (such as referee, cashier, etc.) or time slots. Due to this missing key feature, Clubdesk also drops out as a possible solution for our customer.

2.3 Webling

Webling⁴ is another web solution that provides functionality to manage members, do accounting tasks and manage documents. The member management module looks very similar to Fairgate and would therefore fit our needs. But unlike Fairgate, Webling does not offer any sponsor management. Additionally, the system lacks event management support and is thus rejected.

In general, Webling seems to be a tool that focuses on the administrative board work. The advantages of the system would be the good and easy-to-use role management to handle different access rights as well as a material management module (which no other solution offers). However, these two features are not key requirements for the Unihockeyteam Eggiwil so we reject Webling as well.

2.4 Conclusion

Table 2.1 sums up the capabilities of the three software systems regarding some of our key requirements. While all of the considered solutions offer member and role management functionality, only Clubdesk provides users with a module to manage events. But unfortunately, these event management capabilities are limited and do not include some of the desired features, namely the administration of time slots and roles at events. Therefore, we need to build a customer-specific application that combines member, sponsor, and event management functionalities.

³http://clubdesk.ch/, accessed 27 February 2017

⁴http://webling.ch/, accessed 27 February 2017

	Webling	Clubdesk	Fairgate
Member management	\checkmark	\checkmark	\checkmark
Sponsor management		\checkmark	\checkmark
Event management – member participation		\checkmark	
Event management – time slots			
Event management – roles			
User frontend	\checkmark	\checkmark	\checkmark

Table 2.1: Summary of the capabilities of the solutions regarding our key features.

B Requirements

Our next step was to meet the customer to get an overview of their requirements. We created a software requirements specification document which we discussed in a second meeting with them to verify that we understood everything correctly. Since the system will be split up in two parts, one for the board and the other for the members, we are going to describe the requirements for these two parts in separate sections of this chapter (section 3.1 for the board and section 3.2 for the member requirements). Last but not least, we will introduce system requirements which are relevant to both parts in section 3.3.

3.1 Board requirements

Independent from the individual committee role, board members get access to the full board functionalities so that they can help out each other in busy times or in case of unexpected absences. The system should basically provide four main components for the board: sponsor management, member management, event management, and open issue handling. The requirements for all these modules are described in detail in the following paragraphs.

3.1.1 Member management

The member management allows the board to keep track of active and inactive members and replaces the Excel file which is currently used as member register. The system will make use of the recorded information to associate information about the events

CHAPTER 3. REQUIREMENTS

(e.g., hours served) with the members and to handle the access rights. Every recorded member is a potential system user, but no one should be able to access the system without being a recorded member.

3.1.1.1 Add a member

Every board member should be able to add a new member to the system. The following mandatory fields need to be recorded:

- Sex (male or female)
- First name
- Last name
- Address (street and house number)
- Zip code
- City

The subsequent items are optional values that can be registered for every member.

- Date of birth
- Team
- License
- Two phone numbers
- Two e-mail addresses
- Profession
- Status (active, passive, neutral or leaved the club)

The Unihockeyteam Eggiwil uses some special member types which should be represented. The necessary data fields to record them are as follows.

- **Referee** License number, partner (since every game is led by two referees and referee pairs stay together during a season), remarks, and the salary.
- **Honorary member** Date of promotion (the date of the general assembly at which they got elected as honorary member).
- **Functionary** Function and indication if they are part of the board or not. This second information is a Boolean value and determines the access rights, that is, it dictates whether a user sees the member frontend or the board frontend.

3.1.1.2 Member overview

A system user also wants to get an overview of all recorded members. This general summary should display some of the attributes recorded about the members (e.g., name, address, team, e-mail, and status) and offer a keyword search. Further, the customer wishes to have a function that exports this (potentially partial) view to an Excel file containing all known data about the currently displayed members.

3.1.1.3 View and update a member

The previously mentioned member overview provides buttons to view and edit the information about a member. The view function simply displays all recorded data about the selected member. To change the information, one has to edit the member. In this case, the same fields as in the function to add a member (see section 3.1.1.1) are mandatory.

3.1.1.4 Delete a member

System users need the option to delete a member from the system by clicking a button in the member overview. This process can only be completed if no reference to the selected member exists in the event management section. Otherwise, a message is shown indicating that the member cannot be deleted.

3.1.2 Sponsor management

The sponsor management offers our customer the possibility to manage all sponsors in one place and replaces the Excel sheet that has been used so far. Since the functionality is equal to the member management, we only describe the differences concerning addition and deletion of sponsors.

3.1.2.1 Add a sponsor

One can add a sponsor by recording the following mandatory information:

- Sponsor name
- Address (street and house number)
- Zip code
- City
- E-mail
- Contact name (a representative of the sponsor)
- Sponsor type (e.g., club sponsor, team sponsor, patron, outfitter)

The optional data about a sponsor encompasses the items listed below.

- Phone number
- Contract information (start date, end date, amount paid per year, and remarks)
- E-mail and phone number of the contact person

3.1.2.2 Delete a sponsor

A sponsor can be deleted without checking any remaining references since it is not reused in another system module. Like for the members, the system should display a confirmation dialog before a sponsor is effectively deleted.

3.1.3 Event management

Before we describe the requirements concerning the event management, we need to introduce some terms which will occur repeatedly.

- **Season** Normally describes the time during which the championship is played. In floorball, the championships get under way mid September and finish between February and April. In our terms, a season is equivalent to a club year which lasts from May to April so that every event can be related clearly to one season.
- **Event** For example, a season match, a lotto, or an intern activity which is organized by the club. It is associated with one particular season.
- **Helper service** Every event needs some club members as helpers for different tasks like game secretary, bistro assistant, speaker, and so on. Each of these services is related to a particular event and will take a predefined number of hours.

As already mentioned in the introduction (see chapter 1), every member has to serve a (not publicly communicated) number of hours doing helper services. Members who are off the target risk to be sentenced by the board to pay a fine. The goal is not communicated on purpose because the club does not know the exact number of events at the start of a new season and would otherwise risk not having enough helpers over the course of a season.

We will now introduce the requirements in a top-down approach starting with the season and then going into more details step-by-step, first with the events and then with the helper services. The workflow is illustrated in figure 3.1.



Figure 3.1: Simplified event management workflow.

3.1.3.1 The season

The season acts as a container for the events of a year and the club manages the helper services season-by-season. The new system should therefore allow a season based event management. The new workflow over the course of a season is straight-forward: add a season at the beginning, then modify it throughout the year (manage events, edit the members that have to help, ...), and finally close it at the end.

Add and edit a season

Board members should be able to add a new season to the system by giving it a title and selecting the members that have to help during it. In principle, all recorded club members are required to help and should therefore be selected by default. As soon as a season is added to the system, the selection of the helpers can be adapted by adding new members or deleting existing ones.

Get helper overview

At any point, board members should be able to get an overview of the helpers that shows them how many hours each member achieved so far during the season. Furthermore, they also want a possibility to get a detailed view of the already fulfilled services as well as the planned services for a specific member. The general member overview should be

CHAPTER 3. REQUIREMENTS

searchable and exportable to an Excel file so that, for example, a team's request for the current status can be easily served.

Close a season

At the end of a season, board members can close it so that no more changes can be made. This means that all data recorded for the season are still accessible with read access, but nobody can edit anything.

Manage events

The most important thing a board member wants to manage during the season are the events. We should provide them with an event overview page from where all activities concerning the events can be controlled. The following section takes a deeper look at the customer's perspective of how the event management shall be handled in the new system.

3.1.3.2 The event

An event (e.g., a league match of one of the teams) acts as a container to bundle several helper services to a unit. As already mentioned, every event belongs to exactly one season. The workflow for events is straightforward and resembles the one for seasons: add it, edit it, and then close it.

Add and edit an event

As a board member, one can add an event by specifying the following five items:

- Event date
- Event title
- Event location
- Registration deadline
- Description

Except for the description, all of the values are mandatory and can be changed at any time.

Close an event

Once the event date lies in the past, the event can be closed so that no more changes are feasible. At the latest when a season shall be closed, all events for this season have to be closed.

CHAPTER 3. REQUIREMENTS

A closed event cannot be deleted any more (while pending events can be deleted) and one can only take a look at the recorded information or export it to an Excel file. The export functionality should also be provided for upcoming events so that, for example, the organizer can print out an overview of the members that should be present.

3.1.3.3 The helper service

The last layer of our container model (figure 3.2) is the smallest piece, called helper service. At a league match the club needs, for example, a speaker, somebody cashing in at the bistro, a time keeper, and so on. All these tasks represent one helper service. Therefore, each helper service can be related to exactly one event. We are going to use a similar workflow as before (add, edit, and close helper services for an event).



Figure 3.2: Container model for the event management. The season acts as a container to bundle the different events that belong to it. Each event then is a container to bundle one or several helper services.

Add and edit helper services

Every helper service consists of the following information:

- Task that has to be done
- Start and end time
- Member
- Indication whether the member was present

We agreed with the customer to use a two-step approach to add new helper services to reduce copy-and-paste work. In a first step, the board member enters only the task and the number of required members for this task (so that not every single task has to be created by hand). For example, one could state that two speakers, one time keeper, and three persons in the bistro are needed at the event. Our system will then automatically create the correct number of helper services so that they can be edited in a second step by specifying the time slots and optionally recording a member. While editing a helper service, all data except for the task itself can be changed. If any task is not needed any more, one can delete it.

Close a helper service

The completion of helper services happens in the same step as the closing of an event. To close an event, all helper services have to be completed by filling out all fields and mark if the appropriate member was present or not. This information is very important since the helped hours are only credited if the member was present. Otherwise, no hours are credited and the member may has to help for more hours than others. Helper services are still editable as described before during the completion and new ones can be added.

3.1.4 Open issue management

The last module accessible for board members, the open issue management, is a niceto-have feature for our customer. Today, the board has a monthly meeting where the secretary writes two protocols, one for the open issues and one containing the decisions. These two files are distributed by e-mail afterwards. The open issue protocol consists of a list with tasks and a responsible person for every task. If an open issue is completed, it is deleted from the list at the successive board meeting so that the protocol always contains only the open tasks.

We would like to replace the open issue protocol with an online solution. A board member should be able to add a new task to the system by entering the task and optionally selecting a responsible person (one of the board members recorded in the system). One can also get an overview of all open issues and mark a selected issue as completed.

According to the customer, a completed issue can be deleted from the system without offering any possibility to view the completed tasks again.

3.2 Member requirements

The requirements concerning the second user group are all related to the event management except for one point. Since many members forget to inform the club about address changes, our system should offer the members a possibility to update their contact information in the new system. This means, a member will be able to update the following information of its entry in the member management part:

- Address (street and house number)
- Zip code
- City
- Both e-mail addresses
- Both phone numbers
- Profession

The benefit of this function will be a reduction of dead letters and therefore less extra work for the board members.

3.2.1 Event management requirements

With the new system, the distribution process for the helper services changes substantially. They are not allocated to the different teams and then distributed within the team any more. Instead, every club member has the possibility to manage their services with our new system and is therefore able to fulfill any available helper service.

We provide the members with a search feature to find open helper services. We call a helper service open if no member has already signed up for it. Since some of the services are reserved for specific members or should not be available to everybody, we need to expand the helper services with an additional flag which can be used by the board members to specify if a service can be seen from the members or not. As the entire event management is based on seasons, the search should include only the current season as well.

If one wants to take over a helper service (found through the search), it should be possible to register for this service. The registration should then also be visible in the corresponding event on the board part of the application. We therefore do not need approval from a board member if one wants to take over a specified service. If a member gets a conflict and is not able to do a task for which he is already registered, we should provide a possibility to unsubscribe from the appropriate helper service. Registration as well as withdrawal are only available until the registration deadline. Afterwards, only board members can change the helper services of an event (e.g., by removing a member from a service).

To keep an overview of one's personal services, the system should provide a functionality that allows members to get a summary of all (planned and performed) personal services of a season. According to our customer, a member should see the amount of his personal services during the current season. But one should never get any information about the total hours (or the average) of all members during a season.

3.3 General requirements

The application should have a responsive design so that its usability does not depend on the screen size and can be used on any end device (computer, tablet, mobile phone, and so on). Concerning the system's language, the customer wishes German since this is the mother tongue of the club members. Aside of these non-functional requirements, there are some functional requirements that touch both application parts. These requirements mainly concern the user authentication and are discussed in the following section.

3.3.1 Authentication

The login credentials to the application consist of a username and a password. Since the e-mail address is not a required field in the member management part, our customer agreed to use a system-generated username.

The users for the board members shall be created during the setup of the application. All the other user accounts can be created by the board members trough a provided functionality in the member management. After a user account has been created, the board member can set an initial password for the newly created user and pass it to the corresponding member on another channel (e.g., per e-mail).

Our system should provide every logged-in user with a password reset function. The board members additionally get a password reset function at their disposal so that they can reset the password for every system user. The request for a password reset is sent to a board member on another communication channel.

4 Solution

During a three-month period, we developed the web application *ClubAdmin* by using iterative development. In the first cycle, the member management has been developed since it is the basis for two other modules (event management and open issues). In the second cycle, we finished the member management and then moved towards the event management functionality. We continued the implementation of this module in the third cycle and started the implementation of the sponsor management and the open issue management in parallel. The fourth and last cycle was used for improvements and clean up work.

One of the advantages of this iterative approach was that our customers saw a functioning version of the system three weeks after we started with the implementation and that they could follow the progress of our work throughout the iterations. This reduced the risk of going into a wrong direction since we could collect feedback frequently to make sure that we are on the right track.

In the remainder of this chapter, we explain our solution in detail. We start by introducing the used architecture and technologies in section 4.1 and then describe the user experience in section 4.2.

4.1 Technologies and architecture

The first decision we had to take was about the framework we use for our work. We considered several Java and PHP frameworks which all offer enough functionality to implement the required features. Finally, we have chosen the Java-based *Spring Web*

 MVC^1 framework since it is most familiar to us and it offers all needed functionality to implement the requirements. As the name suggests, Spring MVC is a web framework that follows the Model–View–Controller concept which we will introduce briefly in general before taking a look at the Spring implementation.

We used the build automation tool *Apache Maven*², which simplifies our deployment process. It acts as a dependency manager and can be used to build the project as well.

4.1.1 MVC (Model-view-controller)

MVC is an architectural pattern for software systems which divides the application in the three parts model, view, and controller. There exist a lot of prominent frameworks that implement it, for example, AngularJS³, Ruby on Rails⁴, Laravel⁵, and many others. The benefits of the pattern are that code can be reused (due to the decoupling of the components) and that one can extend the system easily. The components interactions are illustrated in figure 4.1.



Figure 4.1: Interaction between the different components of the MVC pattern.

A *model* holds the application's data (e.g., the information about the members) and contains the business logic. Each model object represents one object from the problem domain (for example a member or an event) and defines therefore one table in the database. All the modifications in the model are initiated by the controller. The model is independent from the presentation (i.e., the view).

```
<sup>1</sup>http://projects.spring.io/spring-framework/
```

```
<sup>2</sup>https://maven.apache.org/
```

```
<sup>3</sup>https://angularjs.org
```

```
<sup>4</sup>http://rubyonrails.org/
```

```
<sup>5</sup>https://laravel.com/
```

The *view* is responsible to display the data it receives from the controller and to offer the user a possibility to manipulate the data with input interactions. A view object knows the model (with all the attributes) it has to represent, but cannot modify the model directly. It is possible to represent the same information with different views depending on parameters (user rights, context, etc.).

Controllers act as intermediaries between the model and the view. They accept user inputs from the views, ask the model to perform a certain action (depending on the input) and then inform the views to update themselves.

4.1.2 MVC in the context of Spring MVC

We will now show how the components of the MVC pattern are implemented and used in Spring. The following sections describe model, view, and controller.

Model

We used Spring MVC together with the Java Persistence API $(JPA)^6$ (an interface that provides a persistence model) and Hibernate⁷, which allows us to represent each model as a simple class. All that is needed is the Hibernate annotation @Entity at the beginning of the class. When using this annotation, the framework knows that the class represents a model and creates the corresponding database table in the background if it does not exist already. The model class itself only contains the attributes for the represented object and accessors (getters and setters) for all these attributes. The following code example acts as a simplified excerpt of the model of a task (which consists only of an id and a name) in our application:

```
@Entity
public class Task {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
}
```

⁶http://www.oracle.com/technetwork/java/javaee/tech/ persistence-jsp-140049.html ⁷http://hibernate.org/

```
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
```

The database itself is accessed through *data access objects (DAO)*. A DAO is an object which is used to access data from a database. The DAO implements a particular interface on which all classes that need to access data depend. With this approach, the application's data access can be separated from the concrete data access implementation and therefore simplifies an eventual change of the underlying database.

In Spring, one only needs to define the interface for the DAO and the framework will then create the concrete implementation automatically so that we do not need to care about it. One possible implementation of these data access objects is to define an interface that extends the Spring class CrudRepository (CRUD stands for the database operations create, read, update, and delete). The methods of the interface represent the executable queries on the table that contains the corresponding model. For example, the DAO for the task model from above looks like this:

```
public interface TaskDao extends CrudRepository<Task,Long> {
   public List<Task> findAll();
   public Task findFirstByName( String name );
}
```

We can query the task table either for all tasks (which are returned as a list) or for a task with a specific name (which is returned as single task object).

Together with the correct configuration, we only inject the DAO into the classes that have to access task data and then we can use it directly to query the database as follows:

```
public class TaskService implements ITaskService {
    @Autowired private TaskDao taskDao;
    public Task getTaskByName( String name ) {
        assert( name != null );
        return taskDao.findFirstByName( name );
    }
    ...
}
```

The method simply returns the task object with the specified name or **null** if no matching entry in the table was found.

Database

Our application uses a $MySQL^8$ database in which the tables are created automatically by the framework. A model of the database schema is shown in figure 4.2. The most interesting part of our system is the event management which uses the six tables in the upper left corner (*Member*, SeasonMember, Season, Event, EventTask, and Task).

The table *SeasonMember* represents an n-to-n relationship between the two models *Season* and *Member* without storing any additional values. We use it to determine which member has to help in which season.

For every season, there can exist between zero and n events which are represented in the table *Event*. On the other hand, every event belongs to exactly one season so that we have a 1-to-n relationship between these two models. The *Event* table stores some additional information about the event such as the date or the registration deadline.

Another 1-to-n relationship exists between the tables *Event* and *EventTask* since every event can be related to zero, one, or several tasks that have to be done at the event, while every event task belongs to exactly one event. We store some more information about the task that has to be done, for example, the start and end time, the visibility, the member that will do the job (in an 1-to-n relationship with the table *Member*), and which task has to be done. This last information is stored in the event task and is associated with exactly one entry from the *Task* table in an 1-to-n relationship as well.



Figure 4.2: Database diagram of the system.

⁸https://www.mysql.com/

View

Our views are implemented as *JavaServer Pages (JSP)*⁹ files. These files can contain both static (in the form of normal HTML content) and dynamic content. For the creation of this dynamic content we mainly use *JSTL (JavaServer Pages Standard Tag Library)*¹⁰, a library of commands for JSP pages. This way, we can avoid plain Java code in the views. For the styling of our application we use the CSS framework *Bootstrap*¹¹ which allows us to implement the desired responsive design with little effort.

The following example shows the combination of JSTL and HTML. First of all, we test with JSTL if a Java object flashMessage has been added to the view (flash messages are notifications that are displayed to notify the user, for example, when a field is not filled correctly). If so, we display a HTML division (div tag) with the Bootstrap class "alert" and the value of the object flashMessageType. The division then simply contains the message the controller has added to the view.

```
<c:if test="${ flashMessage != null }">
    <div class="alert <c:out value="${ flashMessageType }"/>">
        <c:out value="${ flashMessage }" />
        </div>
</c:if>
```

Controller

The controller connects the model and the view as follows: it accepts input from the view, then asks the model to perform a certain action and eventually informs the view to update itself. The mapping of HTTP requests to the controller is specified using annotations. This time, we have to use <code>@RequestMapping</code> to annotate the controller method so that Spring knows which method has to be executed when a request comes in.

In the example below, the first two lines indicate that the method is called as soon as a GET request to the URL "/board/settings/teams" is performed. The method itself then creates a new view from the specified JSP file on the web server and adds two Java objects to it.

```
@RequestMapping( value = "/board/settings/teams",
    method = RequestMethod.GET)
public ModelAndView getDisplayTeamSettings() {
    ModelAndView model = new ModelAndView("settings/teams");
    model.addObject( "teams", settingsService.getAllTeams() );
    model.addObject( "addAndEditTeamForm", new AddAndEditTeamForm() );
    return menuService.getTeams( model );
```

```
<sup>9</sup>http://www.oracle.com/technetwork/java/javaee/jsp/index.html
<sup>10</sup>http://www.oracle.com/technetwork/java/index-jsp-135995.html
<sup>11</sup>http://getbootstrap.com/
```

As mentioned before, we can access and therefore also output all data of the two added objects in our JSP files. The controller does not care about the final representation of the data, it is only responsible to provide the data to the view.

4.1.3 Spring Security

For the authentication and the authorization in our application we used the *Spring Security*¹² framework which is part of the Spring framework family and can be easily integrated. As before, we focus on the most important parts of the configuration here.

Authorization and URL mapping

To handle the access rights for the two system parts, we use two Spring Security roles ADMIN and USER. All club members will get the role USER and the board members will additionally receive the role ADMIN. Together with our chosen URL mapping, this allows us to keep the authorization configuration very simple.

The two entries from our Spring Security configuration file below determine that only users with the role ADMIN can access a URL starting with "/board" while everyone with role USER has the permission to open a page with "/member" at the beginning of the URL.

```
<security:intercept-url pattern="/board/**"
    access="hasRole('ROLE_ADMIN')" />
<security:intercept-url pattern="/member/**"
    access="hasRole('ROLE_USER')" />
```

Table 4.1 shows the URL prefixes we used for the different parts of our system. In case someone tries to access a page without having the necessary permission, we defined the following configuration entry which returns the view specified in the file "403.jsp" (see figure 4.3).

<security:access-denied-handler error-page="/403" />

Authentication

Most of the log-on process is handled automatically by Spring Security so that we do not need to deal with the implementation of it. However, to allow the users access to the appropriate system parts, we need to implement a custom authentication provider and add to the current user the necessary roles. We therefore need to tell Spring Security in the configuration file that it should use our custom authentication provider named "customUserDetailsService":

¹²https://projects.spring.io/spring-security/

```
<security:authentication-manager>
    <security:authentication-provider user-service-ref=
        "customUserDetailsService">
            <security:password-encoder ref="bcryptEncoder"/>
            </security:authentication-provider>
        </security:authentication-manager>
```

In this authentication provider, we then add the necessary roles for the user:

```
private Collection<? extends GrantedAuthority> getAuthorities
  ( Member member ) {
   List<GrantedAuthority> list = new ArrayList<GrantedAuthority>();
   SimpleGrantedAuthority roles =
        new SimpleGrantedAuthority("ROLE_USER");
   list.add(roles);
   if ( member.getIsPartOfTheBoard() ) {
        roles = new SimpleGrantedAuthority( "ROLE_ADMIN" );
        list.add(roles);
    }
   return list;
}
```

So every user of the system gets the USER role, while only board members additionally get the role ADMIN. As already mentioned in chapter 3, the access rights are handled through a Boolean value that indicates whether a member is part of the board or not. Together with the previously introduced configuration for the authorization, we have met all the related customer requirements.



Figure 4.3: Access denied page.

4.2 User level

In this section, we would like to provide the reader with a short insight into the look-andfeel and the features of the application from a user's point of view. We will briefly go through all parts of the application for both board and normal members. The focus is on the event management, since this module has the most complex workflows and offers access for both user types, making it the most interesting part of our system.

4.2.1 Main page and navigation

As soon as board members have entered their credentials correctly, they will be redirected to the main page of the application. This index page looks slightly different depending on the used end device. Figure 4.4 shows what the index page looks like on a desktop computer, while figure 4.5 represents the view on a mobile phone. After customer meetings, we opted for a simple and clean layout that focuses on the core functionality.

For board members, the navigation bar (figure 4.6) contains links to the different parts of the system and also to some settings. We will not discuss the settings in this chapter, but refer the interested reader to appendix A, where we cover these in more detail. In addition, the application offers breadcrumbs right below the navigation to indicate the current location in the navigation hierarchy.

4.2.2 Member management

When board members open the member management, they first see the overview page (figure 4.7). This site contains all the recorded members ordered alphabetically by last name. For each member in the list, the user has different available functions: view all the recorded data about the member, edit it, delete it, or create an account for this member (respectively reset the password if the member already has an account).

The member list can be queried with a full-text search over all displayed attributes. Both a partial list (obtained by applying a search criteria), as well as the entire register

Part	URL prefix
Member management	/board/membermanagement/*
Sponsor management	/board/sponsormanagement/*
Open issue management	/board/issues/*
Event management (board)	/board/seasons/*
Event management (members)	/member/events/*

Table 4.1: URL prefix mapping for core features.



Figure 4.4: Index page of a board member (desktop view).



Figure 4.5: Index page of a board member (mobile view).

1		2			3
Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung -	Eventverwaltung -	Pendenzen	🛱 Einstellungen -
Home 4					

Figure 4.6: Navigation bar and breadcrumbs. (1) Link to the index page. (2) Links to the different system modules. (3) Link to the settings. (4) Breadcrumb.

can be exported as Excel files. The last feature the member overview provides users with is a way to add a new member to the system.

0 T Einträge	anzeigen							s	uchen				
Nachname A	Vorname 🍦	Strasse 🔶	Nr. ♦	PLZ	Ort 🔶	Team 🔶	E-Mail 🔶	Status 🔶	Benutzer 🔶	÷	÷	÷	
Berger	Linda	Altes Bauemhaus	1	5689	Land	-	linda@bauemhof.ch	passiv	Iberger1	۹	<i>.</i>	Ē	-
Doutaz	Andy	Rue de Genève	2	9372	Village	-	andydoutaz@mail.ch	ausgetreten	2	۹	1	Ô	1
Habegger	Adrian	Hauptstrasse	23D	5648	Stadt	Junioren C	habeggers@mailservice.de	passiv	ahabegger1	۹	1	â	-
Jenni	Susanne	Mustergässli	4	4785		Damen 2	susanne@jenni.ch	aktiv	sjenni1	۹	1	Ô	•
Meier	Martin	Dorf	136	2345	Musterort	Junioren U21	martin2001@mail.com	aktiv		۹	1	Î	1
Meister	Martin	Dorfstrasse	17	6589	Auf dem Land	Herren 3	martin@meister.ch	aktiv		Q	1	â	1
Muster	Мах	Mustergasse	1	1234		Herren 1	max@muster.ch	aktiv		۹	1	Ô	1
Müller	Fritz	Dorfstrasse	5A	4433	Stadt	Herren 3	fritz@mueller.ch	aktiv		۹	1	â	
Rothenbühler	Stefan	Bernstrasse	35	8362	Stadt	Herren 2	stefan@stefan.ch	aktiv		Q		Ô	1
Stettler	Andrea	Burgdorfstrasse	ЗA	3847	Beispielort	Damen 1	andrea@andrea.ch	aktiv		۹	1	Ô	1
Mitglied hinzufi	igen 3								Zurück	1	2	Näcl	hste

Figure 4.7: Member management overview. (1) Full-text search. (2) Functions to modify the member. (3) Button to add a new member. (4) Export functions.

4.2.3 Sponsor management

As shown in the database diagram from chapter 4.1.2 (figure 4.2), the sponsor management is completely decoupled from the rest of the system. But it works analogously to the member management. The only differences are the alternative data fields for the sponsors and that sponsors cannot log into the application. We therefore refrain from going into more detail about this module here.

4.2.4 Open issue management

This piece of the application consists of one page and is structured similar to the Word document that has been used previously to track the open issues. As shown in figure 4.8, the upper part displays all the open issues grouped by board member and also a section for issues that are not related to a board member or were related to a member that has been deleted from the system. At the bottom of the page, one can add new open issues by entering the issue and selecting one of the recorded board members as a responsible person.

4.2.5 Event management

Both board and normal members have access to the event management. Since the interface is different depending on the type of user, we describe the interfaces individually in the following sections.

4.2.5.1 Manage events as a board member

If board members select the event management in the navigation bar (see figure 4.9), they have three possibilities. They can either add a new season, go to an active season, or get an overview of the past seasons.

After selecting one of the active seasons, the overview page for this season is displayed (figure 4.10) where one has several options. The upper part of the page contains different functions, for example, the possibility to get an overview over the achieved hours, add an event, edit the members that have to help during the season, export the events, or close the season. The lower part contains a list with all the events of the current season grouped into upcoming events and past events. The past events are split up again into those that are already closed and those that are not yet (which we will name open events from now on). Depending on this status, one has several options for the events. While closed events can only be accessed to read the data, upcoming as well as open events can be edited. For the open events, one has additionally the possibility to close them. All events can be exported but only events that are not closed so far can be deleted.

Manage your club! Mitgliederverwaltung-	Sponsorenverwaltung -	Eventverwaltung -	Pendenzen	🗘 Einstellungen -
Home / Pendenzenliste				
Pendenzenliste				
Andrea Stettler Aktuell keine Pendenzen zugeordnet.				
Linda Berger				
Pendenz				Abschliessen
Eine Pendenz				~
Noch eine Pendenz				×
Vorstand / nicht zugeordnet Pendenz				Abschliessen
Pendenz für den Vorstand				✓
Pendenz für den Vorstand Noch eine Pendenz für den Vorstand				✓ ✓
Pendenz für den Vorstand Noch eine Pendenz für den Vorstand Pendenz erfassen				*
Pendenz für den Vorstand Noch eine Pendenz für den Vorstand Pendenz erfassen Verantwortlicher	Pendenz*			•
Pendenz für den Vorstand Noch eine Pendenz für den Vorstand Pendenz erfassen Verantwortlicher	Pendenz*	en		✓ ✓
Pendenz für den Vorstand Noch eine Pendenz für den Vorstand Pendenz erfassen Verantwortlicher Hinzufügen	Pendenz* ▼ Pendenz eingeb	en		
Pendenz für den Vorstand Noch eine Pendenz für den Vorstand Pendenz erfassen Verantwortlicher Hinzufügen	Pendenz* Pendenz eingeb	en		

Figure 4.8: Open issue management page. The upper part contains all the open issues grouped by member. The bottom of the page contains a form to record a new open issue.

Home Home Description Descri	Manage your club! Mitgliederverwaltung - Sponsorenverwaltung -	Eventverwaltung Pendenzen	💠 Einstellungen -
Herzlich Willkommen! Was möchtest du tun? Mitgliederverwaltung Sponsorenverwaltung Eventverwaltung 2017/2018 Eventverwaltung 2017/2018	Home	Saison hinzufügen 1 2017/2018 2	
Was möchtest du tun? Mitgliederverwaltung Sponsorenverwaltung Eventverwaltung 2017/2018 Eventverwaltung 2018/2019	Herzlich Willkommen!	2018/2019 Abgeschlossene Saisons	
Sponsorenverwaltung Eventverwaltung 2017/2018 Eventverwaltung 2018/2019	Was möchtest du tun?		
Pendenzenliste	Sponsorenverwaltung Eventverwaltung 2017/2018 Eventverwaltung 2018/2019 Pendenzenliste		

Figure 4.9: Menu for the event management module. (1) Add a new season. (2) Open an active season. (3) Get an overview of all past seasons.

The already closed events cannot be deleted since the credited hours for them should stay in the system (independent from the status of the members that helped).

Manage your clui	b! Mitgliederverwaltung → Sponsorenv	erwaltung - Eventverwaltu	ing∓ Pendenzen				¢	Einstellungen▼
Home / Übersicht	der Saison 2017/2018							
Saison 2 Verfügbare Akt	2017/2018 tionen							
Event hinzufügen	Helfer bearbeiten Übersicht Helferstund	en Saison abschliessen	Alle Events exportieren	1				
Anstehend	de Events							
Datum	Anmeldeschluss	Titel	Ort					
20.10.2017	19.10.2017	Lotto	Hirschen Eggiwil			1	â	02
Verganger	ne Events							
Noch nicht al	baeschlossene Events							
Noch nicht al	bgeschlossene Events Titel	Ort						
Noch nicht al Datum 10.11.2016	bgeschlossene Events Titel Waldweihnachten	Ort	1		1	*	â	0 2
Datum 10.11.2016 02.03.2017	bgeschlossene Events Titel Waldweihnachten Meisterschaftsspiel HH1	Ort Wak BOE	1		1	*	Ê	0 2 0
Datum 10.11.2016 02.03.2017 15.08.2017	bgeschlossene Events Titel Waldweihnachten Meisterschaftsspiel HH1 Sponsorenlauf	Ort Wak BOE Schi	d E Lulhaus Horben		1	- J - J - J	Ê	0 2 0
Datum 10.11.2016 02.03.2017 15.08.2017	bgeschlossene Events Titel Waldweihnachten Meisterschaftsspiel HH1 Sponsorenlauf ene Events	Ort Wak BOE Schr	d E ulhaus Horben		1 1 1	\$ \$ \$	â â	0 2 0
Noch nicht al Datum 10.11.2016 02.03.2017 15.08.2017 Abgeschloss Datum	bgeschlossene Events Titel Waldweihnachten Meisterschaftsspiel HH1 Sponsorenlauf ene Events Titel	Ort Wak BOE Sch	d E ulhaus Horben Ort		1 1 1	9 9 9	Î Î	0 2 0
Datum Datum 10.11.2016 02.03.2017 15.08.2017 0.00000000000000000000000000000000000	bgeschlossene Events Titel Waldweihnachten Meisterschaftsspiel HH1 Sponsorenlauf ene Events Titel Weihnachtsessen Weihnachtsessen	Ort Wak BOE Schi	d E ulhaus Horben Ort Restaurant		1	* * *	Î	02 0
Noch nicht al Datum 10.11.2016 02.03.2017 15.08.2017 Abgeschloss Datum 23.12.2016 01.05.2017	bgeschlossene Events Titel Valdweihnachten Meisterschaftsspiel HH1 Sponsorenlauf eene Events Titel Veihnachtsessen Cupspiel	Ort Wak BOE Sch	d iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii		1	• • •	Ê	0 2 0 1 0 2 0 2 0
Noch nicht al Datum 10.11.2016 02.03.2017 15.08.2017 Abgeschloss Datum 23.12.2016 01.05.2017	bgeschlossene Events Titel Maldweihnachten Meisterschaftsspiel HH1 Sponsorenlauf Titel Titel Veihnachtsessen Cupspiel	Ort Wak BOE Sch	d E ulhaus Horben Ort Restaurant BOE		1	• • • Q	î Î	0 2 0 0 0 2

Figure 4.10: Season overview. (1) The different available functions for the season (e.g., get an overview over the achieved hours, add an event, close the season, and so on). (2) Depending on the status of the event, the user has several options for the events.

Add an event

By clicking on the "add event" button, a new page with a form is displayed (figure 4.11). First, all the necessary information about the event has to be recorded like event date, registration deadline (for the members), location, and so on. Afterwards, all the tasks that have to be done at the event can be filled in together with the number of times this task has to be done.

Note::::::::::::::::::::::::::::::::::::		Soonsorenvenvaltung – Eventvenvaltung – Pendenzen – ÖFinstellungen –								
Verex v 0 Vereskut der Saaon 2017/2018 / Event Intruditigen Strongtionen zum Event Verrex: Der Anmeideschuss mus mindesters einen Tag vor dem Event Higgen Verrex: Der Anmeideschuss mus mindesters einen Tag vor dem Event Higgen Verrex: Der Anmeideschuss mus mindesters einen Tag vor dem Event Higgen Verrex: Der Anmeideschuss mus mindesters einen Tag vor dem Event Higgen Verrex: Der Anmeideschuss mus mindesters einen Tag vor dem Event Higgen Verrex: Der Anmeideschuss mus mindesters einen Tag vor dem Event Higgen Verrex: Der Anmeideschuss mus mindesters einen Tag vor dem Event Higgen Det mindester Tag vor dem Event Higgen Strongter Helfer Aurgabe		pponsorchronwalkung - Fonderzon								
Neuen Event hinzufügen Informationen zum Event Informationen zum Event Isgevor dem Event Isgevor Image in Information I	Home / Übersicht der Saison 2017/2018 / Event hinzufügen									
Informationen zum Event Informationen zum Event Informationen zum Event Informationen zum Event Iter Tournament Ben Anneideschluss (im Format TTMM.JJJJ)' Anneideschluss (im Format TTMM.JJJJ)' Beschreibung Beschreibung eingeben Beschreibung eingeben I Aufgabe Anzahl Heffer Anzahl Heffer Anzahl Heffer Anzahl Heffer Anzahl Heffer Anzahl Heffer Anza	Neuen Event hinzufüc	ien								
Minimitation of a constant series and a good dem Event liegen! Ter Tor Tor Tor Tor Tor Tor Tor Tor Tor To										
Hindestruction Ter Tornament Datum (im Format TTMM.JJJJ)* 30.09.2017 Beschreibung eingeben Benötigte Helfer Aurgabe Anzahl Helfer Speaker 2 Aufgabe Anzahl Helfer										
Tournament Bern Datum (im Format TIMM.JJJJ)* 30.92017 30.92017 Beschreibung Beschreibung eingeben Anzahl Heifer Referee 2 Aufgabe Anzahl Heifer Speaker 2 Aufgabe Anzahl Heifer Aufgabe Anzahl Heigeben	Hinweis: Der Anmeideschluss muss mindestens einen 1 Titel*	ag vor dem Event liegeni Ort*								
Datum (im Format TIMM.JJJJ)* 30.09.2017 Beschreibung Beschreibung eingeben Beschreibung eingeben Autgabe eingeben	Tournament	Bern 🖌								
30.092017 20.09.2017 Beschreibung Beschreibung eingeben Benötigte Helfer Aufgabe Anzahl Helfer 2 Aufgabe Autgabe <	Datum (im Format TT.MM.JJJJ)*	Anmeldeschluss (im Format TT.MM.JJJJ)*								
Beschreibung eingeben Beschreibung eingeben Benötigte Helfer Aufgabe Anzahl Helfer 2 Aufgabe Anzahl Helfer 2 Aufgabe Anzahl Helfer Anzahl eingeben Anzahl eingeben Anzahl Helfer Angabe Angab	30.09.2017	▶ 20.09.2017 ◆								
Beschreibung eingeben Benötigte Helfer Aufgabe Anzahl Helfer 2 Aufgabe Aufgabe Aufgabe Autgabe Aufgabe Begeben Autgabe Aufgabe	Beschreibung									
Benötigte Helfer Aufgabe Anzahl Helfer Referee Aufgabe Anzahl Helfer 2 Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl eingeben Aufgabe eingeben Aufgabe eingeben Aufgabe eingeben Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe eingeben Anzahl ein	Beschreibung eingeben									
Aufgabe Anzahl Heifer Speaker 2 Aufgabe Anzahl Heifer Aufgabe eingeben Anzahl Heifer Aufgabe Anzahl Heifer Aufgabe eingeben Anzahl Heifer Aufgabe eingeben Anzahl Heifer Aufgabe eingeben Anzahl Heifer Aufgabe Anzahl Heifer Aufgabe Anzahl Heifer Aufgabe eingeben Anzahl Heifer Impressum Impressum	Benötigte Helfer Aufgabe Referee	Anzahl Helfer								
Speaker 2 Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl Helfer Hinzufügent Impressum	Aufgabe	Anzahl Helfer								
Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe Anzahl Helfer Aufgabe Anzahl Helfer Aufgabe Anzahl Helfer Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe eingeben Anzahl Helfer Impressum Impressum	Speaker	2								
Aufgabe eingeben Anzahl eingeben Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl eingeben Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe eingeben Anzahl Helfer Impressum Impressum	Aufgabe	Anzahl Helfer								
Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl Helfer Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl Helfer Hinzufügent Impressum	Aufgabe eingeben	Anzahl eingeben								
Aufgabe eingeben Anzahl eingeben Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl eingeben Hinzufügen! Impressum	Aufgabe	Anzahl Helfer								
Aufgabe Anzahl Helfer Aufgabe eingeben Anzahl eingeben Hinzufügent Impressum	Aufgabe eingeben	Anzahl eingeben								
Aufgabe eingeben Anzahl eingeben Anzahl eingeben Hinzufügent	Aufgabe	Anzahl Helfer								
Hinzufügen!	Aufgabe eingeben	Anzahl eingeben								
Impressum	Hinzufügen!									
Impressum										
	Impressum									

Figure 4.11: Form to add an event.

Edit an event

In a second step, one can edit the event (figure 4.12) by specifying the time slots for the different tasks, select a member if it is already clear who will do the job, or determine that a service should not be visible to the normal members. Individual tasks can also be deleted from an event by clicking on the trash icon at the end of every task.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung - Eventve	erwaltung - Pendenzen			🗘 Einstellungen -
Home / Übersicht der Sa	ison 2017/2018 / Event be	arbeiten				
Event Tourr	nament bea	arbeiten				
Informationen zum	Event					
Hinweis: Der Anmeldeschlus	s muss mindestens einen Ta	ag vor dem Event liegen!				
Titel*		Ort*				
Tournament		Bern				
Datum (im Format TT.MM.J	JJJ)*	Anmeldeschluss (im Forma	t TT.MM.JJJJ)*			
30.09.2017		20.09.2017				
Beschreibung						
Beschreibung eingeben						
		h.				
Helfereinsätze verw	valten					
Hinweis: Bitte Uhrzeiten im F auswählen, ob der Einsatz öl	Format HH:MM oder HHMM ffentlich sichtbar sein soll od	angeben. Falls die Startzeit vor der Ier nicht (sichtbar = Häkchen gesetz	Endzeit liegt, werden keine Stun t) 2	iden gutgeschriebe	en. Bitte in der h	nintersten Spalte
Referee 1	09:00	12:00	Adrian Habegger (Junioren	C) •		[®] 4
Referee	13:00	17:00		•	3	m
Speaker	09:00	12:00		•	8	<u> </u>
Speaker	13:00	17:00		Ŧ	ē ī	m
Helfer hinzufügen						
Aufgabe		Anzahl Helfer				
Aufgabe eingeben		Anzahl eingeben				
Speichern!						
Impressum						

Figure 4.12: Form to edit an event. (1) Time slots for the different tasks. (2) Drop-down to select a member if it is already clear who will do the job. (3) Check mark to determine if the helper service should be visible to the normal members (if the check mark is set, the service is visible). (4) By clicking the trash icon, the helper service is deleted.

Close an event

After the event has taken place, one of the responsible board members closes the event in the frontend. The form to close the event (figure 4.13) resembles the edit event page. The board member can still edit the time slots and members, delete tasks, or add new tasks. But instead of selecting the visibility, one can specify at the beginning of every line if the corresponding member was present or not. This flag controls the crediting of the hours: if it is set, the hours are credited to the member. Afterwards, the hours are used to calculate the average and amounts in the member overview (see the following chapter) and also displayed in the members profile (see chapter 4.2.5.2).

Manage	e your club! Mitg	liederverwaltung - Spo	nsorenverwaltung -	Eventverwaltung -	Pendenzen	🕏 Einstellungen v
Home	/ Übersicht der Saison	2017/2018 / Event absch	liessen			
Eve	ent "Tourn nationen zum Ev	ament" ab: ^{ent}	schliesse	en		
Helfer	Titel Tour Datum 31.0 Ort Berr einsätze verwalt	mament 17.2017 1 T en				
Hinweis: auswähle	Bitte Uhrzeiten im Form en, ob das Mitglied anwe	at HH:MM oder HHMM ang send war oder nicht (anwes	eben. Falls die Startz send = Häkchen gese	eit vor der Endzeit liegt, tzt).	werden keine Stunden gutgeschrieben. Bit	te in der vordersten Spalte
✓ 1	Referee	09:00	1	2:00	Adrian Habegger (Junioren C)	▼
•	Referee	13:00	1	7:00	Andrea Stettler (Damen 1)	•
•	Speaker	09:00	1	2:00	Linda Berger (-)	•
	Speaker	13:00	1	7:00	Linda Berger (-)	v
Helfer	hinzufügen					
Aufgabe			Anzahl Helfer			
Aufgab	e eingeben		Anzahl eingeben			
Abschli	iessen!					

Figure 4.13: Form to finish an event. (1) Check mark to determine whether the member was present or not. This information is used to credit the hours (if the mark is set, the member was present and will therefore get the hours).

Get member overview

As a board member, one can always get an overview of the current hours standing of the different members. The corresponding page (figure 4.14) displays the current average and all the members with their registered and fulfilled hours. With the icon at the end of every line, one can get a detailed list of all the tasks the selected member is associated with.

Home / Ubersicht der Salson 2017/2018 / Helferübersicht der Salson 2017/2018 Schobübersicht Anzahl Mitglieder: 6 Anzahl Stunden: 400 Durchschnitt: 6.677 Doresicht über die Mitglieder: 10 • Einträge anzeigen Name Geburtsdatum Team Status h geleistet h geplant 0 0 2 2		g				+ Emotoniangon ·
Anzahl Mitglieder: 6 Anzahl Mitglieder: 6.00 Durchschnitt: 6.00	Home / Übersicht der Saiso	n 2017/2018 / Helferübersicht de	r Saison 2017/2018			
Anzahl Mitglieder: 6 Anzahl Mitglieder: 6 Anzahl Stunder: 400 Ururchschnitt: 6.67 1 Dersicht über die Mitglieder 0 CEinträge anzeigen Suchen Name Geburtsdatum O Team Status O hgeleistet O hgeplant O O Linda Berger - passiv 18.0 16.0 Q 2 Andrea Stettler 19.12.1989 Damen 1 aktiv 7.0 7.0 Q Stefan Rothenbühler 02.07.1992 Herren 2 aktiv 4.0 6.0 Q Martin Meister 25.06.1995 Herren 3 aktiv 0.0 9.0 Q Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Adrian Habegger 15.09.2006 Junioren C passiv 7.0 8.0 Q zurück 1 Nachste Export Att Mitglieder exportieren Attuelle Anscht exportieren	lelferübersi	cht				
Anzahl Mitglieder: 6 Anzahl Stunden: 400 Durchschnitt: 6.67 1 Ubersicht über die Mitglieder 0 Einträge anzeigen Name <u>Geburtsdatum</u> <u>Team</u> <u>Status</u> <u>h gelistet</u> <u>h geplant</u> Inda Berger - Andrea Stettler 19.12.1989 Damen 1 aktiv 7.0 7.0 Q Stefan Rothenbühler 02.07.1992 Herren 2 aktiv 4.0 6.0 Q Stefan Rothenbühler 25.06.1995 Herren 3 aktiv 0.0 9.0 Q Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Q Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Zurüc 1 Nachset Export Att Mitglieder exportieren	robübersicht					
Name Geburtsdatum Team Status h geleistet h geplant Linda Berger - passiv 18.0 16.0 2 2 Andrea Stettler 19.12.1989 Damen 1 aktiv 7.0 7.0 2 Stefan Rothenbühler 02.07.1992 Herren 2 aktiv 4.0 6.0 2 Martin Meister 25.06.1995 Herren 3 aktiv 0.0 9.0 2 Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Export Zurück 1 Nachste 2 1 Nachste	Anzahl Mitglieder: 6 Anzahl Stunden: 40 Durchschnitt: 6,6 bersicht über die Mi	⁰ ³⁷ 1 itglieder			Suchan	
Linda Berger - passiv 18.0 16.0 2 2 Andrea Stettler 19.12.1989 Damen 1 aktiv 7.0 7.0 Q 2 Stefan Rothenbühler 02.07.1992 Herren 2 aktiv 4.0 6.0 Q Martin Meister 25.06.1995 Herren 3 aktiv 0.0 9.0 Q Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Export Attuelle Ansicht exportieren Attuelle Ansicht exportieren Attuelle Ansicht exportieren Attuelle Ansicht exportieren	Name	🔶 Geburtsdatum	🔷 Team	▲ Status 🔶 h geleistet	h geplant	\$ \$
Andrea Stettler 19.12.1989 Damen 1 aktiv 7.0 7.0 Q Stefan Rothenbühler 02.07.1992 Herren 2 aktiv 4.0 6.0 Q Martin Meister 25.06.1995 Herren 3 aktiv 0.0 9.0 Q Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Export Xurück 1 Nachster Yurück 1 Nachster	inda Berger		-	passiv 18.0	16.0	Q 2
Stefan Rothenbühler 02.07.1992 Herren 2 aktiv 4.0 6.0 Q Marlin Meister 25.06.1995 Herren 3 aktiv 0.0 9.0 Q Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Export Kulle Ansicht exportieren Aktuelle Ansicht exportieren Kutuelle Ansicht exportieren Kutuelle Ansicht exportieren	Andrea Stettler	19.12.1989	Damen 1	aktiv 7.0	7.0	٩
Martin Meister 25.06.1995 Herren 3 aktiv 0.0 9.0 Q Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Export Image: State	Stefan Rothenbühler	02.07.1992	Herren 2	aktiv 4.0	6.0	Q
Fritz Müller 17.03.1965 Herren 3 aktiv 4.0 4.0 Q Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Zurück 1 Nächste	Martin Meister	25.06.1995	Herren 3	aktiv 0.0	9.0	Q
Adrian Habegger 15.09.2005 Junioren C passiv 7.0 8.0 Q Zurück 1 Nächste Export Aktuelle Ansicht exportieren Aktuelle Ansicht exportieren	Fritz Müller	17.03.1965	Herren 3	aktiv 4.0	4.0	Q
Zurück 1 Nächste Xport Alle Mitglieder exportieren Aktuelle Ansicht exportieren	Adrian Habegger	15.09.2005	Junioren C	passiv 7.0	8.0	Q
	Xport Alle Mitglieder exportieren	Aktuelle Ansicht exportieren			Zurück	1 Nächste

Figure 4.14: Overview of the currently supplied and planed hours. (1) Current average over all members. (2) Icon to get a detailed list of all tasks the corresponding member is associated with.

4.2.5.2 Manage personal helper tasks as a club member

Users that are not part of the board will see a slightly different view after the log-in. As shown in figure 4.15, the navigation bar contains only the active seasons where the

member has to help. For every season, a member has two options: either view the helper services (served and registered) or search for open tasks.

Manage your club!	2017/2018	🕏 Einstellungen -
Home	Meine Helfereinsätze 1 Helfereinsätze suchen 2	
Herzlich V	/illkommen!	
Was möchtest du tun?		
Meine Adressdaten änder Mein Passwort ändern Helfereinsätze für 2017/20	1 18 suchen	

Figure 4.15: Navigation bar for the normal members. (1) View the member's helper services. (2) Search for open tasks.

Search for open helper tasks

By selecting this functionality for one of the available seasons, the member gets an overview of all helper services at upcoming events where the registration deadline has not yet passed and where no member is recorded so far. For every helper task, a green button to register oneself is available (figure 4.16).

Manage your club	! 2017/2018 ~				💠 Einstellungen -
Home / Helfereinsä	tze suchen				
Offene H	elfereinsätze	e der Saison 20	17/2018		
Datum	Event	Tätigkeit	Beginn	Ende	
30.09.2017	Tournament	Speaker	09:00	12:00	Anmelden
30.09.2017	Tournament	Speaker	13:00	17:00	Anmelden
20.10.2017	Lotto	Speaker	19:00	23:00	Anmelden
20.10.2017	Lotto	Kartenkontrolleur	20:00	23:00	Anmelden
Impressum					

Figure 4.16: Overview of all open helper tasks at upcoming events. By clicking the green button, the member subscribes for the corresponding helper service.

Get overview of personal tasks

The other option offers the member an overview of all tasks of the current season as shown in figure 4.17. This means, the member sees all the past as well as upcoming services it is related to. For those tasks with an open registration window, the member can cancel the registration by clicking on the red button at the end of every entry. Additionally, the member sees the number of hours helped and planned.

Manage your	club! 2017/2018 -						🖨 Einstellungen -
Home / Meine	e Helfereinsätze						
Meine	Helfereinsätze	der Saiso	on 201	7/20	18		
Datum	Event	Tätigkeit	Beginn	Ende	h	Status	
10.11.2016	Waldweihnachten	Bistro	20:00	23:00	3.0	Noch nicht abgeschlossen	
02.03.2017	Meisterschaftsspiel HH1	Bistro	18:00	20:00	2.0	Noch nicht abgeschlossen	
01.05.2017	Cupspiel	Bandenrichter	16:00	20:00	4.0	Abgeschlossen	
15.08.2017	Sponsorenlauf	Rundenzähler	14:00	17:00	3.0	Noch nicht abgeschlossen	
30.09.2017	Tournament	Speaker	09:00	12:00	3.0	Geplant	Abmelden
Stunden geholfen Stunden geplant: Impressum	: 4.0 11.0						

Figure 4.17: Overview of all personal tasks for the current season. By clicking the red button, one can cancel the registration for the corresponding helper service. (1) Number of hours helped and planned.

4.2.6 Feedback

In cases where the user tries to perform an invalid action (e.g., trying to view a member with an id that does not exist), we rarely use dedicated error pages. Instead, we tried to implement most of the feedback by flash messages that are displayed on the page on which the error occurred. This way, the user can directly retry the action that caused the misbehaviour. Apart from the error flash message types, the system also uses warning and success flash message types to inform the user about the result of the performed action. Examples of the three flash message types can be seen in figure 4.18. The flash messages are always displayed right below the breadcrumb at the top of the page content.

Additionally, we used Javascript to validate the forms on the client side so that users get quick feedback through the user interface. For example, the fields on the upper part of the form in figure 4.11 have a green border and a mark to indicate that they have been

filled out correct. If one would, for example, enter a letter in a field that accepts only numeric values, the border would be red and the field would contain a cross.

After the submission, every form is validated on the server to make sure that we only store valid data in the database. If this validation fails, the user gets informed about the error and has to adapt the faulty fields in the form.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung -	Eventverwaltung -	Pendenzen	🗢 Einstellungen -
Home					
Sample text for a succe	ess flash message.				
Sample text for a warning	ng flash message.				
Sample text for an error	r flash message.				

Figure 4.18: The different flash message types.

5 Validation

We offered our customer (represented by Jakob and Christoph) access to the system already in a very early stage of the development process. The deployed system was always fully functional so that they could use the system and follow changes without the need of additional install or update instructions. All current board members got access to the system as well, since they will be the main users in the future, and may provide valuable input and requirement changes too. All system testers were encouraged to give us feedback about errors and change requests at any time.

In addition to the continuous review and feedback, we had two formal meetings with our customer to discuss the current standing of our system and possible modifications and changes to the requirements. We will now present the findings from the two meetings and then introduce some of the feedback we got from the board members.

5.1 First meeting: 16th May 2017

The focus of the first meeting with Jakob and Christoph was on the two modules member and sponsor management, as well as on some general topics like the structure and design of the application. Due to the good preparation of our customer, we simply discussed their issues and remarks point-by-point. The most notable points that should be changed in the system after the meeting were the following ones:

• All lists (member overview, sponsor overview, settings) should be displayed in alphabetical order by name.

- The export should create Excel files (replacing the comma separated file output in the initial requirements). In addition, it should be possible to export partial views from the member and sponsor overview (for example after searching for an entry) and not only the entire register.
- The customer wished to extend the model of the sponsors with the "remarks" field for special cases (for example, to note why a sponsor does not have to pay).
- According to the club's constitution, a board member that does not play in one of the teams is not a formal club member. But since their address data should be recorded in the system as well, we introduce a neutral status that can be used for these members.

The initial design ideas were accepted by the board with only minor comments and change requests.

5.2 Second meeting: 26th June 2017

In the second formal meeting with Jakob and Christoph, we focused mainly on the account creation and the event management. Since the customer was again very well prepared, we proceeded like at the first meeting. Among other things, we made the following decisions and changes to the requirements:

- Contrary to the original requirements, the e-mail address should not be a mandatory field any more since the youngest members may not have a personal e-mail account. This also impacted the log-on process, since it does not make any sense to use a field that is not mandatory as a username. We therefore decided together with the customer to use the log-on process described in chapter 3.
- Since we had no requirements for the design of the main page (the one displayed after the log-on), we asked the customer about it. They stated that it is enough if the page contains links to the different system parts and nothing else.
- Members with related event management data records cannot be deleted due to database constraints. For these members, the status of the member should be set to "left the club" instead.
- Board members should have a possibility to specify if an event task has public visibility (and therefore normal members can subscribe for it) or if it is only visible to the board members.
- Members that have registered for an event task should be able to cancel their registration. But this function should only be active until the registration deadline.

In addition to these findings, we also received feedback from the other board members by mail or during regular board meetings. Most of the questions regarding features or the lack thereof could be answered directly by referencing to the software requirements specification and possibly informing them that the desired functionality was not part of the project scope. One person requested the ability to rename teams, which can be, for example, necessary after a team is relegated (since the team name normally contains the league in which the team plays). After consulting the customer, we added this feature to our solution.

5.3 Final evaluation of the board part

After having implemented all features and the change requests from the second meeting, we asked our customer to test the system again and give us feedback. Since we got the reply that everything was the way they wanted it and that they had no more change requests in the scope of the project, we did not schedule an additional formal meeting to discuss the system again and decided that the board part will remain the way it is.

The customer agreed that the initial SRS, along with the changes made during the meeting, has been fulfilled. Our system meets all must-have as well as most of the nice-to-have requirements. However, our customers have more things in mind that could be part of the system in the future. We will discuss some of these possible system extensions in the next chapter.

5.4 Testing the member part (20th August 2017)

As we had only tested the system with board members so far, we decided to do a test for the other part together with two club members. To assess the user-friendliness of the website, we designed two tasks for the members and observed them as they solved them.

First, we asked them to register for a service at one of the upcoming events for the current season. Both of them completed the task in few seconds by clicking on the link on the main page and clicking the subscribe button for the specified task afterwards. So both testers had chosen our designated path to complete a registration without any detour. In the discussion that followed, both stated that it was quite obvious what they had to do and they could not suggest any improvements.

The second task we gave our testers was to find out the number of hours they had helped so far during the current season. While one of the testers found the correct information via the designated path within some seconds, the other struggled a little bit. He first tried to find the appropriate information by using the same link as before. After taking a closer look at the page, he finally found the correct link in the navigation bar and could tell us the correct number of hours. In the discussion, the second tester explained that he thought that one could find everything directly from the main page and did not even notice that a navigation bar was present. He therefore suggested to add the links to the overview of the personal services to the main page as well so that one can find everything directly from there. We ended up adding this accordingly.

Additionally, we received a suggestion of improvement from the first tester which we have implemented as well. He recommended to move the number of hours helped and planned in front of the overview of the personal tasks since this is the information most of the members will be interested in when opening this page.

Aside from these two design issues, both testers described the application as usable and straightforward so that one can find the needed information. We therefore decided to leave the rest of the member part as it is.

Conclusion and Future Work

In this thesis, we presented a software solution that allows sports clubs to manage their member, sponsor, and event data in one place. We developed the application for the specific needs of the floorball club Unihockeyteam Eggiwil. We conclude that the system covers all requirements and will be used by the club to replace the existing solution.

The Unihockeyteam Eggiwil is led by a board of seven persons of whom two are responsible to organize and manage the club events (e.g., league matches, sponsor events, and so on). All these events require a lot of helpers doing jobs like refereeing, cashing in at the bistro, or being a speaker. At the end of a season, all members should have helped approximately the same amount of hours. To check this as well as to manage the distribution of the task between the different teams, the event organizers use simple Microsoft Excel files. The member register and the sponsor register both consist of an Excel file as well, which leads to a lot of time-consuming communication between the board members to get the latest versions of all these files.

Before we started our project, we reviewed several existing applications to ensure that no one already fulfills the customer requirements. Indeed, all current solutions only offer little to no event management functionality so that we had to develop an own application. We had two meetings with the customer to collect all the requirements. We then wrote a software requirements specification to determine the project scope on which the customer agreed.

As a next step, we developed the web application ClubAdmin during three months using iterative development. The solution bases on the two frameworks Spring MVC and Spring Security and covers all the must-have requirements and most of the nice-to-have requirements. To make sure that the solution is useful to the customer, we offered them access to a running version of the system throughout the development process and encouraged them to give us feedback at any point. Additionally, we had two formal meetings to talk about the system and change requests. At the end, the customer agreed that the system fulfills the SRS accepted at the begin of the project and therefore, the club plans to use the system in production.

6.1 Lessons learned

Over the course of this project, we faced several challenges concerning different topics. On the technical level, the biggest challenge was the limited and cumbersome form handling capabilities of Spring MVC. It took quite a long time to figure out how one can generate dynamic forms using the framework (i.e., forms with a dynamic number of fields). We use such forms, for example, to allow users the modification of all helper services of an event at the same time.

Concerning the implementation, our biggest issue was the maintenance of tests and code documentation. One needs a lot of self control to always update the tests and all related information when changing a piece of code.

Another challenge on a completely different level was writing down this report since it was our first longer report in a scientific context. Therefore, it took some time to become aware of the correct language and the dos and don'ts when writing to a scientific community.

On the other hand, we learned a lot on managing a software project and writing a scientific report. Due to our pre-existing domain knowledge, we got a good understanding of the requirements and needs which helped us throughout the whole project. We also benefited from our previous work with Spring MVC in the sense that we did not loose a lot of time for setting up the framework. This allowed us to spend most of the time on programming the application during the implementation phase. Therefore, we were able to fulfill all the customer requirements within the scope of this project.

Finally, we also profited from the good collaboration with the customers. Whenever we asked them, for example, to test a new functionality, we received feedback within a short amount of time. This encouraged us to keep working on the project even in stressful times.

6.2 Future work

Additional features that are out of scope of this project have been suggested by the client. The two biggest extensions that may be implemented in the future are described in the following sections.

6.2.1 E-mail connection

Nowadays, the board normally uses letters to inform the members about important topics and news. This leads to a lot of administrative work for the secretary (print and pack the letters) as well as costs for the club (for the stamps). To reduce the incurred workload, it would make sense to inform the members by e-mail. Since our application maintains all the necessary information (especially the e-mail address), it would be the easiest to send the mails directly from our system. Members without an e-mail address would still receive the information via letters.

Such an e-mail connection could be used to send reminders as well. For example, the president currently needs to track the expiration of sponsor contracts on his own. If he misses to update the contract, the sponsor does not pay any more and the club lacks money. In this case, we could use an e-mail connection to remind the president some weeks in advance about the expiration.

6.2.2 Team management module

One of the board members is responsible for the athletic sector and, amongst other things, schedules the games for all the teams. To complete this task, information about the absences within the different teams has to be provided from the coaches (so that the person does not schedule a game when half the team is on vacation). On the other hand, the board often gets requests from trainers to provide them with the latest contact information of their team members.

One possible solution to reduce the communication overhead between the responsible board member and the coaches could be, for example, enlarging our system with a team management module. In this module, the coaches would see the latest available contact information (coming from the member management) and the presence of their players. This last information could be entered by the players themselves if we offer them access to this system component as well. With this extension, the scheduling could happen without sending e-mails back and forth.

A

Anleitung zu wissenschaftlichen Arbeiten

In this tutorial, we are going to show how one can use the application ClubAdmin to manage an entire season of a sports club. To do this, we assume that the application has been installed correctly and that at least one board member is recorded in the database so that we can log into the application.

We use the following scenario for our tutorial even if we know that it has not the dimension for which we designed the application. The curling club *CC Software Composition* has ten regular members (divided in two teams) and is led by a board of two additional persons. In this tutorial, we play the role of a board member. The season of the club lasts from September to April. One of the two teams only meets once a month to play some hours of curling for fun while the other team trains weekly and participates in several tournaments in the region as well. Some of these tournaments are organized by the club itself and require therefore a lot of manpower from the members. To earn some more money in addition to that raised by the bistro at the tournaments, the club participates at the local market twice a year by selling chicken wings to the crowd.

The constitution of the club says that every member from the fun team needs to help at events at least 5 hours a year, while the members of the other team have to help at least 15 hours. If players do not help enough hours, they will be forced to pay double the member contribution. To keep an overview of the supplied hours, the club will start to use our system right now.

A.1 Preparation

Before we can start using the system, we need to set up our settings. To do this, one needs to be logged into the system.

A.1.1 Record the teams

We have to record the different teams that are playing for our club so that we can relate the members with the corresponding teams. As shown in figure A.1, we select *Einstellungen* \Rightarrow *Teams verwalten* (settings \Rightarrow manage teams) in the navigation bar, which opens the empty team overview. We can enter in the form at the bottom the abbreviation and the team name and then click *Hinzufügen* (add) to record our fun team (figure A.2).

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung -	Eventverwaltung -	Pendenzen	🗢 Einstellungen -
Home					Teams verwalten
Herzlich W Was möchtest du tun?	/illkommen	!			Meine Adressdaten ändern Passwort ändern Abmelden
Mitgliederverwaltung Sponsorenverwaltung Pendenzenliste					

Figure A.1: Navigate to *Einstellungen* \Rightarrow *Teams verwalten* (settings \Rightarrow manage teams) to manage the teams.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung -	Eventverwaltung-	Pendenzen	🕏 Einstellungen -
Home / Teams verwal	ten				
Teams ver	rwalten				
Aktuell erfasste Keine Teams erfasst.	e Teams:				
Team erfassen					
Abkürzung* Fun Hinzujiigeni		Teamname* ✓ Fun Team		~	
Impressum					

Figure A.2: The information about the fun team are recorded in the form.

The team we just added then appears in the team overview (see figure A.3). We can edit this team by clicking on the pen icon or delete it with a click on the trash icon.

After having added the competitive team as well, our team settings page looks like in figure A.4.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung -	Eventverwaltung -	Pendenzen		💠 Einstellungen -
Home / Teams verwalte	en					
Team erfolgreich hinzuge	efügt!					
Teams ver	walten					
Aktuell erfasste	Teams:					
Abkürzung	Teamna	me				
Fun	Fun Tea	m	1		a 2	
Team erfassen						
Abkürzung*		Teamname*				
Teamabkürzung eingebe	n	Teamname einge	eben			
Hinzufügen!						
Impressum						

Figure A.3: The view after the fun team has been added. (1) Edit team. (2) Delete team.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung-	Eventverwaltung -	Pendenzen		🗢 Einstellungen -
Home / Teams verwalt	len					
Teams ver	walten					
Aktuell erfasste	Teams:					
Abkürzung	Teamn	ame				
Comp	Compe	titive Team	1		â	
Fun	Fun Tea	am	1		â	
Team erfassen						
Abkürzung*		Teamname*				
Teamabkürzung eingebe	en	Teamname einge	eben			
Hinzufügen!						
Impressum						

Figure A.4: Team overview after having added both teams.

A.1.2 Record the sponsor types

Since we want to use the sponsor management module as well, we record the different sponsor types in a similar way via the navigation point *Einstellungen* \Rightarrow *Sponsorentypen verwalten* (settings \Rightarrow manage sponsor types). The functionality of the sponsor management is similar to the one for the member management which we will describe in the following section.

A.2 Add the club members

In a second step, we have to add all the club members to our system. To do so, we select *Mitgliederverwaltung* \Rightarrow *Mitglied hinzufügen* (member management \Rightarrow add member) in the navigation bar. As shown in figure A.5, one can then fill out the form with all the known member data before clicking *Hinzufügen* (add). Afterwards, the user is forwarded to the member overview page where one sees all the recorded members and can perform different actions on them.

Figure A.6 shows the member overview after having added our first member. The second visible member is our test account we use for this tutorial. In a next step, we are going to create the user account for the previously added member by clicking on the user icon. The system automatically assigns a username to the member and displays us a form to set a new password for the member (see figure A.7).

If we need to change the password for a member, we can click on the lock icon in the member overview (see the lock icon at the very back of our test account in figure A.6) which replaces the user icon as soon as the member has an account.

The other icons in the member overview allow the user to view all recorded data about a member (magnifier icon), edit it (pen icon), or delete it (trash icon). The form to edit a member is exactly the same as the one to add the member except that all fields are already filled with the recorded values.

To record the other members of our club, we can proceed as described in the previous paragraphs: first we add the member to the system and then we create the account for it. The credentials (username and password) have to be transmitted to the member on another communication channel (for example by e-mail or text message). After having added all the club members, our member management overview should look as shown in figure A.8.

A.3 Manage the events

In this chapter, we show how the events can be managed over the course of a season. Instead of showing repeatedly the same actions for every event, we will show the workflow in detail for one event. All other events can be managed the same way.

Manage your club!	aliodon/onualtuna - Spor	soronvonvaltung – Evontv	onvoltung – Rondon	A Einstellungen-
	gliederverwaltung + Spor	isorenverwaitung	erwaltung • Penden	zen ¥Enistenungen*
Home / Mitgliederverwaltung	Mitglied hinzufügen			
Neues Mitali	ed hinzufüc	ien		
Persönliche Informati		JCIT		
	onen			
Geschlecht*				
Vorname*		Nachname*		
Peter	~	Gerber		•
Strasse*		Hausnummer*	PLZ*	
Mainstreet	✓	3A 🖌	1234	✓
Ort*		Geburtsdatum (im Format	TT.MM.JJJJ)	
City	✓	12.05.1989		✓
Spielbetrich				
- opieineriien				
Team L Competitive Team	Competitive Team	Lizenznummer		
	competitive ream	120400703		•
Kontaktdaten				
Telefon 1		Telefon 2		
+41 79 123 45 67	•	+41 31 987 65 43		✓
E-Mail 1		E-Mail 2		
Weitere Informationer	1	Status		
Computer Scientist	•	aktiv •		
Ebrenmitalied				
let Ebrenmitglied:	Datum dar Ernannung (im F	Cormat TTMM (111)		
Nein •	Datum der Ernennung (IM F	eben 🗸		
Schiedsrichter				
Ist Schiedsrichter	SR-Lizenznummer		SR-Lohn	
Nein •	Lizenznummer eingeben	✓	Lohn eingeben	✓
SR-Partner		SR-Bemerkungen		
Partner eingeben	•	Bemerkungen eingeben		~
Funktionär				
let Eunktionär	Funktion		let im Vorstand	
Nein •	Funktion eingeben	✓	Nein	T
	~			
Hinzufügen!				

Figure A.5: Form to add a new member to the system.

Mitglie	ederve	erwaltung						s	Suchen			
Nachname	 Vorname 	♦ Strasse ♦	Nr. \$	PLZ 🔶	Ort 🔶	Team 🔶	E-Mail 🔶	Status 🔶	Benutzer 🔶	÷	¢	\$
Berger	Linda	Altes Bauernhaus	1	5689	Land	Fun Team	linda@bauernhof.ch	passiv	Iberger1	۹	1	î (
Gerber	Peter	Mainstreet	ЗA	1234	City	Competitive Team	p.gerber@testmail.ch	aktiv		Q	1	î l
									Zurü	ck	1	Nächst
Mitglied hinz	ufügen											

Figure A.6: Member management overview after having added the first member. (1) Icon to create a system account for the user.

Manage your club! Mitgliederverwaltung - Sp	oonsorenverwaltung - Eventverwaltung -	Pendenzen	🗢 Einstellungen 🗸
Home / Mitgliederverwaltung / Passwort zurücksetze	n		
Account pgerber1 erfolgreich erstellt!			
Passwort von Peter Ge	erber zurücksetzen	1	
Passwort zurücksetzen			
Neues Passwort*	Neues Passwort wiederholen*		
Zurücimetzen!			
Impressum			

Figure A.7: Form to set a new password for the created user.

50

51

0 • Einträge	e anzeigen								Suchen				
Nachname 🔺	Vorname 🝦	Strasse 🔶	Nr. ≑	PLZ 🔶	Ort	Team 🍦	E-Mail 🔶	Status	Benutzer 🔶	×	*	\$	-
Berger	Linda	Altes Bauernhaus	1	5689	Land	Fun Team	linda@bauernhof.ch	aktiv	Iberger1	۹	1	Î	-
Doutaz	Andy	Rue de Gen?e	2	9372	Village	Competitive Team	andydoutaz@mail.ch	aktiv	adoutaz1	۹	/	Ē	8
Gerber	Peter	Mainstreet	3A	1234	City	Competitive Team	p.gerber@testmail.ch	aktiv	pgerber1	Q	1	Î	-
Habegger	Adrian	Hauptstrasse	23D	5648	Stadt	Fun Team	habeggers@mailservice.de	aktiv	ahabegger1	۹	1	â	8
Jenni	Susanne	Musterg?sli	4	4785	Testort	Competitive Team	susanne@jenni.ch	aktiv	sjenni1	Q	1	â	-
Meier	Martin	Dorf	136	2345	Musterort	Competitive Team	martin2001@mail.com	aktiv	mmeier1	Q	1	Ē	-
Meister	Martin	Dorfstrasse	17	6589	Auf dem Land	Fun Team	martin@meister.ch	aktiv	mmeister1	۹	1	â	•
Muster	Max	Mustergasse	1	1234	Dort	Competitive Team	max@muster.ch	aktiv	mmuster1	Q	/	â	•
Müller	Fritz	Dorfstrasse	5A	4433	Stadt	Fun Team	fritz@mueller.ch	aktiv	fmüller1	۹	1	â	•
			05	0000	Stadt	Eup Toom	stefan@stefan.ch	aktiv	srothenhühler1	٩		÷	

Figure A.8: Member overview after having added all members.

A.3.1 Add a season

Before we can start managing events, we need to add a season to our system. We therefore select *Eventverwaltung* \Rightarrow *Saison hinzufügen* (event management \Rightarrow add season) in the navigation bar. In the form that is shown afterwards (figure A.9), we can enter the season title and select all the members that have to help. In our case, all members have to help (and are therefore selected).

Manage ye	DUR Club! Mitgliederver	waltung - Sponsorenverwaltu	ng∓ Eventverwaltung∓	Pendenzen	🛱 Einstellungen -
Home / S	aison hinzufügen				
Saiso	on hinzufüg	en			
Titel					
Tite!*					
Season 20	17/2018	✓			
Mitgliede	r auswählen				
Bitte bei allen	Mitgliedern, welche in diese	r Saison Helferstunden leisten mü	issen, zuvorderst ein Häkchen	setzen.	
	Linda Berger	aktiv	Fun Team	10.08.1985	
	Andy Doutaz	aktiv	Competitive Team	13.12.1955	
•	Peter Gerber	aktiv	Competitive Team	12.05.1989	
	Adrian Habegger	aktiv	Fun Team	15.09.2005	
•	Susanne Jenni	aktiv	Competitive Team	27.12.1985	
•	Martin Meier	aktiv	Competitive Team	05.03.2001	
•	Martin Meister	aktiv	Fun Team	25.06.1995	
•	Max Muster	aktiv	Competitive Team	01.01.1990	
•	Fritz Müller	aktiv	Fun Team	17.03.1965	
•	Stefan Rothenbühler	aktiv	Fun Team	02.07.1992	
	Andrea Stettler	aktiv	Fun Team	19.12.1989	
•	Beat Wegmüller	aktiv	Competitive Team	06.11.2001	
Hinzufügen	1				
Impressum					

Figure A.9: Form to add a new season to the system.

After the successful addition of the season, we are forwarded to the season overview page (figure A.10). This overview provides us with some functions for the season which we will introduce within the next chapters. One of the functions, called *Helfer bearbeiten* (edit helpers), allows users to modify the selection of helpers. The form to edit the helpers is exactly the same as we have seen before while adding the season.

53



Figure A.10: The season overview after having added the season. (1) The available functions for a season.

A.3.2 Manage an event

In this section, we will show what the life cycle for one single event looks like, from the addition till the closing. We will also provide insight into how a club member subscribes for helper services of this event and manages its services.

A.3.2.1 Add the event

Our first step to add the event is to click on the button *Event hinzufügen* (add event) in the season overview (see again figure A.10). In the form that is displayed afterwards (figure A.11), we can enter some general information in the upper part (e.g., the event date, the location, and the title). In the lower part, we can specify line-by-line the different tasks and the number of helpers we need for them (e.g., two referees, one speaker, and four persons for the bistro). We then click *Hinzufügen* (add) to add the event to the system.

After the addition, the application automatically forwards us to the edit event page (figure A.12) where we can add more detailed information for the different helper services. For each service, we can record the start and end time, the member that will do the service (if we already know it), and specify if the service should be visible to all members or not. If a service is not needed any more, we can remove it by clicking on the trash icon at the end of the line.

As soon as the recorded information about the helper services is saved, the system

	Jonsorenverwaltung	🗢 Einstellungen
Home / Übersicht der Saison Season 2017/2018 / Ev	vent hinzufügen	
Neuen Event hinzufüg	an	
Hinweis: Der Anmeldeschluss muss mindestens einen Tag	u vor dem Event liegeni	
Tite!*	Ort*	
Tournament Oktober 🗸	Curling rink 🗸	
Datum (im Format TT.MM.JJJJ)*	Anmeldeschluss (im Format TT.MM.JJJJ)*	
01.10.2017 🖌	30.09.2017 🗸	
Beschreibung		
Monthly tournament		
Benötigte Helfer Aufgabe	Anzahl Helfer	
Benötigte Helfer Aufgabe Bistro	Anzahi Helfer	
Benötigte Helfer Aufgabe Bistro Aufgabe Soeaker	Anzahl Helfer 4 Anzahl Helfer	
Benötigte Helfer Aufgabe Bistro Aufgabe Speaker	Anzahl Helfer 4 Anzahl Helfer 1 Anzahl Helfer	
Benötigte Helfer Aufgabe Bistro Aufgabe Speaker Aufgabe Referee	Anzahl Helfer 4 Anzahl Helfer 1 Anzahl Helfer 2	
Benötigte Helfer Aufgabe Bistro Aufgabe Speaker Aufgabe Referee Aufgabe	Anzahl Helfer 4 Anzahl Helfer 1 Anzahl Helfer 2 Anzahl Helfer	
Benötigte Helfer Aufgabe Bistro Aufgabe Speaker Aufgabe Referee Aufgabe Aufgabe	Anzahl Helfer 4 Anzahl Helfer 1 Anzahl Helfer 2 Anzahl Helfer Anzahl Helfer Anzahl Helfer	
Benötigte Helfer Aufgabe Bistro Aufgabe Speaker Aufgabe Referee Aufgabe Aufgabe Aufgabe	Anzahl Helfer 4 Anzahl Helfer 1 Anzahl Helfer 2 Anzahl Helfer Anzahl Helfer Anzahl Helfer Anzahl Helfer Anzahl Helfer	
Benötigte Helfer Aufgabe Bistro Aufgabe Speaker Aufgabe Referee Aufgabe Aufgabe Aufgabe eingeben Aufgabe Aufgabe	Anzahl Helfer 4 Anzahl Helfer 1 Anzahl Helfer 2 Anzahl Helfer Anzahl Helfer Anzahl Helfer Anzahl eingeben Anzahl eingeben	

Figure A.11: Form to add an event. In the upper part, the general event information (e.g., date, location, title) has to be entered. In the lower part, one can specify line-by-line the different tasks that have to be done and the number of needed helpers for each task.

Manage your club! Mi	itgliederverwaltung - Spo	nsorenverwaltung - Eventv	erwaltung - Pendenzen			💠 Einstellungen -			
Home / Ubersicht der Saise	on Season 2017/2018 / Eve	nt bearbeiten							
Event erfolgreich hinzugefüg	jtl								
Event Tourn	ament Okto	ber bearbeit	ten						
Informationen zum E	vent								
Hinweis: Der Anmeldeschluss i	muss mindestens einen Tag v	or dem Event liegen!							
Titel*		Ort*							
Tournament Oktober		Curling rink							
Datum (im Format TT.MM.JJJ	IJ)*	Anmeldeschluss (im Format TT.MM.JJJJ)*							
01.10.2017		30.09.2017							
Beschreibung									
Monthly tournament									
Helfereinsätze verwa	alten								
Hinweis: Bitte Uhrzeiten im For auswählen, ob der Einsatz öffe	mat HH:MM oder HHMM ang ntlich sichtbar sein soll oder r	jeben. Falls die Startzeit vor de nicht (sichtbar = Häkchen geset	r Endzeit liegt, werden keine Stunden zt)	gutgeschrie	ben. Bitte in de	er hintersten Spalte			
Bistro 1	08:00	12:00	2	•	☑ 3	<u>ش</u>			
Bistro	08:00	12:00		T	e	m			
Bistro	12:00	16:00		T	•	Î			
Bistro	12:00	16:00		Ŧ	•	m			
Speaker	09:00	16:00	Martin Meister (Fun Team)	Ŧ	v	m			
Referee	09:00	12:00		•		â			
Referee	13:00	16:00		T		â			
Helfer hinzufügen									
Aufgabe		Anzahl Helfer							
Aufgabe eingeben		Anzahl eingeben							
Speichern									

Figure A.12: Form to edit an event. For each helper services, one can enter the following information: (1) Start and end time of the service. (2) The member that will do the service (if already known). (3) Specify if the service should be visible to all members (checked means that the item is visible).

displays the season overview. This time, we can see the added event listed up in the section *Anstehende Events* (upcoming events, figure A.13). At the end of the event entry, we have the possibility to edit, delete, or export the event. If one decides to edit the event, the same form as before (figure A.12) will be displayed where the helper services can be adjusted. Exporting the event creates an Excel file which contains all the recorded information about the event.



Figure A.13: The season overview with our recorded event listed up in the section *Anstehende Events* (upcoming events).

A.3.2.2 Manage helper services as a member

Once the event has been added to the system by the board member, all the other members can subscribe to the helper services (if they have an account). We briefly switch the perspective from the board view to a club member's view. When a member has logged into the system, they see an index page (figure A.14), which is different from the one board members see. The navigation bar contains the seasons where the member has to help (in this case only the one season we added before). For each season, the member has two options: either search for helper services or get an overview of the subscribed and fulfilled services.



Figure A.14: The index page for normal members. In the navigation bar, members see all the seasons where they have to help.

Search for helper services

By selecting Season $2017/2018 \Rightarrow$ Helfereinsätze suchen (Season $2017/2018 \Rightarrow$ search for helper services) in the navigation bar, we get forwarded to an overview page with all open helper services (i.e., helper services where no member is registered so far). As shown in figure A.15, only four (of seven) helper services for the event are visible to our member. The reason is simple: the speaker task is already related with a member and the two referee jobs should not be visible to the members according to what we recorded in the system (see again figure A.12).

By clicking on the green button for one of the bistro helper services, the member can subscribe to the corresponding task. On the board side, we can immediately see such a registration in the corresponding event (figure A.16), since the member gets automatically filled out for the selected helper services.

Get overview of the personal helper services

A member can select Season $2017/2018 \Rightarrow$ Meine Helfereinsätze (Season $2017/2018 \Rightarrow$ personal helper services) in the navigation bar which forwards it to an overview of the personal helper services (figure A.17). On this page, we can see the helper service for which we registered before. If we want to cancel it, we can do this by clicking the red button, but this function is only available until the event's registration deadline. Additionally, we can see the number of hours served and planned by the member until today.

A.3.2.3 Close the event

We will now switch back to the board perspective. After the event, the season overview displays the event in the section *Noch nicht abgeschlossene Events* (open events, fig-

Manage your clu	b! Season 2017/2018▼				🗢 Einstellungen -					
Home / Helfereins	sätze suchen									
Offene Helfereinsätze der Saison Season 2017/2018										
Datum	Event	Tätigkeit	Beginn	Ende						
01.10.2017	Tournament Oktober	Bistro	08:00	12:00	Anmelden					
01.10.2017	Tournament Oktober	Bistro	08:00	12:00	Anmelden 1					
01.10.2017	Tournament Oktober	Bistro	12:00	16:00	Anmelden					
01.10.2017	Tournament Oktober	Bistro	12:00	16:00	Anmelden					
Impressum										

Figure A.15: The overview of all open helper services. (1) By clicking the green button, the member can subscribe for the corresponding service.

ure A.18). To close it, one can click on the check mark next to the event.

The form to close an event (figure A.19) is similar to the edit event page. But instead of having the possibility to determine the visibility of a helper service, we can specify whether the members were present or not. This information is important for the crediting of hours: only present members will get the hours credited.

During closure, one can still add new helper services to the event, delete existing ones, and modify them (e.g., change the member). In our example, all members except for one were present at the event. We will see the impact of the non-attendance of *Fritz Müller* soon. After we have recorded all the information, we can click on *Abschliessen* (close) and the system forwards us to the season overview once again. This time, our event appears in the section with the closed events and can therefore only be opened with read access (figure A.20).

A.3.3 Get hours overview

After the closure of the first event, we would like to get an overview of the hours served. This overview can be reached by clicking on the button *Übersicht Helferstunden* (overview helper hours). As shown in figure A.21, *Fritz Müller* did not get any hours credited. All the other members got credited the hours they served at the event. By clicking on the magnifier icon for one of the members, we get a detailed list of all the helper services the member is related to. Additionally, this page provides the board member with the average number of hours helped.

Hinweis: Der Anmeldesch	luss muss mindestens eine	en Tag vor dem Event liegen!				
Titel*		Ort*				
Tournament Oktober		Curling rink				
Datum (im Format TT.MI	*(LLLL)	Anmeldeschluss (im For	mat TT.MM.JJJJ)*			
01.10.2017		30.09.2017				
Beschreibung						
Monthly tournament						
		li li				
Helfereinsätze ve	erwalten					
Hinweis: Bitte Uhrzeiten i	n Format HH:MM oder HH	MM angeben. Falls die Startzeit vor d	ler Endzeit liegt, werden keine Stunden g	utgeschrieben.	Bitte in der hintersten S	Spalte
Hinweis: Bitte Uhrzeiten i auswählen, ob der Einsat	n Format HH:MM oder HH z öffentlich sichtbar sein so	MM angeben. Falls die Startzeit vor d il oder nicht (sichtbar = Häkchen ges	ler Endzeit liegt, werden keine Stunden g etzt)	utgeschrieben.	Bitte in der hintersten	Spalte
Hinweis: Bitte Uhrzeiten i auswählen, ob der Einsat Bistro	m Format HH:MM oder HH: z öffentlich sichtbar sein so 08:00	MM angeben. Falls die Startzeit vor d ill oder nicht (sichtbar = Häkchen ges 12:00	ter Endzeit liegt, werden keine Stunden g etzt)	utgeschrieben. ▼	Bitte in der hintersten S	Spalte
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro	m Format HH:MM oder HH z öffentlich sichtbar sein so 08:00	MM angeben. Falls die Startzeit vor d II oder nicht (sichtbar = Häkchen ges 12:00	fer Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team	uutgeschrieben.	Bitte in der hintersten S	Spalte
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro	m Format HH:MM oder HH z öffentlich sichtbar sein so 08:00 08:00	MM angeben. Falls die Startzeit vor d II oder nicht (sichtbar = Häkchen ges 12:00	ler Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team)	utgeschrieben.	Bitte in der hintersten S	Spalte
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro Bistro	m Format HH:MM oder HHi z öffentlich sichtbar sein so 08:00 08:00 12:00	MM angeben. Falls die Startzeit vor d ill oder nicht (sichtbar = Häkchen ges 12:00 12:00 16:00	ler Endzeit liegt, werden keine Stunden (etzt) Martin Meier (Competitive Team)	utgeschrieben.	Bitte in der hintersten S	Spalte
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro Bistro Bistro	n Format HH:MM oder HHi z öffentlich sichtbar sein so 08:00 08:00 12:00	MM angeben. Falls die Startzeit vor o il oder nicht (sichtbar = Häkchen ges 12:00 12:00 16:00 16:00	ler Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team)	uutgeschrieben.	Bitte in der hintersten 5 m m m m	Spalte
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro Bistro Bistro	m Format HH:MM oder HHI z öffentlich sichtbar sein so 08:00 08:00 12:00 12:00	MM angeben. Falls die Startzeit vor d il oder nicht (sichtbar = Häkchen ges 12:00 12:00 16:00 16:00	ler Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team)	vutgeschrieben.	Bitte in der hintersten 5	Spalte
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro Bistro Bistro Speaker	m Format HH: MM oder HHI z öffentlich sichtbar sein so 08:00 08:00 12:00 12:00 09:00	MM angeben. Falls die Startzeit vor d II oder nicht (sichtbar = Häkchen ges 12:00 12:00 16:00 16:00 16:00	ter Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team) Martin Meister (Fun Team)	vutgeschrieben.	Bitte in der hintersten S	Spaite
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro Bistro Bistro Speaker Referee	m Format HH: MM oder HHI z öffentlich sichtbar sein so 08:00 12:00 12:00 09:00 09:00	MM angeben. Falls die Startzeit vor d II oder nicht (sichtbar = Häkchen ges 12:00 12:00 16:00 16:00 12:00 12:00	ter Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team) Martin Meister (Fun Team)	vutgeschrieben.	Bitte in der hintersten 3	Spalte
Hinweis: Bitte Uhrzeiten la uuswählen, ob der Einsat Bistro Bistro Bistro Speaker Referee Referee	n Format HH:MM oder HHi z öffentlich sichtbar sein so 08:00 12:00 12:00 09:00 09:00	MM angeben. Falls die Startzeit vor d 11 oder nicht (sichtbar = Häkchen ges 12:00 16:00 16:00 16:00 12:00 10:00 1	er Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team) Martin Meister (Fun Team)	vutgeschrieben.	Bitte in der hintersten S	Spalte
Hinweis: Bitte Uhrzeiten la uuswählen, ob der Einsat Bistro Bistro Bistro Bistro Speaker Referee Referee	n Format HH:MM oder HHi z öffentlich sichtbar sein so 08:00 12:00 12:00 09:00 09:00 13:00	MM angeben. Falls die Startzeit vor d 11 oder nicht (sichtbar = Häkchen ges 12:00 16:00 16:00 16:00 16:00 16:00 16:00 16:00	ter Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team) Martin Meister (Fun Team)	vutgeschrieben.	Bitte in der hintersten 5	Spalte
Hinweis: Bitte Uhrzeiten la auswählen, ob der Einsat Bistro Bistro Bistro Bistro Speaker Referee Referee Helfer hinzufüger	n Format HH: MM oder HHi z öffentlich sichtbar sein so 08:00 12:00 12:00 09:00 09:00 13:00	MM angeben. Falls die Startzeit vor d il oder nicht (sichtbar = Häkchen ges 12:00 12:00 16:00 16:00 16:00 16:00 16:00 16:00 16:00 16:00	ler Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team) Martin Meister (Fun Team)	vutgeschrieben.	Bitte in der hintersten 3	Spaite
Hinweis: Bitte Uhrzeiten li auswählen, ob der Einsat Bistro Bistro Bistro Speaker Referee Referee Helfer hinzufüger Aufoabe	n Format HH:MM oder HHi z öffentlich sichtbar sein so 08:00 08:00 12:00 12:00 09:00 09:00 13:00	MM angeben. Falls die Startzeit vor d il oder nicht (sichtbar = Häkchen ges 12:00 12:00 16:00 16:00 16:00 16:00 16:00 16:00 16:00 16:00	ter Endzeit liegt, werden keine Stunden g etzt) Martin Meier (Competitive Team) Martin Meister (Fun Team)	vutgeschrieben.	Bitte in der hintersten 3 m m m m m m m m m m m m m	Spalte

Figure A.16: The edit event page of our event (board view). In the red square, we can see the registration we did with the member account.

Manage your c	lub! Season 2017/2018 →						🖨 Einstellungen				
Home / Meine I	Helfereinsätze										
Meine Helfereinsätze der Saison Season 2017/2018											
Datum	Event	Tätigkeit	Beginn	Ende	h	Status					
01.10.2017	Tournament Oktober	Bistro	08:00	12:00	4.0	Geplant	Abmelden 1				
Stunden geholfen: Stunden geplant: 4	0 .0 2										
Impressum											

Figure A.17: The helper service overview for the member. (1) Button to cancel the registration. (2) Overview of the served and planned hours for this season.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung-	Eventverwaltu	ung - Pendenzen			¢	Einstellungen▼
Home / Übersicht der	Salson Season 2017/2018							
Saison Se Verfügbare Aktion Event hinzufügen	eason 2017 nen letter bearbeiten Übersi	2/2018	on abschliessen	Alle Events exportieren				
Anstehende Keine zukünftigen Events	erfasst.							
Vergangene	Events							
Noch nicht abg	eschlossene Ev	ents						
Datum	Titel			Ort				
01.10.2017	Tournament Ok	tober		Curling rink	1	×	â	0
Abgeschlossen Keine abgeschlossenen E	ie Events							
Impressum								

Figure A.18: After the event, the system displays it in the section *Noch nicht abgeschlossene Events* (open events). It can be closed by clicking on the check mark.

nform	nationen zum E	vent					
	Titel To Datum 01 Ort Cu	urnament Oktober .10.2017 Irling rink					
Helfer	reinsätze verwa	lten					
linweis: uswähl	Bitte Uhrzeiten im For en, ob das Mitglied anv	mat HH:MM oder HHMM an vesend war oder nicht (anw	igeben. Falls die S esend = Häkchen	Startzeit vor der Endzeit gesetzt).	t liegt, werd	den keine Stunden gutgeschrieben. Bitte in	der vordersten Spalte
0	Bistro	08:00		12:00		Andy Doutaz (Competitive Team)	â
9	Bistro	08:00		12:00		Martin Meier (Competitive Team)	â
	Bistro	12:00		16:00		Andy Doutaz (Competitive Team)	â
	Bistro	12:00		16:00		Fritz Müller (Fun Team)	Ê
9	Speaker	09:00		16:00		Martin Meister (Fun Team)	â
9	Referee	09:00		12:00		Beat Wegmüller (Competitive Tean 🔹	â
	Referee	13:00		16:00		Peter Gerber (Competitive Team)	Ē
lelfer	r hinzufügen						
ufaaba	•		Anzahl Helfe	r			

Figure A.19: The form to close an event. In the first column, one can specify if the members were present or not. Only members that have helped will get the hours credited.

62



Figure A.20: The season overview after the event has been closed. The event therefore appears in the section *Abgeschlossene Events* (closed events).

Helferübersi	cht					
Grobübersicht						
Anzahl Mitglieder: 12 Anzahl Stunden: 23 Durchschnitt: 2,	2 5.0 08					
Übersicht über die M	litglieder					
0 • Einträge anzeigen					Suchen	
Name	Geburtsdatum	Team	Status	h geleistet	🔶 h geplant	÷ ÷
Andy Doutaz	13.12.1955	Competitive Team	aktiv	8.0	0.0	۹
Peter Gerber	12.05.1989	Competitive Team	aktiv	3.0	0.0	Q
Susanne Jenni	27.12.1985	Competitive Team	aktiv	0.0	0.0	Q
Martin Meier	05.03.2001	Competitive Team	aktiv	4.0	0.0	Q
Max Muster	01.01.1990	Competitive Team	aktiv	0.0	0.0	Q
Beat Wegmüller	06.11.2001	Competitive Team	aktiv	3.0	0.0	Q
Linda Berger	10.08.1985	Fun Team	aktiv	0.0	0.0	Q
Adrian Habegger	15.09.2005	Fun Team	aktiv	0.0	0.0	۹
Martin Meister	25.06.1995	Fun Team	aktiv	7.0	0.0	٩
Fritz Müller	17.03.1965	Fun Team	aktiv	0.0	0.0	۹
				_	Zurück 1 2	Nächste

Figure A.21: The overview of the current hours. The red square shows that the member that was not present (*Fritz Müller*) did not get any hours credited.

A.3.4 Manage the events throughout the season

Our tutorial to demonstrate the life cycle of an event just used one event in the season overview. Of course, the system allows users to manage several events at the same time throughout the course of a season. One example, how the season overview could look like at an arbitrary point of the season, is shown in figure A.22. At that point, the season contains three upcoming as well as one open and one closed event.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung -	Eventverwaltung] ▼ Pende	enzen				¢∈	nstellungen -
Home / Übersight der :	Saison Season 2017/2018									
Saison Se Verfügbare Aktion Event hinzufügen	ason 2017 Ien Elfer bearbeiten Übersie	/2018	n abschliessen	Alle Events	exportieren					
Anstehende	Events									
Datum	Anmeldeschluss	Titel			0	rt				
08.12.2017	01.12.2017	Tourname	nt December		C	Curling rink			ŵ	0
05.01.2018	20.12.2017	Tourname	nt January		C	Curling rink			ŵ	0
25.03.2018	10.03.2018	Local mar	ket		М	arket place		1	â	0
Vergangene	Events									
Noch nicht abg	eschlossene Eve	ents								
Datum	Titel		Ort							
03.10.2017	Local market	t	Market place			1	×	â		0
Abgeschlossen	e Events									
Datum	Titel				Ort					
01.10.2017	Tourname	nt Oktober			Curling rink			۹		0
Impressum										

Figure A.22: The season overview at an arbitrary point during the season. Currently, three upcoming as well as two past events are recorded.

A.3.5 Close the season

At the end of the season, as soon as all events have been closed, we can close the season as well by clicking the button *Saison abschliessen* (close season) in the season overview. We then have to confirm in a second step that we really want to close the season (figure A.23).



Figure A.23: The dialog to close a season.

The closure of a season ensures that no more changes can be done to the season (and the events of the season). If we want to access a closed season, we can navigate to *Eventverwaltung* \Rightarrow *Abgeschlossene Saisons* (event management \Rightarrow closed seasons). The system will then provide us with an overview of all closed seasons (figure A.24). We can open the season overview for a past season by clicking on the magnifier icon. The season overview will then allow us to read all the data, but we cannot record any new information.

Manage your club!	Mitgliederverwaltung -	Sponsorenverwaltung -	Eventverwaltung -	Pendenzen		🗢 Einstellungen -
Home / Übersicht der vergangenen Salsons						
Übersicht der vergangenen Saisons						
Titel	-	-				
Season 2017/2018					۹	
Impressum						

Figure A.24: The overview of all the closed seasons. By clicking on the icon at the end of a season, one can open the season overview.

A.4 Conclusion

We have covered the main features of the website within this tutorial. We started with the member management and went through creation, maintenance, and closure of a full season of a sports club afterwards.