JobChallenge information system

Master’s Thesis

Bc. Tomáš Pouzar

Brno, Spring 2018
Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Bc. Tomáš Pouzar

**Advisor:** Mgr. Pavel Budík
Acknowledgement

First, I would like to thank my thesis advisor Mgr. Pavel Budík and consultant Mgr. Bc. Ladislav Říšák. They consistently helped whenever I needed it.

Secondly, I wish to express my sincere thanks to all JobChallenge organizers, with whom the cooperation was very pleasant during the whole research period.

I would also like to acknowledge RNDr. Michal Batko, Ph.D. as the reader of this thesis, and I am grateful for every valuable comment on this thesis.

Finally, I must express my gratitude to my parents for their support and continuous encouragement throughout my years of study and through the process of writing this thesis. This accomplishment would not have been possible without them. Thank you.
Abstract

The JobChallenge fair already provides an information system including a web application which allows people to manage their accounts, registrations and orders. The system also includes an administration for organizers to manage accounts and orders. Unfortunately, there are several flaws: the website contains visible errors, the workflow is often clumsy, and the administration of the whole system is highly complicated. English versions of individual pages often do not work and the registration process does not follow the basic guidelines of modern user experience. Moreover, the information system is split into two different CMSs which is highly inefficient, and both the code and the database management is very slow due to a poor design. Overall, a redesign of this information system would significantly help both participants of the job fair and the JobChallenge organizers.

This thesis aims to improve the JobChallenge information system so that it is easier to manage for organizers and more pleasant to use for users. This goal will be achieved by first analyzing the old system, secondly by designing and implementing a new system. The result of this work is a modified information system that consists of a web application which is improved with friendlier user experience and simplified back-end management. Both can significantly reduce the occurrence of errors on the website and during data administration. Further development is recommended especially in the area of web content management which was not the primary purpose of this thesis but still crucial for effective work with the system.
Keywords

JobChallenge, job fair, information system, web application, Umbraco
# Contents

**Introduction** 1

1 Information system development 4
   1.1 System development 4
      1.1.1 Methodologies overview 4
      1.1.2 Analysis 8
      1.1.3 Planning 9
   1.2 Security 12

2 Planning 15
   2.1 JobChallenge Job Fair 15
   2.2 Old system analysis 16
      2.2.1 Overall functionality 18
      2.2.2 Front-end 22
      2.2.3 Back-end 24
      2.2.4 Database 26
      2.2.5 Security 27
   2.3 New system requirements 31

3 Methodology 36
   3.1 Design of the new system 36
      3.1.1 Web application 36
      3.1.2 Umbraco administration 41
      3.1.3 Database 43
      3.1.4 Back-end 43
   3.2 Technology 51
   3.3 Security measures 52
   3.4 Result 53

4 Conclusion 55

Bibliography 56

A Content of the CD 58
List of Figures

1.1 The difference between a plan-driven and an agile approach to system development 5
1.2 Waterfall development framework model 6
1.3 Incremental model development framework 7
1.4 Spiral model development framework 8
1.5 Scrum, an agile framework for managing work [5] 9
2.1 Architecture of the first version of JobChallenge information system 17
2.2 Architecture of the JobChallenge information system after the migration into the new version of Umbraco 18
2.3 UML Use Case Diagrams shows who can access which part of the information system and what they can do 20
2.4 A screenshot of an Umbraco 6 orders administration as an example of a clumsy design – especially the “Delete” link next to the .pdf icon or the cramped columns. 24
2.5 Diagram of the old back-end layout 25
2.6 Entity–relationship model of the old JobChallenge database 28
3.1 Design of the old order process 38
3.2 Design of the new order process 38
3.3 A draft, drawn in online tool Draw.io, showing one step of the ordering process. 39
3.4 Example of the old unclear design for the stand type selection 40
3.5 Result of the new design for the stand type selection 41
3.6 Old administration structure where each box represents a single page and the list below its functionality 42
3.7 New administration structure where each box represents a single page, the subheadings marks separate tabs and the lists below their functionality 43
3.8 An example of a draft made for administration of orders in Umbraco 44
3.9 Early functional design for administration of orders in Umbraco 44
3.10 Part 1 of the new database containing tables related to exhibitors 45
3.11 Part 2 of the new database containing tables related to orders 46
3.12 Part 3 of the new database containing tables related to visitors 47
3.13 Part 4 of the new database containing tables related to talk shows 48
3.14 Overview of the new back-end architecture 49
3.15 A brief overview of the Angular 5 application used within the web application 50
3.16 Illustration of server-level data processing 51
Introduction

The JobChallenge is a job fair for students and graduates and it is being held every year since 2007 and supported by Masaryk University. The job fair is designed for both students and graduates who want to find a job, but also for employers looking for young talents. A wide range of companies sign up every year as exhibitors and they are looking for potential employees among thousands of visitors. All participants have an information system at their disposal that allows them to create and manage accounts, submit and review CVs, order services at the fair and many others. It is clear that information system for such business event is essential and it is vital for it to operate as smoothly and flawlessly as possible; and since the job fair involves personal data and money, the safety of information is necessary.

Unfortunately, due to a prolonged and complicated development, there are many flaws in the current version of the JobChallenge information system. It consists of a web application for participants and an administration part built into a content management system (CMS) called Umbraco CMS (Umbraco), all hosted by Institute of Computer Science (ICS) of Masaryk University, specifically by Information Systems Division. The CMS provides tools for editing web content and managing user accounts. Nevertheless, most of the system including user interface, business logic and database was designed and implemented by an external company. All the parts were then put together and merged into Umbraco by WebCentrum which is a project managed by ICS. To make it even more complicated, ICS started using a new version of Umbraco (Version 7) and the information system split, thus the content management of the web application moved to the new version, but administration and database stayed in the old version.

One of the main objectives of the work is to merge the functionality of both CMSs into the newer version of Umbraco, so the web content editing and user administration are at one place, and the organizers do not have to access two different systems. Moving the administration part involves designing a new user interface and workflow that is consistent with the new version of Umbraco. The user data administration is heavily connected to the database model that has several data
integrity issues and table relations are not clear, which often leads to serious problems with querying and reporting from the database. Therefore, the database model needs major revision, which is closely associated with a new design of the account registration and the reservation of the exhibition space. Throughout the reservation process, there are many items available for ordering containing different texts or prices that vary over the years. Unfortunately, values of these items are hard-coded at several locations across the entire program code, and therefore are not editable by the organizers but only by developers at ICS which significantly slows down the process of yearly updates. So not only that we have to redesign user interface and database but also the business logic; all details will be addressed later in this thesis.

In the next chapter 1, there is a summary of several different areas of web development in accordance with the current best practices. Furthermore, the chapter gives details on techniques I followed during the development and which are based on modern literature. For the most part, the chapter covers security and describes the basics of risk assessment, threat and vulnerability analysis and security methods that prevent potential system attacks.

The following chapter 2, called Planning, describes JobChallenge Job Fair and its current information system in more detail. The chapter is mostly focused on detailed study of individual parts of the information system including user experience evaluation, database and security analysis. A portion of the chapter covers the CMS system Umbraco mentioned earlier and it is thoroughly explained how the old JobChallenge information system was split into two separate CMSs. This chapter is further largely dedicated to the JobChallenge organizers’ requirements and visions of the new system. These are further elaborated in more detail in the next chapter.

The last chapter 3 illustrates the methodology of my work – e.g. the process of designing the new system and the following development work. All of this is consistent with the practices mentioned in the chapter Web development. I evaluate the security measures I took into consideration and how they are implemented. The rest of the chapter describes the maintenance of the information system and how the development progresses.

The chapter 3 ends with section 3.4 which presents the final results of my work and demonstrates the changes and additions to the in-
formation system. Finally, I discuss how the new information system improved the JobChallenge Job Fair and what are the possibilities of future development.
1 Information system development

Development of an information system is a lengthy and complicated process. Therefore, it is necessary that the development life-cycle is well structured and organized. Before I describe how we developed the system itself, this chapter goes through the basic principles of software development, web development and security with respect to our scenario.

1.1 System development

Software products are often custom-built and made to fit certain customers’ requirements, especially, when it comes to web applications. Such software products have to meet four basic requirements [1]:

1. **Usability** – The software should have a reasonable user interface and appropriate documentation.

2. **Reliability** – The product should not cause physical or economic damage be in case of a failure.

3. **Efficiency** – The application should not waste system resources (memory, time, processor, etc.).

4. **Maintainability** – The software should be developed according to changing customer needs.

In the next subsection, I describe some basic system development life cycles which help to achieve these system requirements.

1.1.1 Methodologies overview

System development life cycle (SDLC) is a term used in software engineering to describe a process of an information system development. SDLC describes several different stages involved in the project making. These stages – like analysis, planning or design – define a work flow for both developers and customers. There are two main approaches to system development [2]:
1. **Plan-driven** – This approach tries to eliminate process and product variability. Development is phase-based, sequential and highly predictive. Overall, this approach assumes developers have all of the correct information up front, but they have to estimate the requirements and plans as it is shown in figure 1.1.

2. **Agile** – In contrast, Agile development is more suitable for those projects where properties cannot be determined beforehand. Therefore, the development should be iterative and incremental, and it should continually adapt to customers’ changing requirements. For web development, where time and resources are limited, but features are flexible, an agile approach is more suitable. The difference between these approaches is clearly shown in figure 1.1.

![Figure 1.1: The difference between a plan-driven and an agile approach to system development](image)

Among plan-driven methodologies belong well-known models like Waterfall model, Incremental model or Spiral model:

1. **Waterfall model** – This model has a simple linear structure of tasks, where each one should be completed before the next one. This visualized in figure 1.2.

2. **Incremental model** – The incremental model offers a more dynamic approach. Within this model, specification, development and validation run concurrently and together they continuously
create iterations. Each iteration is a life cycle on its own (for example, a waterfall model). Figure 1.3 clearly describes the workflow inside the incremental model.

3. **Spiral model** – The last life-cycle model is based on the waterfall model, changing its phases into many pieces of iteration processes to reduce the risk of a project. The principle of the spiral model is shown in figure 1.4.

Figure 1.2: Waterfall development framework model

As mentioned before, next to plan-drive models, there are agile life-cycles, which put the emphasis more on the customer than the long-term planning. Such approach is much more suitable when it comes to information system or web development. Among the best known methods belong the following:

1. **Scrum** – The first agile framework defines an iterative and incremental development strategy for a development team. A basic principle of Scrum is anticipation of customers or developers changing requirements during the process. Figure 1.5 shows the workflow of Scrum in practice. Every increment of the final software is called Sprint and during this time, the development
1. Information system development

![Incremental model development framework](image)

Figure 1.3: Incremental model development framework

The development team implements a portion of the product’s backlog (a set of features). The development team regularly meets in short periods of time (every 24 hours) to discuss the progress. After each Sprint, the team presents its results to customers and either continue to work on the current increment or proceed to next portion of Backlog. [3]

2. WebML – Web modeling language (WebML) is both a methodology and a modeling framework for designing web applications. WebML has several models which are developed in an iterative process. [4]

(a) **Structural model** – Describes the data content of the web application using well known UML diagrams.

(b) **Hypertext model** – Specifies every view of the final website and navigation between them using basic wireframe models

(c) **Presentation model** – Transforms Hypertext models into graphically precise designs.

(d) **Personalization model** – Defines presentation style depending on different profile data.

Since the work on the information system for JobChallenge was heavily influenced by frequent changes of requirements and from the
most part, we were developing a web application, we used a combination of both mentioned agile methodologies. Scrum very accurately describes our workflow, which is described in the next chapter 2. And WebML inspired us to use several design tools to help us during the process described in chapter 3.

Moreover, all agile methodologies are especially designed to synchronize more people in developer teams. Due to an individual nature of my work, I did not choose to implement one methodology rigidly, but I chose the most suitable parts of multiple approaches.

All mentioned development models have common set of partial tasks and most of them consist of several design choices. Following subsections focus on different ways to approach the tasks like use of design tools, choice of technologies, etc.

### 1.1.2 Analysis

Every development of an information system or a web application should start with a proper analysis. It is necessary to set the main purpose and partial goals of the application. The next important thing is to identify the target audience. It helps to design a suitable presentation of the whole project. The analysis can be summarized in three steps [6]:

![Spiral model development framework](image)
1. **Information system development**

![Figure 1.5: Scrum, an agile framework for managing work](image)

1. **Collection of facts** – Requirements are obtained through documentation, interviews and observation.

2. **Scrutiny of the existing system (not necessarily software)** – The current system is analyzed for pros and cons in order to promote the pros and avoid the cons in the new system.

3. **Analyzing the proposed system** – The result of the previous steps is used to prepare the final specifications.

There are several UML (Unified Modeling Language) diagrams helping developers to create an idea on how to focus their work [7]:

1. **Use case diagram** – A use case is a list of actions available in an application. Moreover, each action is assigned to a specific user role. It is a valuable requirement analysis technique widely used in modern software engineering.

2. **Activity diagram** – Activity diagram describes each action of use case diagrams in detail. It is a flowchart representing a flow from one step to another.

### 1.1.3 Planning

Once the idea of the final product is clear, developers must choose among many varying technologies for each part of the system. Since our information system is built upon a CMS and it is partially a web
application, the following selection of technologies is closely related to web development.

At first, developers must choose the most suitable CMS for their project. Using CMS is very beneficial when developing a web based information system. Among the most significant advantages of using a CMS belongs [8][9]:

1. *Simplicity* – Not every user is technically competent to manage web applications, but with CMS the management is noticeably more convenient.

2. *Multiple users* – Every information system consists of numerous accounts and types of users. Designing and developing such a system is possible but it is an unnecessarily demanding process. On the other hand, CMS directly supports account management, authentication and authorization.

3. *Content management* – CMS is not only about editing content of websites, but it offers sophisticated content management. For example, the system automatically removes any links to removed content. Most CMSs also include a basic SEO to make sure its websites rank high in search engines.

4. *Publishing* – One of the most significant advantages of using any CMS is the possibility to publish content whenever the user wants. Users can freely edit the content of websites and leave the changes saved but unpublished.

5. *Complete control* – Overall, an open source CMS provides users almost complete control of their system without the need to often contact developers.

Nowadays, there are many available CMSs on the market [10]:

1. *WordPress* – The most-used open source CMS worldwide with a broad user community. This CMS requires minimal effort for installation and setup. Hence, WordPress is suitable for small and simply structured web applications, especially for blogs. Every server with PHP support is suitable for WordPress, but the most recommended is Apache server [11].
2. Joomla – A CMS more challenging to use than WordPress is Joomla. The system is entirely object-oriented based on an MVC framework. This approach supports development of own extensions and sharing them with the community. Again, Joomla is running on Apache servers, and its middleware is written in PHP [12].

3. Drupal – Originally developed as a student community solution, but now a fully available open source CMS. Drupal focuses on social publishing and small community projects. Therefore, it is most suitable for small or medium-sized projects. Like WordPress and Joomla, Drupal’s middleware is based on PHP. Hence, a web server running Drupal has to support PHP [13].

4. Umbraco – Unlike all previous, Umbraco is primarily written in C#, stores data in Microsoft SQL Server and works on Microsoft IIS. Umbraco is an open source platform, and it is suitable for both small and large-scale projects [14].

The choice of CMS is closely tied to the selection of technologies. As mentioned before, each CMS can use a variety of different technologies. Most of the popular CMSs run on Apache servers which support middleware written in both Java or PHP. Usually, applications running on Apache servers are using Postgres or MySQL database servers. However, as mentioned above, WebCentrum is currently using Umbraco CMS to run all its web applications. Therefore, we decided to implement the information system using Umbraco and its Microsoft technologies.

In contrast with server side, the client side of the application provides more freedom in technology selection. On the one hand, there are many well-known and established technologies like HTML, CSS or JavaScript, but on the other side, there are several new JavaScript frameworks like ReactJS, Vue.js, Ember.js or Angular, which can significantly help in web development.

Next to these JavaScript frameworks, there is also TypeScript, an open-source programming language developed by Microsoft. TypeScript is a strict syntactical superset of JavaScript. Moreover, TypeScript’s compiler is written in TypeScript itself and it compiles to JavaScript. Unlike JavaScript alone or jQuery, TypeScript is designed
for development of both small and large applications. There is one framework that heavily relies on TypeScript and it is called Angular.

Angular is a TypeScript based platform and it is a rewrite of AngularJS. Since Umbraco is based on AngularJS and the company behind Umbraco plans to support Angular in future, we used Angular as well as a tool to build a dynamic and responsive front-end.

1.2 Security

Every information system handles susceptible data like users’ passwords, addresses, billing information or more. Moreover, the JobChallenge information system deals with users’ CVs where security is necessary. Overall, information security is about ensuring the following principles [15]:

1. **Confidentiality** – Only authorized people should be able to access information.

2. **Integrity** – All data should be protected from improper or unauthorized modification.

3. **Availability** – Information is accessible when needed (by authorized people).

In practice, 100% security is unachievable. However, information security is not about preventing loss completely, it is about managing it [16]. The equation below (1.1) shows a simplified view on a risk management with regard to danger and threats. Threat stands for an entity that can harm the organization or the information system. The vulnerability is a weakness that allows a threat to exploit the system. Countermeasures are precautions taken by developers to reduce vulnerability. Finally, the value measures the loss that can be experienced [17].

\[
\text{Risk} = \left( \frac{\text{Threat} \cdot \text{Vulnerability}}{\text{Countermeasures}} \right) \cdot \text{Value} \quad (1.1)
\]

Although risk or loss cannot be removed completely, risk management tries to optimize it to the minimum. Since there is never enough
1. Information system development

time or money to accomplish everything, we have to decide which
level of risk is acceptable using risk assessment. It helps us to iden-
tify values, threats and countermeasures. Information systems are
exposed to several kinds of threats and vulnerabilities:

1. Exposure of sensitive data – One of the most important things is not
to store any sensitive data, mostly passwords, just in plain text.
To make passwords unreadable but still usable for validation it
is possible to use one of the one-way hash functions (e.g. SHA-
1). There is a salted version of the hash function which adds a
specific string to the password before the hashing itself, to make
the encryption even more secure. Even if the database should
get exposed, with this technique, the data will be useless to the
attacker.

2. Denial of Service (DoS) – The primary goal of a DoS attack is to
make the information system unusable, that can be achieved
by overloading or manipulating systems. General methods to
defend against DoS are to offer services only to authorized users
or redirect load from server side to client side.

3. Eavesdropping – Any information system transfers a lot of sensi-
tive data like names, addresses but mostly passwords. In order
to avoid eavesdropping on this information, it is necessary to
make all communication between client and server encrypted.
Therefore replace insecure HTTP protocol with HTTPs protocol.

4. Brute-force attacks – One of the fundamental form of attacks – try-
ing every possible combination that could make up a password
or passphrase – is called brute-force attack. There are several
ways how to prevent this kind of attack. At first, it is recom-
manded to demand a secure password from users and a captcha
validation. Another efficient protection is to limit unsuccessful
login attempts or to add a short time-out before next possible
login.

5. Code injection – A more sophisticated form of attack is code injec-
tion or so-called SQL injection. It is used to bypass login valida-
tion or directly manipulate with database. To prevent such kind
1. Information system development

of attacks, it is usually enough not to accept any SQL queries by plain URL request. Moreover, most programming languages support parametrized SQL queries which makes SQL injection practically impossible.

6. Cross-site scripting – One of the most well-known vulnerabilities is called cross-site scripting (XSS). It takes the advantage of the possibility of modifying and running code on the client side. Consequences of such attacks can be disabling client-side validation of form inputs or manipulating with the price when it comes to the shopping cart. To prevent XSS, next to client-side validation, there should be also a server-side validation of any sensitive information.

Section 3.3 briefly describes how we approached the security in our information system and how we integrated countermeasures for each of the mentioned threats and vulnerabilities.

Now that I have described some of the most relevant parts of web development including methodologies, choice of technologies and security, we can proceed to the development of JobChallenge information system itself.
2 Planning

Before we look at the process itself, I would like to introduce the reader to the JobChallenge Job Fair and provide some context to further reading. The first section of this chapter describes the job fair event, its history and evolution throughout the years, how it looks nowadays and what are the requirements for the new version of the information system.

2.1 JobChallenge Job Fair

JobChallenge was first held in the year 2007 and it was attended by more than 30 companies and 1,000 students. For the first few years, Masaryk University was the only organizer. But since 2011, Mendel University in Brno and Brno University of Technology got involved as well. As a result, the number of exhibitors and visitors rose more than double. Over the course of the following years, JobChallenge has evolved into a modern and fast growing job fair that provides space for more than one hundred exhibitors and thousands of visitors. Nowadays, JobChallenge Job Fair takes place annually at the Brno Exhibition Center (BVV) on a single day in late November. The Job Fair is mostly intended for students and graduates who want to increase awareness of the labor market, develop soft skills and get in touch with most sought after companies. The visitors have a unique opportunity to present their experience and offer their CVs to many employers in a very short time. On the other hand, the companies and their employers can readily introduce themselves to thousands of visitors and look for talented individuals among them. Each exhibitor has a presentation stand (either his own or provided by JobChallenge organizers) where visitors can stop by and discuss their expertise and possible job offers. Moreover, they have access to the database of visitors CVs, so they can easily stay in touch with them.

In addition to this free space filled with stands of various exhibitors, which every passer-by can visit, there is a vast variety of accompanying programme. One of the possibilities is to visit some of the talk shows, which are led by experienced moderators who debate with the staff of particular companies. The talk shows are designed for companies
to present their corporate culture, salaries of their employees and for visitors to hear a career story of some of the companies’ staff. In order for visitor to visit a talk show, he or she has to register in advance using the web application.

Moreover, JobChallenge provides a whole range of workshops which aim to help the visitors find a new job quickly and easily. One of those workshops is a CV coaching where professional consultants help the attendants improve their CVs, so they have a better chance at the job market. CVs can be uploaded into the information system via the web application so the exhibitors can access them in advance before the job fair.

Overall JobChallenge Job Fair is a fast growing project which helps thousands of young people find better jobs, so the quality and stability of its information system is utmost important. In the next few pages, I explain how the old system was developed, what are its main flaws and what has to be done to fix these flaws.

2.2 Old system analysis

JobChallenge is accompanied by an information system from the very beginning. At first, the web application (including a website, a business logic and a database) was implemented by an external company Icontio [18], and it was then integrated into CMS (Umbraco version 4 at the time) and hosted by WebCentrum. The architecture of this system is shown in figure 2.1. At the time, Umbraco allowed administrators to simply edit web content, administrate user accounts of registered users and manage orders.

Later, JobChallenge designed utterly new visuals for their web application and they wanted to replace the old design as soon as possible. At the time, a new version of Umbraco has been released and it introduced a much better system for web content management which was more suitable for the new design. So, WebCentrum decided to migrate the web application into the newer version of Umbraco. Unfortunately, due to several reasons the information system did not migrate to the newer version as a whole but got split and part of the system remained in the older version. It was mostly due to a time limit
before the next year of the job fair, so only a part of the migration was finished.

Figure 2.2 shows the architecture after migration into Umbraco version 7 where is visible the division of functionality into two separate subsystems:

**Umbraco version 6** In the older version there remained the whole administration of accounts and orders and some parts of the business logic including the back end of the administration and parts of the web application like account registration, account management and the order process.

**Umbraco version 7** With more options to edit web content, this is where the rest of the application moved. Next to the web content management the newer Umbraco also implements an exhibitors archive, a blog, JobChallenge partners management and more.
2. Planning

One of the main flaws in such design is the need to access both versions of Umbraco in order to manage accounts or edit the web content which complicates everyday work. Another major complication is the use of multiple databases and their design which will be addressed later in subsection 2.2.4.

2.2.1 Overall functionality

There are three different types of users of the information system:

Visitor A student or a graduate who can visit the job fair as a registered or an unregistered visitor looking for job opportunities.
Exhibitor  A company or a non-profit organization which has to register and orders a place at the job fair in order to present itself to visitors.

Administrator  An organizer or an authorized person who has access to CMS to edit web content or to administrate accounts of visitors and exhibitors.

The summary of their possible actions is shown in a use case diagram displayed in figure 2.3. All types of users have a wide range of options:

1. Visitor
   - Manage account – Every visitor can create an account for free which contains their personal data, information about studies, language skills, and work specialization. Visitors can freely access and edit their accounts but only until the end of the job fair. After that, all visitor accounts are deleted and in order to sign up for a next year, they have to register again.
   - Upload CV – One of the benefits of account is that visitors can upload their CVs in Czech or English and present them to exhibitors.
   - Add exhibitor to favorites – Visitors can also go through all signed-up companies and add them into a list of favorites.
   - Talk show registration – Having an account also allows visitors to sign up for exhibitors presentations, which have limited capacity.

2. Exhibitor
   - Manage account – The account management of visitor and exhibitor is very similar. Instead of study information, the exhibitor accounts allow uploading promotional materials, filling in descriptive texts and selecting work specialization the company is looking for. Unlike visitor accounts, exhibitor accounts are never deleted so they can be used every year.
Figure 2.3: UML Use Case Diagrams shows who can access which part of the information system and what they can do.
2. Planning

- **View visitor’s CVs** – Once an exhibitor is added to a list of visitor’s favorites, the exhibitor can browse through all such visitors and their CVs.

- **Create order** – The most extensive part of the web application is the ordering system. It allows exhibitors to reserve and customize a stand and to buy one of many other provided services.

- **Talk show registration** – Exhibitors can also book a time slot for a talk show presentation.

3. Administrator

- **Exhibitors administration** – In Umbraco, organizers have access to a list of all registered exhibitors. They can view and edit their account information and export the database into Microsoft Excel as .csv file.

- **Students administration** – Same as the exhibitors administration with the difference that it also supports simple filtering (the number of students is much higher than the number of exhibitors).

- **Talk show administration** – Administrators can create several slots for talk show and manually assign exhibitors to each of them. They can also view all enrolled students on individual talk shows and export a .csv file containing a list of them.

- **Stands administration** – Every year administrators have to manually create a list, one by one, including more than a hundred of available stands. Later on, they can see which exhibitor booked which stand.

- **Orders administration** – Administrators can also monitor order states (empty, half full, ordered) of each exhibitor. Once an exhibitor completes an order, a .pdf file containing the order information is created and displayed in the administration. The list of orders includes only data of the current year and it gets deleted after the job fair ends.
2. Planning

- **Global settings** – Each year the information system has slightly different settings like date of the event, order period, contact address for participants or more. Therefore, every year has to be set up first using Global settings in Umbraco. Unfortunately, there is no history of previous years because the new year’s setting overwrites the previous in the database.

- **Edit web content** – The basis of managing a web application is the possibility to edit its content. Umbraco 7, unlike Umbraco 6 offers more flexible approach and more content to be edited.

- **Manage blog** – Content editing also includes simple blog management in which administrators can hierarchically create and write various articles.

- **Partners administration** – Those exhibitors that sign up as partners of the job fair have special privileges and they need additional settings which is accessible through partners administration.

- **Exhibitors archive** – After every year of the job fair the content of the database is deleted, but the list of exhibitors is first stored in a particular archive located in Umbraco. This archive is later accessible and usable for listings in the web application.

The next few subsections briefly describe how some the most essential parts look, what flaws they have and how they are implemented.

### 2.2.2 Front-end

As mentioned before, the complete user interface was designed and implemented by the Iconcio company. They created a modern user interface with a clear design and navigation on the web is easy. Unfortunately, the website is not flawless; here I present these errors in order from negligible to the most serious ones:

- **Not entirely responsive** – In several different places on the web, especially in forms, the user interface is not sufficiently responding to user actions. For example, fields in a registration form
do not respond when a user fills in invalid data or submits the form with missing required fields. First, the form submits itself to the server with incorrect data, then it refreshes the page and then displays an error message. However, this message does not contain enough data to easily spot the erroneous fields.

- **Unexpected behavior** – Some links or buttons are not behaving as they are expected to. For instance, a login button is nowhere to be found but only a link that says "Registration" and its behavior is unpredictable. Sometimes, the link redirects to a registration page and the other time to a login page.

- **Broken English version** – One of the more severe problems is the English version of the web application that only works for half of the pages. A large part of the pages is not translated at all and so they are entirely hidden from the English version. Another example is that some links displayed in the English version redirect to Czech version pages and many pages in the English version are just in the Czech language. For example, the blog is written only in Czech language and so is the ordering process, which is the main drawback.

- **Confusing ordering process** – The service ordering is one the most crucial parts of the whole information system, and yet it is designed poorly. The reservation process has several sub-steps, but instead of creating it as a wizard-like application, all the steps are together in one long and very confusing page. Such a design often leads to mistakes made by exhibitors, which administrators have to deal with. For example, exhibitors are often confused which additional services (chairs, tables etc.) are provided by JobChallenge, so they usually unnecessarily order services already included in the stand.

The administration part in Umbraco 6 also includes many problems which are complicating the workflow. For example, there are such design elements that can be even dangerous for stored data. The controls are limited and delete buttons are acting without asking for confirmation, that can severely damage stored data. Especially, when the delete buttons are right next to other control buttons as shown in figure 2.4.
Some of the minor problems are outdated or looks poor speed – it can take tens of seconds to load a single page.

Figure 2.4: A screenshot of an Umbraco 6 orders administration as an example of a clumsy design – especially the “Delete” link next to the .pdf icon or the cramped columns.

2.2.3 Back-end

This subsection describes the division of implementation and functionality in more detail. Figure 2.5 shows how the system is split into three tiers: client, server and database.

Client tier contains four possible entry points into the information system. There is a web interface for both Umbraco 6 and Umbraco 7, and two for the web applications. Although the application for users looks like a single website, there are two of them. The primary URL address of the site is http://jobch.cz, however, while navigating through the pages, the user is unknowingly redirected between this domain and the secondary http://admin.jobch.cz; both hosted by different servers. One part of the web application is a static website without any business logic except the exhibitors’ archive management; this part is running in Umbraco 7. The second part of the web application includes every page that requires any extensive business logic, which is mostly everything connected to registrations and reservations; the dynamic second website is running in Umbraco 6.

Server tier in figure 2.5 consists of two separate clusters – Umbraco 6 and Umbraco 7. Both of them are running on different servers and they are communicating with each other through API.
As mentioned above, the JobChallenge information system is separately built and running in two different projects. In Umbraco 6 project, the administration part for organizers is implemented as a plugin and it is written in same technologies as the CMS itself. The part of the web application, available on the secondary URL address mentioned before, is also built in the Umbraco 6 project and its structure is illustrated in figure 2.5. The application provides a Web API which is handling user requests made through the website or requests sent by Umbraco 7. Behind the application there is a basic business logic which is either handling the request itself or using Icontio framework.

Icontio is a company that implemented most of the web application functionality built on an older .NET technology called Web forms. The framework is implementing most of logic behind the web application except the account management which is part of Umbraco 6.

Unfortunately, the framework is designed poorly and it violates the most fundamental tenet of programming – code repetition. This problem stands-out in the ordering process that consists of multiple items and the combination of several deals for exhibitors so that the final price may vary significantly in different cases. All prices, dic-
tionary values and other variables are hard-coded in the framework and some of them occur on multiple pages, so all the variables repeat in many parts of the code. On top of that, all prices change every year and organizers cannot edit them in any way; it has to be done by developers. Not only it distracts developers from working on other important projects but such work is also extremely complicated and mistakes occur very often.

It is obvious that such a system consisting of multiple subsystems is challenging to maintain and it is unnecessarily complicated. For example, when the user wants to access a page which contains an archive of exhibitors that participated in the past, the request goes to the server hosting Umbraco 7, because this version is handling a functionality behind the archive of exhibitors. Unfortunately, Umbraco 7 does not have access to a JobChallenge database, so it has to resend this request to Umbraco 6 using its API. In this case, Umbraco 6 can retrieve data without using Icontio framework and send it back to Umbraco 7. The CMS then processes the data, creates views and sends them to the client.

The primary motivation behind the work is to reduce the scale of the information system, therefore, become independent on Icontio framework and implement its functionality again, so it fits the current needs.

2.2.4 Database

The information system is using multiple databases running on several different database servers as it is shown in figure 2.5; they are all built on Microsoft SQL Server. Both CMSs use one release database server. For developers, there are development database servers where data can be modified arbitrarily without publishing the changes. Each CMS uses a database storing data needed for its system functionality like web content, user accounts, various settings and more. Data related to JobChallenge are stored in two separate databases.

One server runs, together with Umbraco 6 database, a JobChallenge database storing data shown in figure 2.6. This database contains data like orders, stands, talk shows and mainly enumeration values. The figure clearly shows a poor database design consisting of a few of large
tables and lots of smaller ones with a lack of mutual relations. Such a database is confusing for developers and challenging to maintain.

Account data of visitors and exhibitors is stored in Umbraco 6 database as a special type called Member. Members are used for registering and authenticating external users to the web application. The member type is divided into two sub-types, one for visitors and one for exhibitors. Database tables for members contain all exhibitors’ and visitor’ information which greatly complicates database management. Administrators are stored as Users which is another type of Umbraco members allowing them access to the CMS.

Umbraco 7 database stores web content including all pages, blog articles, images or more. Administrators of web content have the same type of accounts, User, as they need in Umbraco 6.

2.2.5 Security

Users working with the web application provide a significant amount of sensitive data such as passwords, personal information, CVs or more. Therefore, data security is of utmost importance. Both in the administration part in Umbraco and in the web application, it is necessary to prevent possible threats and to avoid potential damage. Thanks to Umbraco, both of these parts are sufficiently secured.

The administration part in Umbraco deals with following security issues:

- **Authentication** – One of the most fundamental problems of system security is to require authentication in order to provide particular content. Umbraco system has its account system and it requires login every time a user wants to access the system. These accounts are on demand created by administrators of the CMS. Umbraco also supports external login providers, so that not every user has to own a unique Umbraco account. Umbraco managed by WebCentrum offers MUNI (Masaryk University) login provider allowing users to use their university accounts.

- **Authorization** – Each user account is of a specific type, which restricts its access only to particular parts of the system. For example, developers have access to most of the system that contains more than a hundred of different websites and their settings,
Figure 2.6: Entity–relationship model of the old JobChallenge database
but administrators of each site have access only to their specific one. JobChallenge requires more sophisticated administration than the usual website and so the administrators have access to a plugin, which contains the administration part, made especially for the job fair.

- **Confidentiality** – As mentioned before, users have restricted access to information falling within their competence. As far as JobChallenge organizers are concerned, they have access to the administration part where all accounts and their content are stored. Overall, they can see everything that any user of the web application fills in and sends to the server, for example, all account information including personal details, CVs, order information or more with one exception. All passwords are hidden to everyone; they are one way hashed and stored safely in the database.

- **Human error** – No less important part of system security is to anticipate and adequately respond to human mistakes. Umbraco itself supports a history of changes for any item and a rollback to any of its previous versions. There is also a standard recycle bin containing all deleted content with the possibility to restore any of it and so the web content editing is safe without the risk of losing any changes. Unfortunately, the plugin containing the administration part of JobChallenge does not support either of the two. The system is therefore susceptible to human error and it is necessary to work very carefully.

Compared to the administration part, the web application has one disadvantage regarding system security. It is accessible to the general public and so it is much more vulnerable to several threats.

- **Unauthorized access** – Neither students nor companies want their accounts to be misused by a third party. Potential attackers could steal or manipulate sensitive data, create false orders or browse through students CVs. The web application supports a registration/login system provided by Umbraco along with its security but with slight differences. Login attempts to Umbraco is limited to five retries after which the account is blocked. Unfortunately,
the web application does not support anything like this and so it is vulnerable to brute force attacks.

- **Password theft** – As in any other information system, there are multiple ways to steal a password. The most basic attack is the already mentioned brute force attack, against which the system is not very well secured. But it is very slow and ineffective, and so the threat of such an attack is low. But there are more sophisticated forms of attack like eavesdropping or SQL injection against which the system is well protected on the contrary. Both the web application and Umbraco are using the secure communication protocol HTTPS using SSL certificates. This service provides encrypted communication, which reliably prevents the eavesdropping. SQL injection is another form of bypassing or misusing login forms when an attacker submits the form with a SQL command to read or modify a database or compromise the meaning of the original query. Fortunately, when dealing with such forms, Umbraco is validating user input so this attack is impossible. Even if an attacker would get access to the database, all passwords are hashed and stored using salted HMAC-SHA256. Decrypting such data is extremely difficult and it would take enough time for administrators to invalidate such passwords and inform users.

- **Integrity** – Client-server communication often leads to requests for data change. For example, when user changes account information and publishes the changes to the server. If such requests would be treated incautiously using public URL attributes to pass the data, attackers could manipulate with someone else’s accounts using manually crafted URL requests. For example, an attacker could get access to another account by changing the id of the targeted account in URL. Fortunately, communication like this is nowadays overcome by using sessions management, based on the exchange of encrypted cookies. Thus servers always know, who is sending the request without the need to include it in the URL address.

- **Exposure of CVs** – Beside passwords, the information system stores some other sensitive information like contacts, bank ac-
counts but mostly students CVs. Using the web application, students can submit their CVs, but these stay private until the student subscribes to an exhibitor. Until that, CVs are visible only to the organizers in the administration system. Once a student subscribes to an exhibitor, the student gives access to his or her CV to the exhibitor. In order to prevent mishandling, every action over any CV is logged in the information system. Thus organizers can see who and when is looking at whose CV. But still it has one drawback – exhibitors can freely download the shared CVs and after that organizers cannot see how exhibitors handle the files.

Overall, the information system is sufficiently secured and there has never been an attack of a third party that would be known. The only problems that occur are the afore mentioned human errors, which are most of the time just minor mistakes easily correctable. However, there were also serious problems, such as deleting most of the data from the database. These missteps are usually handled very quickly and prevented to happen again.

2.3 New system requirements

Naturally, JobChallenge organizers were also actively involved in the analysis and design of the new system. Several meetings were held during which the requirements for the new system were set.

The meetings took place regularly once a month. We discussed the current state of the information system, shared ideas, set requirements and defined priorities. Incrementally, we went through the whole system starting at registration and reservation process for exhibitors, which is the most crucial part of the system and continued through the rest of the web application and the administration in Umbraco. Later on, we demonstrated functional units of the system and tested them together.

At the beginning of the planning, we sorted all parts of the information system by priority in which order we discussed and implemented them. Specific features of each sub-part are prioritized by the MoSCoW method[19] into one of four categories (must have, should have, could have, and won’t have).
1. Exhibitors administration – Exhibitors accounts and their reservation system is the most crucial part of the entire application. The information system can be started just with this part done and the rest added later.

(a) Registration process – The registration is just a single page form but apparently necessary for the account management. It mostly stays the same as in the old version with minor changes:

- **Modified form items (Must have)** – The next year of the fair requires some critical changes in the form items.
- **Editable texts (Must have)** – All texts displayed in the registration form must be editable in the administration system and not hard-coded in the website.
- **Dynamic user interface (Should have)** – The form should response accurately to user’s action and mistakes. For example, display error messages or loading spinner.

(b) Reservation system – The main focus of the whole work which is described in subsection 3.1.1 in more detail.

- **Wizard-like application (Must have)** – The old reservation system is a lengthy single page application, which is extremely confusing for users. One of the primary goals of this work is to make the reservation application more user-friendly.
- ** Completely editable by organizers (Must have)** – The complete content of the reservation application (stand types, additional items, etc.) must be editable in the administration system, so their values are not hard-coded like in the old system.
- **Dynamic user interface (Should have)** – Using the wizard-like application requires a good user experience design, so users are not confused during the reservation process.
- **Interactive map for stand selection (Could have)** – In the old version, exhibitors selected stand positions by looking into a map and then writing down a number of the
chosen stand. It would be very helpful if there was an interactive map, so users could easily see, which stands’ positions are available or not and select a stand directly in the map.

(c) **Talk show registration** – The front-end for talk show registration is sufficient enough. Thus there are no significant changes needed.

(d) **CV management** – Again, this part remains without any major changes.

2. **Visitors administration** - The web application can be launched without a functional part for visitors. According to the analysis of website traffic, students usually create or login to their accounts less than two months before the job fair. Unlike exhibitors who need to register half a year before JobChallenge.

(a) **Registration process** - A single page registration form that is very similar to the exhibitors registration process.
   - **Modified form items (Must have)**
   - **Editable texts (Must have)**
   - **Dynamic user interface (Should have)**

(b) **CV management** – It is only an option to upload a document from a computer and there are no changes needed.

(c) **Talk show reservation** – The current application for the reservation is clear and smooth, so it only migrates to the new system.

3. **Administration in Umbraco** – At the time of launching the web application, the administration part may not yet be completely functional. Data can be manually set in the database and the administration part can only provide a basic interface to view the data. Thus, although this part is essential in the future, it does not need to be deployed at the same time as the web application; and so, the administration part has the lowest priority. Overall, JobChallenge organizers do not have many requirements.

(a) **Global settings**
2. Planning

- **Simplification (Should have)** – Global settings in the old administration part contains several items that are not used anymore. The new version should include only necessary settings.

- **Archive of settings of previous years (Could have)** – With every new year of the fair, the information about previous years are lost. In the new version, the database should support an archive for past years so it can be displayed in the administration settings.

(b) **Order settings**

- **Completely customizable by organizers (Must have)** – As mentioned before, organizers must have control over all the reservation process, including settings of every orderable item. Since this is not possible in the old version, it has to be designed and implemented in the new version.

(c) **Stands administration**

- **Completely customizable by organizers (Must have)** – The customization of stands is the same problem as with order settings; it must be completely editable.

- **Easier administration of available stands (Must have)** – The fair can contain over a hundred of available stands for exhibitors. The stands were utterly editable in the administration system. However, they had to be created and edited one by one, which is an extremely lengthy process due to their number. Overall, the stand administration must be designed more simply.

(d) **Exhibitors administration**

- **Improved user interface (Must have)** – As mentioned before, the old user interface is poorly designed and contains some dangerous design choices that must be avoided in the new system.

- **Archive of exhibitors (Should have)** – The archive of exhibitors is separated from the rest of the administration. It would be practical if the whole exhibitor administration were in one place.
(e) Students administration

- Improved user interface (Must have) – Very similar to exhibitors administration.
- Archive of students (Should have) – Very similar to exhibitors administration.

(f) Talk show settings

- Easier administration (Should have) – The problem of the talk show administration is the same as the stand administration. Every talk show has to be edited manually, instead of doing it generally.
3 Methodology

This chapter describes the development process of the application. The first section shows the design of the new application consisting of drafts, models and diagrams. The rest of the chapter briefly describes the implementation processes including an introducing to used technologies, security measures and ongoing maintenance.

3.1 Design of the new system

At first, together with JobChallenge organizers, we defined the content and the new look of the application. It was necessary to do this first because it helped us to choose the right architecture and model the database, which I describe later in the chapter. The end of the section is focused on back-end layer of the application.

3.1.1 Web application

To draw drafts and creating all diagrams I used an extensive online tool called draw.io [20]. It helped us easily share and comment every change we made during the process.

User interface of the web application, consisting of authorization (login and registration), exhibitor and student administration, stayed mostly the same as the old version. Therefore, there was no need for any extensive design. However, JobChallenge organizers wanted to completely redesign the ordering process to make it more user-friendly; a single page transforms into a wizard-like application. This change required a detailed analysis of the old ordering process, detecting its components and appropriately splitting the ordering workflow into several steps.

Unfortunately, the content of the ordering process is dynamic and varies by users choices, so it was not intuitive how to split the process. For example, in the beginning, exhibitors have to choose if they are an exhibiting company or a non-profit organization. Depending on this choice, they are allowed to buy a custom stand and select its position in the exhibition area or not because each non-profit organization gets the same type of stand and equipment. The order continues with choosing
additional services as more furniture, refreshment, advertisement and more. The rest of the ordering process is only a recapitulation that is same for both types of orders.

Moreover, JobChallenge organizers decided they needed a new type of order, which is an additional order. This type of order is available only after finishing the standard order. It allows exhibitors to buy additional services in case they forgot earlier.

Figure 3.1 shows all necessary components of the ordering process and how they are distributed across the ordering process. The components are:

- **Order type** – As mentioned before, exhibitors are either companies or non-profit organizations. Depending on this choice, individual components are displayed or hidden; the purple components in figure 3.1 are visible only to exhibiting companies.

- **Stand type** – Exhibiting companies can choose a specific stand type defined by its size, equipment and position.

- **Additional services** – A list of additional equipment, special services and advertisements.

- **Payer info** – The final price is dynamically calculated according to company’s information (VAT payer, non-VAT payer, etc.).

- **Stand position** – A selection of an exact stand position.

- **Recapitulation** – The final summary of prices per each item and in total.

The following figure 3.2 describes how we separated the components into five different steps, so it is easier for users to navigate. The steps labeled with purple color are available only to exhibiting companies and not to non-profit organizations.

Before I implemented any of these steps, I prepared a draft for all of them using Draw.io tool mentioned earlier in this chapter. Figure 3.3 displays one of these drafts and shows one step of the ordering process, specifically the selection of the type of an exhibition area.

The multi-step wizard-like application has several advantages. Mainly, it is much clearer and more intuitive to navigate across the
ordering process. For example, the stand type selection and the additional services contains dozens of choices, usually with additional descriptions, so the distribution between two steps significantly improves the user experience. Moreover, a single screen for each component provides more space and allows the component to have a more sophisticated and more explicit design. For example, the stand type component contains dozens of options which were rendered below each other in the old design. Such a design choice inappropriately prolonged the ordering screen. The difference between both designs is visible in figures 3.4 and 3.5.

One single step is dedicated to the interactive map for the stand selection, which also requires a lot of space. The interactive map is another innovation which JobChallenge organizers needed. Using the map, exhibitors can plainly see, which positions are free or taken by
3. Methodology

Figure 3.3: A draft, drawn in online tool Draw.io, showing one step of the ordering process.
3. Methodology

Figure 3.4: Example of the old unclear design for the stand type selection

which company. Also, the payer info and recapitulation components were put together in the final step, because the prices shown in the recapitulation depends on the choice made in the payer info component. Therefore, exhibitors can see how their choice affects the final price.

Also, the application saves the progress after each step, so exhibitors can leave the ordering process and later continue where they left.

After submitting an order, the exhibitor receives an e-mail with an invoice containing all necessary information. In the old version, exhibitors could create only one order and there was not a link to download the invoice anywhere in the administration. Based on experience, the JobChallenge organizers chose to include an additional
3. Methodology

Figure 3.5: Result of the new design for the stand type selection

order, because many exhibitors decided to buy additional services long after they finished the standard order. Therefore, this additional order allows exhibitors to order additional services later on. Since there is the possibility to create multiple orders, we designed an order summary where exhibitors can view and download invoices of all their earlier orders.

3.1.2 Umbraco administration

Unlike the web application, the Umbraco administration required a significant redesign. Partly because it moved from the old Umbraco to the new version of Umbraco, which uses different technologies, but also because the old administration used clumsy design described in section 2.2.2. Also, the migration was an excellent opportunity to rethink the design and remove unnecessary things.
3. Methodology

Figure 3.6 shows the overview of the old administration. Each box represents a page in the administration menu and the list below demonstrates its functionality. The gray boxes mark obsolete pages which are no longer used. The figure clearly shows an excessive complexity and an unnecessary amount of pages. Not only is it difficult to navigate in it, but it’s also very slow due to the old Umbraco 6 application.

On the other hand, figure 3.7 represents a new design of the administration. Again, each box indicates a single page, but the subheadings in the boxes mark separate tabs in the page. The new structure cumulates similar functions together, so they are easier to find. For example, exporting of exhibitors’ is now intuitively placed in the exhibitors administration, unlike in the older version, where it is placed in global settings. Another example is putting a list of exhibitors and a list of orders together on one page, but in separate tabs. Therefore, administrators can quickly switch between those lists. A draft of the exhibitors administration, specifically the tab containing orders, is shown in figure 3.8. In addition, figure 3.9 shows the early implementation made quickly so the organizers could access exhibitors’ data.

![Figure 3.6: Old administration structure where each box represents a single page and the list below its functionality](image)

The new version of administration is also vastly improved by a history of previous years. In case of a problem, administrators can now go through past orders, exhibitor accounts or settings.
3. Methodology

### 3.1.3 Database

One of the most extensive parts of this work was to correctly design a new database containing all requirements and improvements mentioned before. The old database design is described in section 2.2.4 and shown in figure 2.6. Unfortunately, the design of the new database is so extensive, it does not fit in a single page, therefore, it is split into several figures.

I used several different colors to make the diagrams clearer. Orange Umbraco frame stands for UMBRACO scheme and purple JobChallenge frame stands for JOBCH scheme (There are two schemes related to JobChallenge used in our database.). The light yellow boxes are not database tables but enumeration types and their values. The gray tables indicate they have a connection to tables not shown in the frame. The other colors are just for better orientation.

The first figure 3.10 shows tables related to exhibitors. Tables for ordering process follow in figure 3.11. Visitors are in figure 3.12 and figure 3.13 contains tables for talk show administration.

### 3.1.4 Back-end

As mentioned before in the section 2.2.3, a major part of the old system was part of Umbraco 6. More specifically, it was the Umbraco

---

**Figure 3.7:** New administration structure where each box represents a single page, the subheadings marks separate tabs and the lists below their functionality

<table>
<thead>
<tr>
<th>Exhibitors administration</th>
<th>Students administration</th>
<th>Stands administration</th>
<th>Talk show administration</th>
<th>Global settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exhibitors</td>
<td>List of students</td>
<td>List of stands</td>
<td>Talk shows</td>
<td>Settings</td>
</tr>
<tr>
<td></td>
<td>CVs</td>
<td>Editing</td>
<td>List of talk shows</td>
<td>Exporting</td>
</tr>
<tr>
<td></td>
<td>Documents</td>
<td>Filtering</td>
<td>Editing</td>
<td>History</td>
</tr>
<tr>
<td></td>
<td>Editing</td>
<td></td>
<td>Filtering</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exporting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Filtering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV access monitoring</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Orders</th>
<th>Registrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of orders</td>
<td>List of registrations</td>
</tr>
<tr>
<td>Editing</td>
<td>Filtering</td>
</tr>
<tr>
<td>Exporting</td>
<td></td>
</tr>
<tr>
<td>Filtering</td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Log</th>
<th>CV access monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. Methodology

Figure 3.8: An example of a draft made for administration of orders in Umbraco

Figure 3.9: Early functional design for administration of orders in Umbraco
3. Methodology

Figure 3.10: Part 1 of the new database containing tables related to exhibitors
Figure 3.11: Part 2 of the new database containing tables related to orders
Figure 3.12: Part 3 of the new database containing tables related to visitors
Figure 3.13: Part 4 of the new database containing tables related to talk shows
administration and the user administration in the web application. This whole part had to migrate into the new version of Umbraco. Since the old system used significantly different technologies, a brand new architecture had to be designed; it is visualized in figure 3.14.

Figure 3.14: Overview of the new back-end architecture

In contrast with the old back-end architecture shown in figure 2.5, the new system is simpler and more straightforward. On a client side, there are two entry points into the system. Through Umbraco 7, authorized users can access either a web-site management or the JobChallenge administration. On the other hand, visitors and exhibitors can freely visit the JobChallenge web application. Both Umbraco 7 administration and the administration in the web application are running two separate Angular 5 applications. One is implemented as a plugin in Umbraco 7 and the second one is directly integrated into the JobChallenge web application. An implementation of both Angular 5 applications are following standard Angular guidelines. Therefore, their architecture is very similar. Figure 3.15 shows a top-down architecture of the Angular application used in the web application for account administration. Since the Angular 5 applications are so extensive, they both contain dozens of files, components or more; the figure shows only a fraction of the whole application.

Single web pages in any Angular 5 application are composed of components. For example, there is a component for login and registration form. Angular 5 components can also be nested, for instance, there is a notification component, that can be placed anywhere and it shows notifications in case of errors or warnings. Even the whole application can be viewed as a single component positioned somewhere on the
3. Methodology

Figure 3.15: A brief overview of the Angular 5 application used within the web application

web-site. Overall, the visible part of the Angular 5 application can be viewed as a tree of components.

Another big part of both applications are services. These are auxiliary controllers providing components with functionality. For example, a data service communicates with a server API and provide components or services with data from the database. Both Angular 5 applications use guards, which are a special kind of services, that make sure only authorized users access a specific page.
As mentioned above, both Angular 5 applications communicate with server API in order to get relevant data as it is shown in figure 3.14. The server side is split into three main tiers.

1. **Services** – Acquire data from a database and send them, using DAO models, to modules.

2. **Modules** – Implement a business logic over data received from services and forward them to controllers.

3. **API Controllers** – Handle requests from the client. They call modules for data and return them to client packed in result models.

![Diagram of server-level data processing](image)

Figure 3.16: Illustration of server-level data processing

### 3.2 Technology

The choice of technologies to implement the application is mostly tied to the system that WebCentrum is already using. New JobChallenge database runs on the same database server as the rest of WebCentrum databases. All databases run on Microsoft SQL Server that is a relational database management system developed by Microsoft. The database system is using standard language for managing data called SQL.

Umbraco 7 and all WebCentrum web applications, including the JobChallenge application, are primarily written in C#. Front-end of all web applications are built upon Microsoft’s .NET Framework, using ASP.NET. Umbraco 7’s front-end is implemented using AngularJS for a single-page application experience. At the time of working on the
JobChallenge application, it has also been announced that Umbraco would officially support Angular 2 in the future.

Although most of the JobChallenge web application is built using ASP.NET framework (composed of C#, HTML5, CSS, JavaScript and more), the user’s administration on the JobChallenge website required a more sophisticated approach. We needed a suitable front-end web application framework specialized for single-page applications. There are several JavaScript-based frameworks appropriate for this task like AngularJS, Angular 5, React or more. As mentioned before, we chose Angular 5 for both the Umbraco administration and the web application. Angular 5 is a newer version of AngularJS, heavily based on Angular 2, which was announced by Umbraco to be supported in future release. Therefore, applications implemented using Angular 5 would be compatible with the rest of Umbraco and we could use the experience gained in the process.

### 3.3 Security measures

Throughout the development, we ensured several security measures to protect our information system. At first, Umbraco itself provides good protection for both the CMS itself and its websites. Since version 7.3, Umbraco uses ASP.Net Identity and ASP.Net Membership as the authentication mechanisms to ensure security for storing and validating passwords.

Passwords are by default hashed with salted HMAC-SHA256. Moreover, Umbraco locks out a user after several incorrect login attempts to avoid brute force attacks. Password rules like length or character diversity are completely configurable, and by default, users are automatically logged out after several minutes of inactivity. Also, Umbraco supports a full OAuth login system, which allows a use of a 3rd party login system like a unified login system provided by Masaryk University. All mentioned authentication mechanisms can be used for both the CMS and its websites.

Naturally, Umbraco supports an account authorization, so any user should not have more privileges than they need. For example, a web content editor should have access to a web application configuration and so on.
WebCentrum runs Umbraco using HTTPS together with its websites. JobChallenge web application also runs on HTTPS providing its users an encrypted communication, hence, preventing an eavesdropping and data loss during communication.

To prevent attacks on the web application like XSS or SQL injection, we implemented several security measures. Next to client-side form validation, we added a server-side validation to enhance the security and to prevent XSS trying to bypass the client-side validation. Similar to XSS is SQL injection, which is prevented by using parameterized queries and object-relational mapping (ORM). Moreover, we added restrictions to several API methods, so only authorized users can use them.

On top of that, Umbraco supports a wide range of configuration options to enhance the security. For instance, developers can set limits to a file upload size to prevent denial of service attacks caused by users posting large files to the server. Also, it is possible to modify an execution timeout, which is an attribute indicating the maximum number of seconds that a request is allowed to execute before being automatically shut down by the application.

Altogether, Umbraco alone provides a wide range of security measures enough to run secured websites. When implementing a server side of applications, developers have to follow basic security principles to reduce vulnerability and prevent possible threats. Usually, it is recommended to have information systems tested by penetration testers to find additional vulnerabilities.

### 3.4 Result

Throughout the work, I analyzed the old information system and together with JobChallenge organizers stated all requirements for the new system. Then, I designed the architecture and the look of the new JobChallenge information system. Finally, the necessary part of the whole was implemented, needed to register new exhibitors and create orders.

At the time of finishing the master’s thesis, the JobChallenge web application contained a completely new administration system for exhibitors. It was possible to register, login and edit exhibitors accounts.
But mainly, I completely redesigned the old ordering system, which is part of the exhibitors’ administration, and implemented a new, modern and more user-friendly ordering application. In Umbraco, basics needed to manage exhibitors’ accounts and their orders were implemented. Overall, the new system successfully migrated from the old version of Umbraco. Therefore, the JobChallenge information system does no longer dependent on Umbraco 6.

Moreover, the system was tested by WebCentrum staff together with JobChallenge organizers who regularly provided me with feedback and mistakes, which I promptly repaired. Finally, the system was published in time and thanks to this; the first exhibitors could create their first orders.
4 Conclusion

The primary goal of this master’s thesis was to prepare the JobChallenge information system for the new year of the fair. The work consisted of implementing a new exhibitors administration together with the administration in Umbraco, so the new year of JobChallenge could launch in time.

Throughout the process, I followed general rules of information system development together with security principles. In the end, a first iteration of the information system was successfully implemented and published, which allowed exhibitors to create or update their accounts, but mostly order their participation in the new year.

Hereafter, the JobChallenge information system still has to be extended by an administration for students, and the administration part in Umbraco lacks the complete management of accounts and orders. Further, introducing GDPR (General Data Protection Regulation) brings responsibility to handle user data properly. Mostly, when the students’ CV management will be implemented, it must strictly keep the regulations defined by GDPR. When it comes to security, it is highly recommended to have the system tested by penetration testers to find security flaws in the system.

Moreover, due to frequent changes in requirements and the need to release the first iteration of the system in a short period of time, the source code is rushed in some places and requires refactoring. Before further work, it is recommended to clean up the code, especially in the area of communication with the database.

Reading my thesis, people can learn the basics of information system development including designing a web application and its security. Above that, I introduce it in practice, describing how I implemented a part of the new JobChallenge information system and prepared the groundwork for future development.
Bibliography


A Content of the CD

1. The text of this master’s thesis in a PDF format (folder /text).

2. Source code of both a server and a client side, which is split between two folders (/server and /client). Due to the fact that the JobChallenge information system is integrated into Web-Centrum’s Umbraco, it is not possible to provide an executable standalone application.

3. All drafts created throughout my work are are split between three subfolders in folder /drafts:
   - drafts/authentication – Contains a design of a registration form.
   - drafts/ordering_process – Stores all drafts related to the ordering process.
   - drafts/umbraco_administration – Includes the complete design – screen by screen – of the administration part in Umbraco.