



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM AKVAPARKU

INFORMATION SYSTEM OF A WATER PARK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JOSEF MĚŘÍNSKÝ

VEDOUČÍ PRÁCE
SUPERVISOR

Ing. RADEK BURGET, Ph.D.

BRNO 2016

Zadání bakalářské práce

Řešitel: **Měřínský Josef**
Obor: Informační technologie
Téma: **Informační systém akvaparku**
Information System of a Water Park

Kategorie: Informační systémy

Pokyny:

1. Seznamte se s požadavky vedení akvaparku Aqualand Moravia na informační systém a intranet pro zaměstnance.
2. Prostudujte existující technologie pro tvorbu informačních systémů s webovým uživatelským rozhraním.
3. Po konzultaci s vedoucím a zadavatelem zvolte vhodnou platformu a navrhnete architekturu informačního systému pro zaměstnance Aqualandu.
4. Implementujte navržené řešení na zvolené platformě. Zejména implementujte modul plánování dovolené, úkolů a správu aktualit.
5. Optimalizujte vytvořené řešení pro snadné použití na mobilních zařízeních.
6. Proveďte testování vytvořeného systému a zhodnoťte dosažené výsledky.

Literatura:

- Gutmans, A., Rethans, D., Bakken, S.: Mistrovství v PHP 5, Computer Press, 2012
- Dokumentace k projektu Nette: <http://doc.nette.org/cs/2.2/>

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

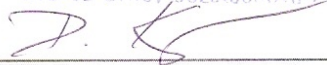
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Burget Radek, Ing., Ph.D.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Hlavním cílem bakalářské práce je navrhnout a implementovat informační systém pro akvapark Aqualand Moravia, který nahradí dosavadní způsob přidělování úkolů. Je uzpůsoben jako informační zdroj pro zaměstnance akvaparku, který byl doposud řešen pouze nástěnkou. Se systémem bude pracovat především TOP management. Usnadní komunikaci mezi jednotlivými odděleními. Bude nahrazovat způsob žádostí o dovolenou, který byl doposud pouze v papírové podobě. Cíle práce byly naplněny bezzbytku.

Abstract

The main aim of bachelor thesis is design and implementation of information system for aquapark Aqualand Moravia, which will replace actual way of allocating tasks. System is designed as an informational source for aquapark employees to replace current bulletin board as source of informations. System will be used especially by TOP management and will make communication between departments easier. System will replace present form of managing vacancies which was held in paper notes. Goals of bachelor thesis were fulfilled completely.

Klíčová slova

Informační systém, HTML, CSS, PHP, JavaScript, Bootstrap

Keywords

Information system, HTML, CSS, PHP, JavaScript, Bootstrap

Citace

MĚŘÍNSKÝ, Josef. *Informační systém akvaparku*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Burget Radek.

Informační systém akvaparku

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Radka Burgeta Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Josef Měřínský
18. května 2016

Poděkování

Rád bych poděkoval svému vedoucímu za užitečné rady a vstřícnost při spolupráci. Také bych rád na tomto místě poděkoval, vedoucímu IT Aqualandu Moravia, Petru Procházkovy za poskytnuté informace a konzultace ohledně fungování organizační struktury firmy a za konzultace ke správné funkčnosti systému.

© Josef Měřínský, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Specifikace a analýza informačního systému	4
2.1	Informační systém	4
2.1.1	Data	5
2.1.2	Informace	5
2.2	Analýza požadavků	5
2.3	Use Case Diagram	5
2.4	Uživatelské role	6
2.4.1	Administrátor	6
2.4.2	Ředitel	6
2.4.3	Vedoucí oddělení	6
2.4.4	Zaměstnanec	6
2.5	Akce	6
2.5.1	Oddělení	7
2.5.2	Zaměstnanci	7
2.5.3	Úkoly	7
2.5.4	Požadavky	7
2.5.5	Aktuality	7
2.5.6	Žádosti o dovolenou	7
3	Použité technologie	8
3.1	PHP	8
3.2	JavaScript	9
3.3	HTML, CSS	9
3.4	Bootstrap	10
3.5	FullCalendar	10
3.6	DateTimePicker	10
3.7	MySQL	11
4	Návrh informačního systému	13
4.1	Konceptuální model	13
4.2	Zaměstnanci	13
4.3	Oddělení	15
4.4	Úkoly	15
4.5	Požadavky	16
4.6	Komentář	16
4.7	Oprávnění úkolů	16

4.8	Žádost o dovolenou	16
4.9	Aktuality	17
5	Implementace	18
5.1	Popis uživatelského rozhraní	18
5.2	Zabezpečení	20
5.2.1	Autentizace	20
5.2.2	Šifrování dat	20
5.2.3	SQL injection	21
5.3	Oprávnění	21
5.4	Adresářová struktura	22
5.5	Bootstrap	22
5.6	Popis hlavních částí	23
5.6.1	Výpis aktualit	23
5.6.2	Zobrazení úkolů	23
5.6.3	Administrace	24
5.6.4	Kalendář dovolených	24
6	Testování	26
7	Závěr	27
7.1	Uvažovaná vylepšení	27
	Literatura	28
	Přílohy	30
	Seznam příloh	31
A	Obsah CD	32
B	Manual	33
C	Use Case Diagram	39
D	ERD	40
E	Zhodnocení vedoucího IT Aqualandu Moravia.	41

Kapitola 1

Úvod

Hlavním cílem bakalářské práce je vytvoření informačního systému pro zadavatele Aqualand Moravia. Systém je vytvořen jako webová aplikace a klade důraz především na vypracovávání a delegování jednotlivých úkolů, které byly doposud řešeny pouze formou emailů. Jedná se především o vytvoření systému, který dokáže jednoduše a efektivně vytvářet, přiřazovat a delegovat jednotlivé úkoly na jednotlivé zaměstnance, či celá oddělení. Dalším cílem je umožnit zaměstnancům prostřednictvím systému požádat o dovolenou, která se následně zobrazí řediteli na schválení.

Jelikož je potřeba vytvořit dynamický systém, aby nebyl omezen pouze na specifické zařízení, je realizován jako webová aplikace, která využívá technologií dostupných u většiny webových prohlížečů.

Pro lepší dynamičnost je potřeba v dnešní době myslet na mobilní zařízení. Proto je systém propojen s platformou Bootstrap, která optimalizuje vzhled informačního systému právě pro mobilní a jiná zařízení.

Následující kapitoly popisují a vysvětlují celý proces vytvoření informačního systému. Kapitola 2 se věnuje analýze požadavků a obecné definici informačního systému. Jednotlivé technologie použité v informačním systému jsou popsány v kapitole 3, následovány návrhem informačního systému v kapitole 4, zahrnující detailní popis funkcí jednotlivých uživatelů a také návrhem konceptuálního modelu. V další kapitole s označením 5 je pojednáváno o konkrétní implementaci ve kterém je pozornost věnována nejen popisem uživatelského rozhraní, ale také například zabezpečení. Před samotným závěrem práce je v kapitole 6 pojednáváno o testování systému, jak v průběhu, tak na závěr prací.

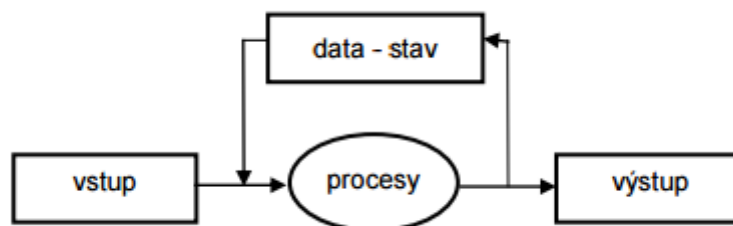
Kapitola 2

Specifikace a analýza informačního systému

Kapitola je věnována specifikaci a obecným definicím informačního systému. Je zde tedy popsán obecný popis takového systému. Následuje analýza jednotlivých požadavků zadavatele, ve které jsou popsány potřebné úkony, které bude třeba navrhovat a modelovat v následujících kapitolách.

2.1 Informační systém

Informační systém zobrazený na obr. 2.1 se skládá ze vstupní a výstupní části, kudy se do systému vkládají, resp. se získávají informace. Tyto informace jsou data [2.1.1], která interpretuje uživatel jako informace [6]. Mezi vstupem a výstupem probíhá transformace těchto dat. Transformační část typicky provádí nad daty různé algoritmy. Probíhají zde procesy a musíme se tedy zabývat způsobem definice procesů, jejich vzájemnou interakcí, paralelním prováděním apod.



Obrázek 2.1: Schéma informačního systému [6]

Typickou částí informačního systému je zpětná vazba, která využívá uloženého stavu systému. Ne všechny výstupy procesů jsou tedy přímo závislé na okamžitých vstupech, ale typicky závisejí na stavu systému. Na obr. 2.1 vidíme typické nezbytné součásti pojmenované pojmy informačních technologií:

- data uchovávající stav systému a
- procesy realizující transformace často ve formě transakcí

2.1.1 Data

Stav informačního systému reprezentují data [6]. Ta jsou jakékoli vyjádření (reprezentace) skutečnosti, schopné přenosu, uchování, interpretace či zpracování. Podstatné je, že data nemusejí mít a často také nemají sémantiku (význam). Mohou být zpracovávána čistě formálně pouze na základě syntaxe (např. formální jazyky). Syntakticky lze data definovat poměrně snadno a jsou k tomu vypracovány velmi dobré postupy (např. gramatiky). Hodnoty dat obvykle udávají stav nějakého systému.

2.1.2 Informace

Interpretací dat získáme informace [6]. V oblasti výpočetní techniky se za informaci považuje kvantitativní vyjádření obsahu zprávy. Za jednotku informace se ve výpočetní technice považuje rozhodnutí mezi dvěma alternativami (0, 1) a vyjadřuje se jednotkou nazvanou bit. Informace jsou data obsahující sémantiku.

2.2 Analýza požadavků

Informační systém, který poslouží především top managementu, je vytvořen pro akvapark Aqualand Moravia, pro zpracovávání úkolů pracovníky jednotlivých oddělení, zlepšení komunikace mezi pracovníky, ale také jako informační intranet zaměstnanců Aqualandu. Mimo jiné poslouží jako systém žádostí a správy dovolených.

Pro správu informačního systému je v první řadě potřeba administrátor, který vytvoří oddělení a přiřadí k nim vedoucí pracovníky. Tito pracovníci již budou spravovat své zaměstnance a budou mít možnost, stejně jako administrátor, je vytvořit, nicméně pouze pro svá střediska. Zvláštním uživatelem systému je ředitel, který bude mít na starost správu dovolené, o kterou si každý zaměstnanec přes systém bude moci zažádat.

Hlavním požadavkem akvaparku je vytvoření systému pro udělování a vypracovávání úkolů a tím vylepšit komunikaci na pracovišti. Je potřeba vytvářet nové úkoly, které se budou přiřazovat jednotlivým zaměstnancům, popřípadě oddělením. U úkolů je potřeba nastavovat datum, do kdy je nutné jej splnit, jeho popis, prioritu úkolu a stav, ve kterém se úkol nachází. Jeden úkol může mít i více požadavků. Dále zařídit, aby tyto úkoly viděli pouze potřební uživatelé.

2.3 Use Case Diagram

Na základě analýzy požadavků zadavatele je vypracován diagram příkladu užití (Use Case Diagram), který zobrazuje chování systému tak, jak ho vidí uživatel. Účelem diagramu je popsat funkcionalitu systému, tedy co se od něj očekává. Diagram vypovídá o tom, co má systém umět, ale neříká jak to bude dělat. Vhodně tak doplňuje a zpřehledňuje následující textový popis, který je z hlediska dalšího návrhu ovšem významnější. Základními symboly diagramu jsou:

- Elipsa, která odpovídá případu užití.
- Postava, která představuje uživatele systému.
- Plná čára, která představuje interakci mezi uživatelem a případem užitím.
- Přerušovaná čára, která představuje vazby mezi případy užití.

- Orámování, které představuje ohraničení systému.

Model případu užití je navíc vhodným základem pro navrhování uživatelského rozhraní systému. Výsledný model, navržený pro akvapark, je zobrazen na obrázku v příloze C. Z tohoto diagramu je následně vycházeno při tvorbě databáze a celého informačního systému.

2.4 Uživatelské role

Na základě analýzy požadavků jsou vytvořeny čtyři uživatelské role: administrátor, ředitel, vedoucí oddělení a zaměstnanec. Registrovaným uživatelům je přidělena právě jedna uživatelská role. Uživatel bez přihlášení má přístup pouze k aktualitám na intranetu.

2.4.1 Administrátor

Klíčový uživatel informačního systému. Má přístup ke všem hlavním funkcím informačního systému, které jsou popsány use case diagramem v sekci 2.3. Jelikož i administrátor plní funkci zaměstnance a tedy někoho, na koho budou delegovány úkoly, je u něj omezen přístup ke všem úkolům. Vidí pouze úkoly, které sám vytvořil, které jsou mu přiřazené anebo přiřazené jeho oddělení.

2.4.2 Ředitel

Ředitel je jeden ze zaměstnanců, který má oproti obyčejnému zaměstnanci přístup ke správě žádostí o dovolenou. K žádosti slouží kalendář, do kterého se promítají již schválené žádosti. Při nové žádosti vidí, kdy dotyčný žádá o dovolenou a z jakého důvodu. Na základě zaplněného kalendáře schválených dovolených se bude moci rozhodnout, zda žádost schválí, nebo zamítne. Při zamítnutí dovolené bude moci sepsat důvod rozhodnutí.

2.4.3 Vedoucí oddělení

Vedoucí je jeden ze zaměstnanců, který vede právě jedno oddělení. Oproti obyčejnému zaměstnanci vidí všechny úkoly, které spadají pod jeho oddělení. Spravuje své vlastní oddělení vytvářením nových zaměstnanců, popřípadě editací stávajících. Vedoucí oddělení může vytvářet oproti běžnému zaměstnanci aktuality na intranet.

2.4.4 Zaměstnanec

Zaměstnanci mohou vytvářet nové úkoly, přiřazovat úkoly jednotlivým zaměstnancům či oddělením, ale také komentovat jim přiřazené úkoly a měnit stavy svých úkolů.

Zaměstnanci si prostřednictvím informačního systému žádají o dovolenou, na základě reakce ředitele se dozví výsledek jejich žádosti.

2.5 Akce

Každý uživatel systému má přístup k určitým možnostem systému. Mezi možnosti patří: správa oddělení, zaměstnanců, úkolů, aktualit a žádostí o dovolenou.

2.5.1 Oddělení

K vytváření a úpravě samotného oddělení má přístup pouze administrátor. Uchovávání informací o příslušnosti zaměstnanců k jednotlivým oddělením, umožní přiřazovat jednotlivým oddělením úkoly. Při vytvoření oddělení je vybrán jeden ze zaměstnanců, který zvolené oddělení povede. Vybraný zaměstnanec získá oprávnění vedoucího oddělení, tudíž možnost spravovat přiřazené oddělení.

2.5.2 Zaměstnanci

Vytvořit zaměstnance a nastavit oprávnění daného zaměstnance a jeho oddělení je v moci administrátora. Kromě něj může zaměstnance vytvářet vedoucí oddělení. Zaměstnanci vytvoření vedoucím oddělením mají automaticky přiřazeno stejné oddělení, jako jejich vedoucí.

2.5.3 Úkoly

Úkol může vytvořit kterýkoliv uživatel systému, který bude v systému přihlášen. Vytvořený úkol může editovat pouze jeho autor, nebo osoba zodpovědná za jeho splnění. Editoři jsou oprávněni přiřazovat další oprávněné osoby, které úkol uvidí. Uživatelé mimo seznam oprávněných, popřípadě vedoucí oddělení úkol neuvidí.

2.5.4 Požadavky

Kdokoliv s oprávněním k danému úkolu, může vytvářet nové požadavky. Pokud uživatel má přístup k úkolu, může editovat kterýkoliv z požadavků a mazat. K jednotlivým požadavkům se přistupuje po zobrazení detailů k úkolu.

2.5.5 Aktuality

Nově vytvořené aktuality jsou zobrazovány na úvodní stránce informačního systému. Aktuality jsou viditelné pro každého uživatele informačního systému, přihlášených i nepřihlášených. Vytváření a editace aktualit jsou umožněny uživatelům v roli: ředitel, administrátor, vedoucí oddělení či uživateli s oprávněním.

2.5.6 Žádosti o dovolenou

Kterýkoliv zaměstnanec má možnost zaslat žádost o dovolenou, která obsahuje termín dovolené a důvod žádosti. Zaměstnanec vidí seznam svých žádostí, které jsou aktuální a v jaké fázi se nachází. Při zamítnutí žádosti zjistí důvody, které ředitele k zamítnutí vedly. Každý zaměstnanec vidí kalendář všech aktuálně schválených dovolených. Přehled informuje zaměstnance o rozložení dovolených pro předejití nahromadění dovolených různých zaměstnanců v jeden termín.

Kapitola 3

Použité technologie

Informační systém je implementován jako webová aplikace, která má za cíl nezávislost na operačním systému a díky využití technologii Bootstrap je snadno použitelný pro různé velikosti displejů. Jednou z výhod je logika informačního systému, která je uložena na serveru. Tím odpadá omezení na výpočetní výkon klientského zařízení. Pro použití informačního systému je vyžadován pouze prohlížeč s podporou JavaScriptu a přístup k internetu.

Pro vývoj informačního systému bylo využito několik rozdílných technologií, podrobněji popsanych v následujících kapitolách.

3.1 PHP

Veškerá logika informačního systému je naprogramována ve skriptovacím jazyce PHP [4]. Jazyk je serverově orientovaný a všechny výpočty a komunikace s databází probíhá na straně serveru, což přináší výhodu pro uživatele, který obdrží od serveru již vytvořený zdrojový kód webové stránky a tu pouze vykreslí webovým prohlížečem.

Proměnné jsou v jazyce PHP poněkud odlišného charakteru, než je tomu v jazycích typu C a Java. Označuje se jako netypový jazyk. Příčinou je slabá typová kontrola, což znamená, že před užitím nemusíme proměnnou deklarovat. Nemusíme deklarovat ani její typ. Typ proměnné tedy můžeme změnit podle toho, jakou hodnotu v ní potřebujeme uložit. Ke správě proměnných se používají tři základní jazykové konstrukce.

- Ověřovat, zda určitá proměnná existuje obstarává funkce `isset()`.
- Odstranit proměnnou obstarává funkce `unset()`.
- Ověřovat pravdivost hodnoty proměnné obstarává funkce `empty()`.

Obecně platí, že jazyk PHP nepodporuje globální proměnné. Ovšem určité speciální interní proměnné se chovají jako globální proměnné, které známe z jiných jazyků. Tyto proměnné se v jazyce PHP nazývají superglobálními a byly předdefinovány pro zjednodušení práce s jazykem PHP. Jedním z příkladů superglobálních proměnných využitých v systému jsou `$_GET[]` a `$_POST[]`. Jsou to pole obsahující všechny proměnné, jež skript PHP získá z formuláře odeslaného prohlížečem metodou `GET` a `POST`.

Funkce v Jazyce PHP mohou být vestavěné, nebo uživatelsky definované. Počet argumentů může být u každé funkce jiný. Jako argument lze použít libovolný platný výraz včetně volání další funkce.

3.2 JavaScript

Jednou z nejdůležitějších vlastností JavaScriptu [13], od kterých se odvíjí i další řada vlastností, je fakt, že jde o programovací jazyk, jehož činnost zprostředkovává prohlížeč webových stránek návštěvníka stránky a ne server, na kterém jsou stránky uloženy. K provádění javascriptového kódu dochází na počítači návštěvníka stránky až po jejím načtení. JavaScript je tedy programovací jazyk, který je interpretovaný na straně klienta.

Z tohoto pro tvůrce stránek avšak vyplývá několik omezení. Kvůli zachování bezpečnosti nejsou zabudovány v JavaScriptu funkce, které by umožňovaly zapisovat a načítat údaje ze souborů. Pro návštěvníka stránky je to pozitivum, jelikož se nemusí obávat toho, že by mu někdo pomocí tohoto jazyka mohl dokázat smazat obsah disku nebo zneužít diskrétní informace.

Nejčastější použití JavaScriptu:

- Záměna obrázku na stránce v závislosti na poloze kurzoru myši
- Vytvoření prvků ulehčující navigaci v podobě hierarchických a roletových nabídek.
- Kontrola správnosti údajů zadaných návštěvníkem stránky do formuláře a jejich korekce
- Práce s okny prohlížeče stránky.
- Využití údajů o aktuálním datu a času a jejich zobrazení.

3.3 HTML, CSS

Uživatel se prostřednictvím internetového prohlížeče dotazuje webového serveru a ten mu na základě jeho požadavku posílá data. Nejdůležitější na HTML je skutečnost, že je založen na značkách (anglicky tag) (v odborné terminologii se setkáme jak se slovem tag, převzatým z angličtiny, tak s českým názvem značka). Každá značka představuje jeden příkaz. Tyto příkazy se uzavírají do ostrých závorek. Značky však mohou obsahovat parametry. Tyto parametry upřesňují význam značek. Každá parametr má svoji hodnotu, která se mu přiděluje pomocí rovnítka. [5]

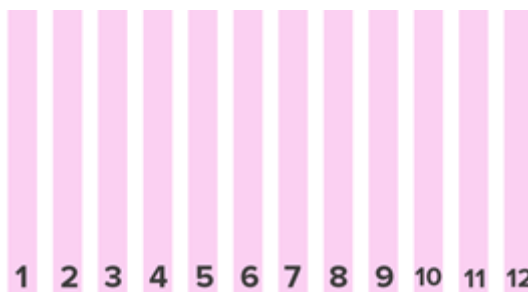
Způsob, jak ovlivňovat prezentaci dokumentu nebo kolekce dokumentu představují kaskádové stylové předpisy, anglicky Cascading Style Sheets (CSS). Kaskádové styly by sami o sobě neměly být využity, bez dokumentu, který by prezentovaly. Dokument v této roli představuje HTML. CSS především umožňuje docílit mnohem propracovanějšího vzhledu dokumentu, než je možné prostřednictvím samotného HTML. Umožňuje nastavit barvy textu a pozadí pro jakýkoliv prvek, umožňuje vytvořit orámování okolo jakéhokoliv prvku, měnit vzhled textu co se týče velkých a malých písmen, dokonce i specifikovat, zda se nějaký text na stránce vůbec objeví. Stylové předpisy redukuje pracovní zatížení autora webu, soustřeďují také příkazy pro jisté vizuální efekty ne jediné místo, což znamená, že příkazy nejsou rozstrkané na nejrůznějších místech dokumentu. Přidané prvky kaskádových stylů, mezi které patří `p`, `table`, `span`, či `div`, tvoří základní strukturu dokumentu. Každý prvek generuje box obsahující obsah daného prvku.

Pomocí CSS můžeme nejenže soustředit všechny informace o stylech stránky na jediné místo, můžeme také vytvořit stylový předpis, který je možné aplikovat na více stránek.

Kromě vizuální síly CSS a jeho výbavy, která umožňuje posílit autora i čtenáře stránky, vede použití CSS totiž k podstatně nižší velikosti HTML dokumentů, čímž urychluje jejich načítání. [10]

3.4 Bootstrap

V současné době je Bootstrap [12] jeden z nejpoblárnějších HTML, CSS a JS frameworků pro vývoj responzivních webových stránek. Značně urychluje a usnadňuje vývoj front-end webových aplikací. Nejdůležitější využití frameworku Bootstrap je přizpůsobení webové stránky zařízením s různou velikostí a poměrem stran displeje. Obsahuje vlastní soubor s kaskádovými styly, které lze snadno využít při tvorbě nových projektů a na kterém lze snadno stavět vlastní zdrojový kód. Kromě sady zdrojových kódů usnadňujících návrh vzhledu aplikací je součástí frameworku spousta vlastních komponent a jQuery plug-inů. Důležitou vlastností je možnost nastavovat rozložení zobrazení. Bootstrap pracuje s logikou rozdělení obrazu na 12 sloupců, viditelných na obrázku 3.1.



Obrázek 3.1: Ukázka rozložení sloupců [3]

Jednotlivé sloupce jsou postupně přizpůsobovány podle zdrojového kódu, kde `col-xs` označuje nastavení pro nejmenší typy displejů, `col-sm` následně pro střední a `col-lg` pro velké.

3.5 FullCalendar

Pro správu dovolených je potřeba využít přehledného kalendáře, ve kterém se zobrazí schválené dovolené zaměstnanců. Pro potřebu informačního systému je využít JavaScriptový kalendář událostí FullCalendar [2]. Kalendář zobrazuje předložené události, které lze získat z databáze a je tedy vhodným nástrojem pro práci s docházkou.

Plugin pracuje s několika parametry, kterými jsou `end`, `start`, `title` a `url`. Parametry popisují časový interval, kdy se událost v kalendáři zapíše, název zapsané události a adresa stránky, na kterou se po výběru dané události aktuální stránka přesměruje.

3.6 DateTimePicker

V informačním systému je použit Widget – `DateTimePicker` [9], díky kterému nemusí uživatel zadávat datum ručně. Při kliknutí na příslušné pole ve formuláři se zobrazí kalendář se dny v měsíci. Po výběru data může uživatel zvolit libovolný čas ze seznamu v intervalech půlhodin. Při zvolení data a času se do vybrané kolony vyplní zvolené možnosti.

Widget je přizpůsoben k práci s platformou Bootstrap, obsahuje vlastní soubor popisující JavaScriptové funkce a soubor s formátováním v podobě kaskádových stylů.

3.7 MySQL

V informačním systému je použit relační databázový systém MySQL. Databáze je uspořádaná množina dat, která je normálně uložena v jednom nebo několika datových souborech. Data jsou strukturována v tabulkách, s možnými referencemi mezi tabulkami. Existence těchto relací mezi tabulkami vedla k pojmu relační databáze. Mezi úkoly pro relační databázové systémy nepaří jen bezpečné skladování dat, ale také úkoly jako vykonávání příkazů pro vyhledávání, analyzování a třídění existujících dat, a také ukládání dat nových.

Každá řádka v tabulce relační databáze se nazývá datový záznam. Struktura každého záznamu je dána definicí tabulky.

Popis databáze složené z několika tabulek se všemi poli, relacemi a indexy se jmenuje databázový model. Tento model definuje systém datových struktur a také formát, ve kterém budou konkrétní data uložena. Tabulky obsahují data v náhodném pořadí. Pro efektivní využití dat je však nutné z neuspořádaných dat vytvořit podle jednoho nebo několika polí uspořádaný seznam. Například můžeme získat seznam zákazníků seřazených podle směrovacího čísla, kteří si v posledních dvanácti měsících objednali některou knihu.

Abychom získali takový seznam, vytvoříme dotaz. Výsledkem dotazu je opět tabulka, avšak v tomto případě v operační paměti (RAM), nikoli na pevném disku.

Pro formulaci dotazu se používají instrukce SQL pro výběr a selekci dat. SQL je zkratka ze slov „Structured Query Language“ (strukturovací dotazovací jazyk), který se stal standardem pro formulaci databázových dotazů.

Když jsou tabulky velké, rychlost, s jakou jsou dotazy zodpovězeny, velmi závisí na existenci vhodného indexu který obsahuje seřazená pole dat. Index se také nazývá klíč.

Tento index zrychluje přístup k datům, má však také nevýhody. Za prvé, každý index zvětšuje velikost datových souborů uložených na pevném disku, a za druhé index musíme při každém vkládání dat aktualizovat, a to stojí čas. Primární index nebo primární klíč je speciálním případem indexu, u něhož je zajištěna jedinečnost identifikace záznamu. Primární indexy hrají důležitou roli v relačních databázích, s jejich pomocí lze značně zrychlit přístup k datům.

Mezi nejdůležitější vlastnosti MySQL patří:

- Relační databázový systém.
- Architektura klient/server.
- Kompatibilita se SQL.
- Vnořené dotazy.
- Mnoho pohodlných uživatelských rozhraní.
- Podpora spousty programovacích jazyků.
- Nezávislé na platformě.
- Rychlost.

Další věcí na MySQL je licence. MySQL je open source projekt, tedy kompletní zdrojový text MySQL je volně k dispozici a je dostupný pod licencí GNU General Public License (GPO), avšak i i ta je nedílně spojena s jistými podmínkami. [8]

PhpMyAdmin

Pro usnadnění správy obsahu MySQL je použit nástroj PhpMyAdmin vytvořený v jazyce PHP. Podporuje širokou škálu operací pro MySQL. Veškeré úkony lze provádět prostřednictvím uživatelského rozhraní, které má stále schopnost spouštět libovolný SQL příkaz.

Kapitola 4

Návrh informačního systému

Tato kapitola popisuje strukturu databáze, pomocí konceptuálního modelu 4.1. V následujících kapitolách je popsán návrh jednotlivých částí informačního systému. Tento popis vycházel z návrhu ER diagramu jenž je znázorněn v příloze D.

4.1 Konceptuální model

Konceptuální model představuje úplnou a přesnou reprezentaci datových požadavků organizace, kterou má databáze podporovat. Model se zkontroluje, aby se zajistilo, že redundance je minimální a model je schopen podporovat požadované transakce.

Abychom si byli jisti, že přesně rozumíme povaze dat a jejich používání v organizaci, potřebujeme model pro komunikaci, který není technický a neobsahuje víceznačnost. Jedním příkladem takového modelu je entitně-relační model, kde entita je množina objektů se shodnými vlastnostmi, které uživatel nebo organizace identifikuje jako nezávisle existující objekty. Entitně-relační modelování návrhu databáze odpovídá přístupu k návrhu metodou shora dolů. ER modelování začíná určením důležitých dat a relací mezi daty, které je třeba v modelu reprezentovat. Pak se postupuje k podrobnostem jako například informacím, které je třeba o entitách a vztazích uchovávat a omezením platným pro entity, vztahy a atributy.

Základním konceptem ER modelování je entita (množina objektů se shodnými vlastnostmi, které uživatel nebo organizace identifikuje jako nezávisle existující objekty), která představuje množinu objektů skutečného světa „skutečného světa“, které mají shodné vlastnosti. Každý objekt, který by mělo být možné jedinečně identifikovat v rámci této množiny, se nazývá výskyt entity. Entita existuje nezávisle a může představovat objekt s fyzickou existencí nebo objekt existující pojmově.

Každou entitu identifikujeme jedinečným jménem a seznamem vlastností, nazývaných atributy. Ačkoli má entita jedinečnou množinu atributů, každá entita má vlastní hodnoty pro každý atribut. Databáze obvykle obsahuje mnoho různých entit. [1]

4.2 Zaměstnanci

U zaměstnance uchovává uchovává jméno, příjmení, oprávnění určující pravomoci, unikátní login, email pro zaslání notifikací, zaměstnancovu pozici a heslo k účtu. Dále se uchovává ID oddělení, do kterého zaměstnanec patří. Zde jsou tedy uchovány veškeré informace o všech uživateli systému.

Zaměstnancem je kterákoliv osoba přihlášená do systému, o rozdílech mezi jednotlivými zaměstnanci rozhoduje hodnota oprávnění. Oprávnění jsou rozdělena do 4 kategorií, dle uživatelských rolí. Nově vytvořený uživatel má automaticky přiřazení základní oprávnění, které následně může změnit pouze administrátor ručně, nebo přiřazením zaměstnance k oddělení, čímž zaměstnanec získá automaticky oprávnění „2“.

Popis oprávnění:

- „1“ - Běžný uživatel: Uživatel má přístup k vytváření/editaci úkolů a požadavků v nich a žádostem o dovolenou
- „2“ - Uživatel s možností přidávání aktualit: Uživatel s rozšířením pravomocí o možnost vytváření aktualit.
- „3“ - Ředitel: Uživatel, který má navíc k dispozici správu dovolených.
- „4“ - Administrátor: Uživatel, který má k dispozici všechny výše zmíněné funkce, navíc jako jediný může zakládat a editovat jednotlivá oddělení.

Vztahy zaměstnance:

- Může podat žádost o dovolenou.
- Může vést oddělení.
- Patří do jednoho z oddělení.
- Může mít oprávnění k úkolu.
- Může vytvořit úkol.
- Může být odpovědný za úkol.
- Může vytvořit komentář k přiřazenému úkolu.
- Může editovat vytvořený úkol, u kterého má odpovědnost či autorství.

Formulář pro vytvoření zaměstnance:

- Jméno zaměstnance.
- Příjmení zaměstnance.
- Login (Jedinečný).
- Oprávnění (Výběr ze seznamu).
- Oddělení (Výběr ze seznamu).
- Název pozice.
- Email.

4.3 Oddělení

U oddělení je uchováván jeho název, stručný popis oddělení a také ID vedoucího oddělení. Oddělení vytváří a edituje pouze administrátor.

Vztahy oddělení:

- Může mít oprávnění k úkolu.

Formulář pro vytvoření oddělení:

- Název oddělení.
- Popis oddělení.
- Vedoucí oddělení (Výběr ze seznamu).

4.4 Úkoly

U úkolu je uchováván jeho název, termín zadání a ukončení, stav rozpracovanosti, popis úkolu a jeho priorita. Dále je potřeba uchovat informaci o zaměstnanci, který úkol vytvořil a zaměstnanci odpovědného za úkol. Editovat úkol může pouze jeho autor, nebo zaměstnanec odpovědný za úkol.

Variace možných stavů:

- Návrh
- Příprava
- Řešení
- Dokončení

Variace možných priorit:

- Důležité
- Spěchá
- Nespěchá
- Banalita

Vztahy úkolu:

- Může obsahovat požadavek.
- Může obsahovat komentář.
- Obsahuje oprávnění k zaměstnanci nebo oddělení.

Formulář pro vytvoření úkolu:

- Název úkolu.
- Termín dokončení (Výběr data i času z kalendáře).
- Popis úkolu.
- Stav úkolu (Výběr ze seznamu).
- Priorita úkolu (Výběr ze seznamu).
- Odpovědný zaměstnanec (Výběr ze seznamu).

4.5 Požadavky

Úkol může obsahovat seznam požadavků, které se skládají z názvu, stavu, termínu ukončení a popisu. Úkoly a požadavky lze komentovat. Požadavek se může nacházet v jednom ze stavů, které mohou nastat u úkolu.

Formulář pro vytvoření úkolu:

- Název požadavku.
- Termín dokončení (Výběr data i času z kalendáře).
- Popis požadavku.
- Stav úkolu (Výběr ze seznamu).

4.6 Komentář

U komentáře je potřeba uchovat pouze ID zaměstnance, který jej vytvořil, datum vytvoření a znění komentáře. Komentář může smazat pouze jeho autor.

Formulář pro vytvoření komentáře:

- Znění komentáře.

4.7 Oprávnění úkolů

Na základě přiřazování jednotlivých úkolů je potřeba uchovávat jednotlivá oprávnění k úkolům. Tato oprávnění se skládají z ID úkolu a následně ID zaměstnance, nebo ID oddělení podle požadavku na přiřazení úkolu zaměstnanci nebo celému oddělení. Oprávnění se ruší společně se smazáním úkolu, nebo při editaci úkolů každé zvlášť.

4.8 Žádost o dovolenou

U žádosti o dovolenou je uchována informace o autoru, časové rozpětí dovolené, důvod dovolené a v případě zamítnutí důvod zamítnutí. Dále je také potřeba uchovat stav, ve kterém se daná žádost nachází.

Stav žádosti:

- Čekající – Žádost se po vytvoření nachází v tomto stavu.
- Schválená – Pokud ředitel schválí dovolenou, ta se následně zapíše do kalendáře dovolených.
- Neschválená – Zamítnutá žádost, kterou uvidí žadatel v seznamu svých žádosti. Neschválená žádost uchovává informace o důvodu zamítnutí.

Formulář pro vytvoření žádosti o dovolenou:

- Od (datum).
- Do (datum).
- Důvod.

4.9 Aktuality

U aktuality je potřeba uchovávat její název, datum vystavení a obsah této aktuality. Dále se uchovává ID zaměstnance, který aktualitu vytvořil. Vytvářet novou aktualitu a editovat kteroukoliv bude moci zaměstnanec s oprávněním 2 a vyšším.

Formulář pro vytvoření aktuality:

- Název aktuality.
- Obsah aktuality.

Kapitola 5

Implementace

V následující kapitole jsou popsány funkce nejdůležitějších prvků v systému. Patří mezi ně návrh webové stránky pomocí frameworku Bootstrap, autentizace uživatele při přihlášení do systému, šifrování dat v databázi ochrana proti SQL injection, popis hlavních funkcí naprogramovaných v PHP spolu s popisem uživatelského rozhraní.

5.1 Popis uživatelského rozhraní

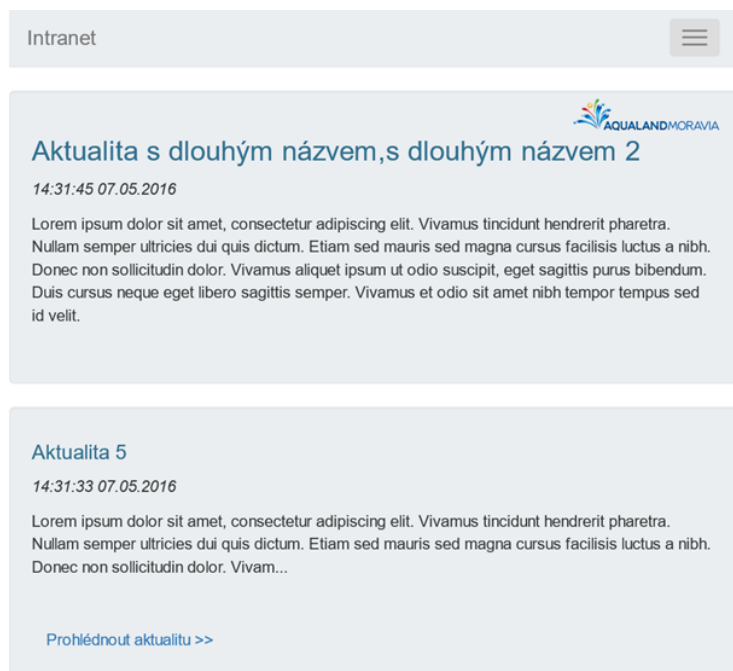
Při návrhu uživatelského rozhraní je kladen důraz na co nejjednodušší užívání systému. Hlavním cílem je jednoduchý styl. Celý systém obsahuje pouze několik barev, které dobře vystihují kontrast zobrazené informace.

Design je vytvořen tak, aby bylo snadné přizpůsobovat prvky podle velikosti zobrazovacího zařízení, o které se stará technologie Bootstrap 3.4. Je tedy důležité myslet na to, že stránka není pouze v jednom zobrazení, ale rovnou ve 3 rozdílných, podle toho, na jakém zařízení stránku zobrazíme. Na obrázku 5.1 je vidět rozložení a vzhled hlavní stránky na běžných stolních počítačích, nebo laptotech.



Obrázek 5.1: Zobrazení na velkém displeji

Na obrázku 5.2 dále vidíme přizpůsobení horního menu mobilnímu zařízení a přeskládání obsahu celé stránky podle velikosti zobrazovacího okna, tedy v případě hlavní stránky je zobrazena pouze 1 aktualita na řádku místo 3, které byly na obrázku 5.1.



Obrázek 5.2: Zobrazení na malém displeji

Veškeré formuláře pro editování, nebo přidávání nových položek, jsou znázorněny na obrázku 5.3. Formuláře se ve výchozím stavu nachází skryty a zobrazují se po stisknutí nadpisu „Přidat požadavek“. V textových polích je jednoznačně popsáno, která informace se v něm má nacházet. Dále je zde pro zjednodušení zadávání dat při kliknutí do pole „Termín dokončení“ zobrazen kalendář, ve kterém se nastavuje datum i čas.

Po	Út	St	Čt	Pá	So	Ne
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Obrázek 5.3: Formulář s DateTimePicker

Jednotlivé úkoly se zobrazují v miniaturách na stránce úkolů. Miniatury jsou opět přizpůsobeny pro jednotlivá zařízení, tedy na velkých displejích se zobrazují 4 úkoly vedle

sebe, na středních 3 a na malých pouze 1 úkol. Úkoly jsou barevně rozlišeny podle důležitosti daného úkoly. Vzhled úkolů je možné vidět na obrázku 5.4. Zde je možné vidět barevné rozložení, kde červená barva zdůrazňuje velmi důležitý úkol, oranžová barva již méně důležitější úkol a světlá nejméně důležitý.



Obrázek 5.4: Zobrazení úkolů

5.2 Zabezpečení

Zabezpečení je podstatnou součástí každého systému, který může obsahovat citlivé informace. Tématu o zabezpečení se věnuje právě tato podkapitola, která přibližuje využití technologie při šifrování dat, nebo ověření přihlášení uživatele do systému.

5.2.1 Autentizace

Přihlášení uživatelů do informačního systému probíhá na základě autentizace uživatelského jména a hesla. Tyto dva údaje jsou následně porovnávány s databází, ve které je uložen jejich správný protějšek. Na základě porovnání je přístup do systému následně povolen nebo zamítnut.

Z důvodu zvýšení úrovně bezpečnosti je heslo uloženo v databázi v zakódovaném řetězci. Heslo je zabezpečeno pomocí Secure Hash Algorithm [7], konkrétně SHA-512, který se v dnešní době považuje stále za bezpečné řešení. Řetězec je složen ze zakódovaného hesla a následně zakódovaného řetězce, který byl náhodně vygenerovaný při vytvoření aktuálního hesla a následně také uložen v databázi.

Po úspěšném porovnání uživatelského jména a hesla je uživateli umožněn přístup do systému. Při přihlášení je současně vytvořena relace, která uchovává informaci o přihlášeném uživateli a čas jeho aktivity v systému. Na základě dlouhé neaktivity je uživatel po uplynutí 10 minut automaticky odhlášen ze systému. Tento krok také napomáhá ke zvýšení zabezpečení systému.

5.2.2 Šifrování dat

V rámci zabezpečení soukromých informací uložených v databázi dochází k šifrování pro případ, že by se útočník zmocnil samotné databáze.

Pro šifrování je zde využita knihovna `mcrypt` [11], která je od verze PHP 5.3.X implementována. Data se před vložením do databáze nejdříve zašifrují.

Šifrování dat probíhá následně:

1. Použijeme námi zvolený klíč, v hašované podobě (SHA-256).
2. Zvolíme mód pro šifrování dat, který je následně využit i při dešifrování dat.
3. Na základě zvolených módů získáme velikost vektoru z algoritmu specifikovaným šifrovacím deskriptorem v bytech pomocí funkce `mcrypt_get_iv_size`.
4. V dalším kroku se provede inicializace vektoru na základě získané velikosti vektoru pomocí funkce `mcrypt_create_iv`.
5. V posledním kroku se na základě vektoru, klíče a opět zvolených módů data zašifrují pomocí funkce `mcrypt_encrypt`.

Dešifrování dat:

1. Opět se použije stejný klíč použit při šifrování dat.
2. Získáme velikost vektoru z algoritmu specifikovaným šifrovacím deskriptorem v bytech opět pomocí funkce `mcrypt_get_iv_size`.
3. Rozdělíme vstupní šifrovaný řetězec na základě délky vektoru do 2 částí. Jednou částí je vektor a druhou je šifra. V tomto kroku poslouží funkce `substr`
4. V posledním kroku se na základě získaného vektoru, klíče a opět zvolených módů šifra dešifruje a to pomocí funkce `mcrypt_decrypt`.

5.2.3 SQL injection

Přímý SQL injection příkaz je technika, kdy útočník vytváří, nebo mění stávající SQL příkazy k odhalení skrytých dat, změně, nebo smazání dokonce i celé databáze. Může také dokonce spouštět nebezpečné systémové příkazy na úrovni hostitele databáze.

Ochranou systému před SQL injection je filtrování všech jednotlivých vstupů a dotazů. Ochranu zajišťuje funkce `checkvar`, která při volání metod `GET` a `POST` kontroluje obsah jednotlivých proměnných, které obsahují metody při jejich využití. Funkce prochází jednotlivé obsažené řetězce a v případě výskytu znaků `<`, `>`, `\\`, `\`, `'` znaky maže.

5.3 Oprávnění

Vzhledem k rozdílným rolím uživatelů v systému je potřeba správně zvolit jednotlivá oprávnění přístupů. Každý uživatel má své přiřazené oprávnění na základě své funkce, popřípadě nastavení administrátorem. Každá z částí informačního systému prověřuje, zda dotyčný uživatel má na zobrazení stránky právo. Účel plní funkce `opraveni`. Jednotlivý uživatelé po přihlášení do systému vidí, které sekce jsou pro něj dostupné, tak, jak je to znázorněno na obrázku 5.5. Vstupním parametrem pro ověření oprávnění je pole jednotlivých oprávnění, pokud pole obsahuje oprávnění 1-4, k dané stránce se může dostat kterýkoliv přihlášený zaměstnanec. Pokud by se neoprávněný uživatel pokoušel o přístup na stránku, dojde k jeho odhlášení ze systému a následně přesměrování na `index.php`. Základní oprávnění jsou rozděleny do několika kategorií popsanych v kapitole zaměstnanci 4.2.



Obrázek 5.5: Zobrazení administrátorských oprávnění

Kromě zmíněných oprávnění je potřeba ověřovat jednotlivé přístupy k úkolům, které nejsou popsány v jednotlivých kategoriích oprávnění, proto každý přístup k jednotlivým detailům úkolů je provázen dodatečně zpětným ověřením, zda přihlášený uživatel je zapsán v databázi v tabulce oprávnění k úkolům.

Další z potřebných ověření oprávnění probíhá u vedoucího zaměstnance. Zaměstnanec má k dispozici přístup k editaci a vytváření nových zaměstnanců do svého oddělení a také vidí základní informace všech úkolů přiřazených zaměstnancům svého oddělení. Rozdíl mezi administrátorem a vedoucím oddělení je určen na základě informací o vedoucích oddělení. Pokud je v databázi informace o zaměstnanci jako vedoucím oddělení, je mu umožněno editovat právě zaměstnance zmíněného oddělení, avšak za podmínky, pokud jeho oprávnění není administrátor.

5.4 Adresářová struktura

Zdrojové kódy využitý v informačním systému jsou rozděleny do několika adresářů podle jejich využití. Hlavní soubory, ve kterých je popsán kód k zobrazení webové stránky v HTML, je umístěn ve zdrojovém adresáři.

Adresář /PHP obsahuje především podpůrné zdrojové kódy napsány v PHP, které obsahují například metody GET a POST, se kterými se pracuje na většině stránek. V podadresáři odpovídajícího názvu se nachází část pro plugin FullCalendar. Nejdůležitější součástí logiky celého systému je soubor, nacházející se v tomto adresáři, `functions.php`, ve kterém se nachází všechny PHP funkce použité v projektu.

Všechny soubory pracující s kaskádovými styly jsou umístěny v adresáři /css. Soubory slouží k nastavení vzhledu webové stránky a také některých plug-inů, jako jsou DatePicker a FullCalendar. V adresáři /js jsou umístěny soubory se zdrojovým kódem psaným v JavaScriptu, mezi které patří zdrojové kódy přidávaných plug-inů a jQuery. Ve složce /img se nalézají obrázky použité v celém systému.

5.5 Bootstrap

Při implementaci systému byly využity zajímavé funkce uvedeného frameworku, které by bylo nutné bez využití této technologie ručně programovat v jazyce JavaScript. Bootstrap nabízí spousty již přednastavených možností k využití tlačítek, jednotlivých tabulek a kolonek pro vkládání textu.

Mezi zajímavé použité funkce patří navigační lišta `navbar`, která se může v závislosti na obrazovce rozšiřovat, nebo se zabalit. Dalším prvkem využitým v informačním systému je `dropdown`, který po výběru zobrazí další možnosti, které byli doposud schované. Prvek

je využít v menu, tedy pokud například chci provést administrativní činnost, dostanu na výběr správu zaměstnanců, oddělení, nebo aktualit.

Zajímavou vlastností Bootstrapu je také změna vzhledu některých prvků. V systému jsou využity tlačítka, kterým je nastavena atribut `btn`. Dle potřebného vzhledu tlačítka jsou zde využity atributy upřesňující vzhled tohoto tlačítka. Těmito atributy jsou například `btn-info`, který zbarví tlačítko do modra, `btn-danger`, který zbarví tlačítko do červena a další. Další vlastností je možnost zobrazení přednastaveného pozadí jednotlivého prvku, například prvku `div`, čehož se docílí zadáním parametru `well`.

5.6 Popis hlavních částí

V následující podkapitole je popis nejdůležitějších stránek a modulů celého informačního systému. Každý popis stručně popisuje návaznost jednotlivých modulů v systému a následně využití důležitých funkcí.

5.6.1 Výpis aktualit

Hlavní stránkou informačního systému je `index.php`. Na této stránce jsou zobrazeny všechny aktuality, které byly vloženy do systému. Stránka je rozdělena na hlavní aktualitu, které odpovídá poslední zveřejněná zpráva výpis všech ostatních aktualit.

Výpis hlavní aktuality je popsán funkcí `hlavni_aktualita`. Funkce defaultně zobrazuje poslední vloženou aktualitu do systému, ovšem pokud uživatel zvolí u některé z ostatních aktualit možnost „prohlédnout aktualitu“, dojde k jejímu vypisání na místo hlavní aktuality.

Rozdíl mezi hlavní aktualitou a ostatními aktualitami, popsány funkcí `ostatni_aktuality`, je v omezeném počtu znaků v nadpise a textu. O omezení se stará JavaScript, ve kterém jsou napsány funkce `.limit` a `.limitTitle`. Funkce na nastavení limitu zjistí velikost vypisovaného řetězce, a pokud je větší než požadovaná délka, oříznou jej a na konec řetězce vypíše „...“. Funkce pro výpis ostatních aktualit také implementuje omezení vypsaných aktualit na stránku. Tento limit je 6. Při překročení limitu se další aktuality již nevypisují. Další aktuality je možné vypsat přepínačem na konci stránky, který se rozšiřuje podle celkového množství aktualit.

5.6.2 Zobrazení úkolů

Nejrozsáhlejší stránkou je `ukoly.php`. Stránka je složena z několika modulů. Jedná se o `ukoly-pozadavek.php` starající se o výpis a vytváření nových požadavků a `ukoly-vice.php` starající se o výpis detailu úkolu a přidávání komentářů.

Hlavní stránka funguje primárně jako řadič, tedy rozhoduje o tom, které informace se aktuálně vypíší. Stránka v defaultním stavu vypisuje všechny úkoly, jedná se o implementaci funkce `vypis_ukol`. V první řadě funkce ověřuje, zda má uživatel oprávnění k úkolu. Pokud ano, vypíše se uživateli základní informace o úkolu. Záleží mimo jiné na tom, zda je uživatel odpovědný za úkol, nebo jej vytvořil. Z pozice tvůrce má uživatel k dispozici možnost úkol smazat, nebo jej editovat. Následně si může vypsat po stisknutí tlačítka „Více“ detailní informace k úkolu.

Úkoly zobrazené uživateli je možné ovšem také řadit dle jednotlivých parametrů, popřípadě zobrazit jen požadované úkoly. K tomuto slouží možnost zvolit jeden z filterů, nebo seřazení v horní části stránky. O metodu vybrání jednotlivých dat z databáze, popřípadě jejich seřazení, se stará funkce `fillfilter`.

O výpis detailních informací se stará modul `ukoly-vice.php`. Náhled popisující modul vypisuje detailní informace o zvoleném úkolu. Pokud se v modulu nachází autor či zodpovědná osoba, má k dispozici editaci zvoleného úkolu. Mezi možnosti editace patří přiřazování dalších zaměstnanců k úkolu, řešeno funkcí `pridat_opravneni_zamestnanec`, rušení aktuálních oprávnění, řešeno funkcí `prirazene_opravneni`, nebo pověření úkolem celé oddělení pomocí funkce `pridat_opravneni_oddeleni`.

Poslední modul `ukoly-pozadavek.php` implementuje výpis všech požadavků k úkolu. Výpis je společně k dispozici s detailem úkolu. Pro výpis jednotlivých požadavků ke zvolenému úkolu je využita funkce `vypis_pozadavku`.

5.6.3 Administrace

Pro vytváření nových aktualit, zaměstnanců a oddělení je vytvořena sekce stránek administrace. Stránka `zamestnanci.php` plní funkci správy jednotlivých zaměstnanců. Zde je možné vytvářet, editovat, mazat ale také restartovat hesla jednotlivým zaměstnancům. Pro stránku zaměstnanců jsou využity funkce `vypis_zamestnanec`, která vypíše seznam jednotlivých zaměstnanců uložených v databázi, dále také `zmen_oddeleni` sloužící ke změně oddělení zaměstnance, ve kterém doposud byl evidovaný. Při vypsání jednotlivých zaměstnanců se po pravé straně zobrazí tlačítka, které umožní právě akci editace a smazání uživatele, nebo resetování hesla. Stránka `oddeleni.php` také spadá do sekce stránek administrace. Pro výpis jednotlivých oddělení je využita funkce `vypis_oddeleni`, která vypíše seznam jednotlivých oddělení. Opět se zde při vypsání zobrazí tlačítka, díky kterým je možné oddělení spravovat.

Každá akce v systému se pro potřebu administrátora ukládá v databázi do logu. K ukládání slouží funkce `insert_log`, která vloží řetězec o provedené události do databáze. Log s událostmi slouží následně pro administrátora a to pro dohledávání nesrovnalostí při vytváření a editaci některých úkolů v systému. Pro výpis seznamu událostí po sobě jdoucích je vytvořena stránka `admin_log.php`. Log je dostupný pouze pro administrátora a nikoho jiného.

5.6.4 Kalendář dovolených

Důležitou součástí informačního systému je kalendář dovolených, ve kterém je zobrazena veškerá schválená dovolená. Kalendář je implementován za pomoci pluginu FullCalendar. Celkový vzhled kalendáře je znázorněn na obrázku 5.6.

Celý kalendář je nastavený ve zdrojovém souboru s JavaScriptem `scripts.js`, nesoucí identifikátory `kalendar` a `kalendar_nahled`. Rozdíl mezi označením kalendář a kalendář náhled je pouze ve vynechání URL adresy při nastavení náhledu kalendáře. Tento náhled slouží pouze jako orientace pro zaměstnance. URL adresa slouží pro administraci dovolené, kde si může přímo v kalendáři uživatel s potřebným oprávněním zvolit jednotlivou událost, která uživatele přesměruje na editaci vybrané události. U editované události je možné měnit zpětně datum dovolené.

Na stránce `kalendar.php` probíhá správa docházek dovolených. Nachází se zde tabulka (Obr. 5.7), ve které je možné listovat všemi žádostmi. Tabulka se v defaultním stavu nachází na seznamu žádostí čekajících. Je možné libovolně pracovat s již schválenými, čekajícími, či zamítnutými žádostmi. U těchto žádostí je, krom data od kdy do kdy, také vždy zobrazena doba trvání dovolené v dnech. Jednotlivé sekce jsou barevně odlišeny, kde čekající žádosti mají barvu světle žlutou, schválené žádosti zelenou a neschválené červenou. Při zvolení

možnosti zamítnout žádost se zobrazí formulář, kam je možné doplnit důvod zamítnutí. Důvod si může následně žadatel zobrazit v seznamu svých žádostí.

Calendar view for květen 2016. Navigation: < > Nyní. View options: Měsíc Týden Den.

po	út	st	čt	pá	so	ne
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
Tomáš Novák		Lukáš Dvořák				
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Obrázek 5.6: Vzhled pluginu FullCalendar

Neschválené žádosti ▾

Jméno	Trvání	Důvod	Akce
Tomáš Novák	18.05.2016 19.05.2016 2 dny	Nemoc	Změnit stav ▾
Tomáš Novák	21.05.2016 24.05.2016 4 dny	Dovolená	Změnit stav ▾

Obrázek 5.7: Správa dovolených

Kapitola 6

Testování

Důležitou rolí na závěr tvorby informačního systému je testování. Je nutné správné ověření veškerých funkcí systému, chování na různé vstupy uživatelů a celkovou funkčnost systému.

Testování je rozdělené na tři části. První část probíhá lokálně při vývoji systému, kde jsou postupně testovány reakce systému při implementaci jednotlivých funkcí. Zde je potřeba otestovat možné stavy a ověřit jejich správnou funkčnost a také vzhled jednotlivých prvků stránky pro jednotlivá zařízení v jednotlivých webových prohlížečích. Některé z webových prohlížečů přizpůsobují rozložení zobrazení podle přednastavené hodnoty. Tímto způsobem bylo testováno správné zobrazování na rozdílných velikostech displejů. V druhé části je primárně prováděno testování systému jako celku, tedy testování s důrazem na propojení jednotlivých funkcí. Třetí a poslední částí je ověření, zda systém dokáže ovládat i člověk, který nemá k funkčnosti systému zkušenosti a ověřit možnost intuitivního ovládání. Testované osoby byli zaměstnanci Aqualandu Moravia, jedná se o jedince seznámené s organizační strukturou společnosti.

Všichni dotázaní zvládli základní úkony s informačním systémem. Šlo o vytvoření a delegování úkolu a vytvoření žádosti o dovolenou. Většina z respondentů následně ocenila intuitivní a přehledné ovládání systému a především možnost využití systému na mobilních zařízeních.

Kritické ohlasy zazněly u ostatních přiřazených zaměstnanců k úkolu, kteří neměli plné oprávnění editovat úkol, a byli nuceni o splnění celého úkolu dát informaci pověřenému zaměstnanci, či autoru úkolu. Avšak i zde respondenti nakonec ocenili možnosti vkládání komentářů k úkolu a popřípadě plnou pravomoc k vytváření a editaci požadavků, ze kterých může vedoucí úkolu rozpoznat, zda je úkol hotový.

Při testování některých funkcí pro nadřazené zaměstnance byla potřeba menší instruktaž, ve které bylo popsáno vytváření a editace zaměstnanců a jejich pověření. Po instruktaži se prokázalo, že většina nadřízených zaměstnanců neměla problém s vytvářením a správou svých podřízených. V poslední části byla předvedena administrace dovolených řediteli, který byl s funkcemi bez výhrad spokojen.

Na závěr testování vyplynulo, že aktuality zveřejněny pro zaměstnance by neměly být úplně volně dostupné, tedy se rozhodlo o vytvoření společného účtu pro zaměstnance, kteří budou moci přistupovat pouze k aktualitám.

Kapitola 7

Závěr

Tvorba informačních systému je v dnešní době poměrně běžnou činností, nicméně velice žádanou, protože každá společnost má na informační systém své vlastní požadavky. Hlavním řešeným požadavkem v systému je optimalizace pro mobilní zařízení, pro kterou byla využita platforma Bootstrap, která je v dnešní době velice moderní a používanou metodou pro tvorbu webových stránek.

Výsledkem předložené práce je systém, který sjednocuje více doposud neřešených úkonů zaměstnanců do jednoho systému a nahrazuje činnosti, které jsou delegovány a řešeny pomocí emailové komunikace. Jedná se například o žádosti a viditelný seznam žádostí dovolených v jednom kalendáři, a to nejen pro ředitele, ale i zaměstnance, kteří si chtějí požádat o dovolenou. Zaměstnanec si tak například může naplánovat svou dovolenou s vazbou na počet lidí, kteří mají v daný termín již dovolenou schválenou. Dále nahrazuje informování zaměstnanců akvaparku, které doposud bylo řešeno pouze nástěnkou na místě pracoviště. Nově si mohou zaměstnanci přečíst aktuality i z pohodlí domova a na cestách.

V následujících měsících bude Aqualand Moravia přecházet na tento systém a současně se tento systém bude vyvíjet pro další možné funkce a vylepšení možností systému. Tímto procesem je navíc velká možnost vylepšení a úprav stavů, které nebyli odhaleny při vývoji a testování funkčnosti systému.

7.1 Uvažovaná vylepšení

Informační systém byl primárně vytvářen pro komunikaci mezi zaměstnanci top managementu firmy. Zaměstnanci bez přihlášení mají z informačního systému přístup pouze k aktualitám v textovém rozsahu, nicméně do budoucna je v úvaze rozdělení sekce aktualit do více podsekcí, kterou bude například galerie obrázků interního rázu. Navíc, jak vyplynulo z testování, se vytvoří pro všechny zaměstnance společný login a heslo, které budou interního rázu a zaměstnanci s těmito údaji se následně dostanou pouze na stránku se základními údaji.

Další plánovanou sekcí, je sekce s kontakty pro potřebu zaměstnanců, jako je kontakt na personální oddělení, pracovní agenturu a další. V úvahu připadá do budoucna přidání možnosti výběru barevného zobrazení stránky. Celkový systém bude v budoucnu doplněn také o notifikační zprávy k jednotlivým akcím, které budou zasílány na emaily jednotlivých uživatelů.

Literatura

- [1] Conolly, T.; Begg, C.; Hlowczak, R.: *Mistrovství - databáze*. Computer Press, a.s, 2009, ISBN 978-80-251-2328-7.
- [2] FullCalendar LLC: *FullCalendar documentation*. online, 2016, [Online; navštíveno 10.4.2016].
URL <http://fullcalendar.io>
- [3] Girl Develop It: *Intro to Bootstrap 3*. online, 2016, [Online; navštíveno 11.5.2016].
URL <http://gdiboulder.github.io/gdi-boulder-intro-bootstrap/#/29>
- [4] Gutmans, A.; Bakken, S. S.; Rethans, D.: *Mistrovství v PHP 5*. CP Books, a.s., 2005, ISBN 80-251-0799-X.
- [5] Hauser, M.; Houser, T.; Wenz, C.: *HTML a CSS Velká kniha řešení*. Computer Press, a.s, 2006, ISBN 80-251-1117-2.
- [6] Hruška, T.; Křivka, Z.: *Informační systémy. Studijní opora*. FIT VUT v Brně, 2012, [Online; navštíveno 9.5.2016].
URL <https://www.fit.vutbr.cz/study/courses/WAP/private/opory/OporaIIS2PISPojemDataProcesyTransakce.pdf>
- [7] Internet Engineering Task Force: *US Secure Hash Algorithms*. online, 2011, [Online; navštíveno 4.5.2016].
URL <https://tools.ietf.org/html/rfc6234>
- [8] Kofler, M.: *Mistrovství v MySQL 5*. Computer Press, a.s, 2007, ISBN 978-80-251-1502-2.
- [9] Malot, S.: *Bootstrap-datetimepicker project*. online, 2016, [Online; navštíveno 11.4.2016].
URL <http://www.malot.fr/bootstrap-datetimepicker>
- [10] Meyer, E. A.: *CSS kompletní průvodce*. ZONER software, s.r.o., 2007, ISBN 978-80-86815-64-0.
- [11] The PHP Group: *Mcrypt documentation*. online, 2016, [Online; navštíveno 10.5.2016].
URL <http://php.net/manual/en/book.mcrypt.php>
- [12] WWW stránky: *Bootstrap documentation*. online, 2012, [Online; navštíveno 27.4.2016].
URL <http://getbootstrap.com>

- [13] Škultéty, R.: *JavaScript: Programujeme internetové aplikace*. Computer Press, a.s, 2004, ISBN 80-251-0144-4.

Přílohy

Seznam příloh

A	Obsah CD	32
B	Manual	33
C	Use Case Diagram	39
D	ERD	40
E	Zhodnocení vedoucího IT Aqualandu Moravia.	41

Příloha A

Obsah CD

- doc/ # dokumentace a manuál
- db/ # databázové skripty
- src/ # zdrojové kódy informačního systému
 - css/ # kaskádové styly
 - img/ # obrázky používané v systému
 - js/ # javascriptové funkce
 - PHP/ # pomocné PHP funkce
 - index.php # hlavní skript

Příloha B

Manual

Připojení k databázi

Připojení do databáze v PHP kódu je řešeno na začátku souboru s funkcemi functions.php, který se nachází ve složce src/PHP/.

Přihlášení do systému

- Přihlášení do informačního systému se provádí tlačítkem „přihlásit se“ v horním menu po pravé straně. Pro úspěšné přihlášení je nutné přihlašovací jméno a heslo, které bylo vytvořeno administrátorem, nebo vedoucím zaměstnancem oddělení. Při prvním použití systému je vytvořen administrátorský účet, na který je možné se přihlásit zadáním přihlašovacího jména „admin“ a hesla „admin123“.
- V případě mobilních zařízení je nejdříve potřeba označit ikonu po pravé straně horního menu, na základě které se zobrazí celé menu i s nabídkou možnosti přihlášení.

Vytvoření nového uživatele

- Vytvoření nového uživatele provádí administrátor nebo vedoucí oddělení, který má ovšem pravomoc vytvářet uživatele spadající pouze do jeho oddělení.
- Vytvoření je možné zvolením sekce „Zaměstnanci“ v záložce na horní liště „Administrace“. Po zobrazení tabulky aktuálních zaměstnanců se objeví v horní části rolovací část „Přidat zaměstnance“, která po zobrazení nabízí formulář pro vytvoření zaměstnance. Pro úspěšné vytvoření zaměstnance je potřeba vyplnit všechny zobrazené kolonky a zvolit tlačítko „Vytvořit zaměstnance“.

Editace uživatele

- Editaci uživatele provádí administrátor nebo vedoucí oddělení, který má ovšem pravomoc editovat uživatele spadající pouze do jeho oddělení.
- Editace je možná zvolením sekce „Zaměstnanci“ v záložce na horní liště „Administrace“. Po zobrazení tabulky aktuálních zaměstnanců se objeví modré tlačítko „Upravit“, nachází se po pravé straně jednotlivých uživatelů. Při zvolení možnosti editace se zobrazí formulář s jeho údaji, který je možný editovat. Potvrzení změn editace se provede zvolením tlačítka „Upravit zaměstnance“.

Smazání uživatele

- Smazání uživatele provádí administrátor nebo vedoucí oddělení, který má ovšem pravomoc smazat uživatele spadající pouze do jeho oddělení.
- Smazání je možné zvolením sekce „Zaměstnanci“ v záložce na horní liště „Administrace“. Po zobrazení tabulky aktuálních zaměstnanců se objeví červené tlačítko po pravé straně jednotlivých uživatelů „Smazat“. Při zvolení možnosti se zobrazí potvrzovací tabulka pro provedení změny. Potvrzení smazání zaměstnance se provede opětovným zvolením tlačítka „Smazat“.

Vytvoření oddělení

- Vytvoření nového oddělení provádí pouze administrátor.
- Vytvoření je možné zvolením sekce „Oddělení“ v záložce na horní liště „Administrace“. Po zobrazení tabulky aktuálních oddělení se objeví v horní části rolovací část „Přidat oddělení“, která po zobrazení nabízí formulář pro vytvoření nového oddělení. Pro úspěšné vytvoření oddělení je potřeba vyplnit všechny zobrazené kolonky, zvolit zaměstnance, který od vytvoření oddělení bude vystupovat jako vedoucí oddělení a zvolit tlačítko „Vytvořit oddělení“.

Přiřazení oprávnění zaměstnance

- Přiřazení oprávnění uživatele provádí pouze administrátor.
- Přiřazení je možné zvolením sekce „Zaměstnanci“ v záložce na horní liště „Administrace“. Po zobrazení tabulky aktuálních zaměstnanců se objeví modré tlačítko „Upravit“, nachází se po pravé straně jednotlivých uživatelů. Při zvolení možnosti se zobrazí formulář s jeho údaji, který je možný editovat. Ve formuláři se nachází kolonka oprávnění, na základě které je přiřazeno zaměstnanci oprávnění. Potvrzení změn editace se provede zvolením tlačítka „Upravit zaměstnance“.
- Oprávnění:
 1. pouze přístup k úkolům a žádostem o dovolenou
 2. možnost vytvářet nové aktuality
 3. správa docházky
 4. umožní přístup ke všem funkcím systému

Vytvoření úkolu

- Vytvoření nového úkolu může provést kterýkoliv uživatel systému.
- Vytvoření je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se objeví v horní části rolovací část „Přidat úkol“, která po zobrazení nabízí formulář pro vytvoření úkolu. Pro úspěšné vytvoření úkolu je potřeba vyplnit všechny zobrazené kolonky a zvolit tlačítko „Vytvořit úkol“.

Editace úkolu

- Editaci úkolu provádí pouze osoba odpovědná za tento úkol nebo uživatel, který úkol vytvořil.
- Editovat úkol je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se uživateli, který má pravomoci editovat, objeví modré tlačítko „Upravit“. Po zvolení této možnosti se zobrazí formulář s editovatelnými detaily k úkolu. Po dokončení editace je potřeba zvolit tlačítko „Upravit úkol“.

Smazání úkolu

- Smazání úkolu provádí pouze osoba odpovědná za tento úkol nebo uživatel, který úkol vytvořil.
- Smazat úkol je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se uživateli, který má pravomoci editovat, objeví červené tlačítko „Smazat“. Při zvolení této možnosti se zobrazí potvrzovací tabulka pro provedení změny. Potvrzení smazání úkolu se provede opětovným zvolením tlačítka „Smazat“.

Přiřazení oprávnění k úkolu

- Přiřazení oprávnění k úkolu provádí pouze osoba odpovědná za tento úkol nebo uživatel, který úkol vytvořil.
- Přiřadit oprávnění je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se u úkolu objeví modré tlačítko „Více“.
- Po zvolení možnosti „více“ se zobrazí detaily k úkolu. Oprávněná osoba může přiřazovat oprávnění k jednotlivým zaměstnancům nebo přímo oddělením. Po zvolení zaměstnance nebo oddělení ze seznamu se oprávnění dokončí po zvolení tlačítka „Přidat zaměstnance“, nebo „Přidat oddělení“.

Vytvoření komentáře k úkolu

- Vytvořit komentář k úkolu může provést kterákoliv osoba s přístupem k úkolu.
- Přidat komentář je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se u úkolů objeví modré tlačítko „Více“.
- Po zvolení možnosti „Více“ se zobrazí detaily k úkolu. Ve spodní části stránky se zobrazí rolovací část „Přidat oddělení“, která po zobrazení nabízí formulář pro vytvoření nového komentáře. Po vyplnění textu formuláře se vloží komentář zvolením tlačítka „Vložit komentář“.

Vytvoření požadavku

- Vytvoření požadavku k úkolu může provést kterákoliv osoba s přístupem k úkolu.
- Přidat požadavek je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se u úkolu objeví modré tlačítko „Více“.

- Po zvolení možnosti „Více“ se zobrazí detaily k úkolu. V horní části stránky se zobrazí rolovací část „Přidat požadavek“, která po zobrazení nabízí formulář pro vytvoření nového požadavku. Po vyplnění textu formuláře se vloží požadavek zvolením tlačítka „Vložit požadavek“.

Editace požadavku

- Editaci požadavku k úkolu může provést kterákoliv osoba s přístupem k úkolu.
- Editovat požadavek je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se u úkolů objeví modré tlačítko „Více“.
- Po zvolení možnosti „Více“ se zobrazí detaily k úkolu. V tabulce pro požadavky se u jednotlivých požadavků po pravé straně zobrazí tlačítko „Upravit“. Po zvolení tlačítka se zobrazí formulář s aktuálním zněním požadavku s možností editace. Pro potvrzení editace požadavku je nutno zvolit tlačítko „Upravit požadavek“.

Smazání požadavku

- Smazání požadavku k úkolu může provést kterákoliv osoba s přístupem k úkolu.
- Smazat požadavek je možné zvolením sekce „Úkoly“ v horním menu. Po zobrazení aktuálních úkolů se u úkolů objeví modré tlačítko „Více“.
- Po zvolení možnosti „Více“ se zobrazí detaily k úkolu. V tabulce pro požadavky se u jednotlivých požadavků po pravé straně zobrazí tlačítko „Smazat“. Při zvolení možnosti se zobrazí potvrzovací tabulka pro provedení změny. Potvrzení smazání zaměstnance se provede opětovným zvolením tlačítka „Smazat“.

Vytvoření aktuality

- Vytvoření aktuality může provést administrátor nebo zaměstnanec s oprávněním druhého stupně.
- Vytvoření aktuality je možné zvolením sekce „Aktuality“ v záložce na horní liště „Administrace“. Po zobrazení tabulky s aktualitami se objeví v horní sekci rolovací část „Přidat aktualitu“, která po zobrazení nabízí formulář. Pro úspěšné vytvoření aktuality je potřeba vyplnit název a obsah aktuality a zvolit tlačítko „Přidat aktualitu“.

Editace aktuality

- Editaci aktuality může provést administrátor, nebo zaměstnanec s oprávněním druhého stupně.
- Editace je možná zvolením sekce „Aktuality“ v záložce na horní liště „Administrace“. Po zobrazení tabulky s aktualitami se objeví modré tlačítko po pravé straně „Upravit“. Při zvolení možnosti se zobrazí formulář s údaji, které je možné editovat. Potvrzení změn editace se provede zvolením tlačítka „Upravit aktualitu“.

Smazání aktuality

- Smazání aktuality může provést administrátor, nebo zaměstnanec s oprávněním druhého stupně.
- Smazání je možné zvolením sekce „Aktuality“ v záložce na horní liště „Administrace“. Po zobrazení tabulky aktualit se objeví červené tlačítko po pravé straně „Smazat“. Při zvolení možnosti se zobrazí potvrzovací tabulka pro provedení změny. Potvrzení smazání zaměstnance se provede opětovným zvolením tlačítka „Smazat“.

Změna hesla

- Každý uživatel si může změnit své heslo.
- Změna je možná zvolením sekce „Nastavení“ v horním menu. Po zobrazení informací o účtu uživatele se objeví rolovací část „Změna hesla“, která po zobrazení nabízí formulář pro změnu hesla. Pro úspěšnou změnu hesla je potřeba zadat staré heslo a následně 2x nové heslo, které bude delší 6 znaků. Pro potvrzení změny hesla je potřeba zvolit tlačítko „Změnit heslo“.
- Změnu hesla může provést administrátor nebo vedoucí oddělení, do kterého uživatel spadá.
- Možnost změny hesla administrátorem či vedoucím oddělení je při zvolení sekce „Zaměstnanci“ v záložce na horní liště „Administrace“. Po zobrazení tabulky aktuálních zaměstnanců se objeví oranžové tlačítko po pravé straně „Resetovat heslo“. Tento restart nastaví heslo do defaultní podoby, které je potřeba si následně změnit. Správné heslo po restartu je možné zjistit od nadřízeného zaměstnance, nebo administrátora.

Změna emailu

- Každý uživatel si může změnit svůj email.
- Změna je možná zvolením sekce „Nastavení“ v horním menu. Po zobrazení informací o účtu uživatele se objeví modré tlačítko „Změnit email“. Po zvolení možnosti se zobrazí kolonka, do které je možné zadat nový email.
- Změnu emailu může provést administrátor nebo vedoucí oddělení, do kterého uživatel spadá, viz editace zaměstnance.

Odeslání žádosti o dovolenou

- Odeslat žádost o dovolenou může provést kterýkoliv uživatel systému.
- Vytvoření je možné zvolením sekce „Podat žádost“ v záložce na horním menu „Správa dovolené“. Po zobrazení aktuálních žádostí se objeví v horní části rolovací část „Vytvořit žádost“, která po zobrazení nabízí formulář pro vytvoření žádosti. Pro úspěšné vytvoření žádosti je potřeba zvolit termín dovolené, zadat důvod žádosti a zvolit tlačítko „Vytvořit žádost“.
- Seznam a stav svých žádostí uživatel uvidí ve stejné sekci, ve které žádá o dovolenou.

Zjistit výsledek žádosti o dovolenou

- Zjistit výsledek své žádosti o dovolenou může provést kterýkoliv uživatel systému.
- Vytvoření je možné zvolením sekce „Podat žádost“ v záložce horního menu „Správa dovolené“. Při zobrazení aktuálních žádostí se jednotlivě rozliší barevně. Pokud je žádost šedá, znamená to, že ji stále ředitel neschválil. Zelené žádosti znamenají schválení dovolené. Neschválené dovolené jsou zvýrazněny červenou barvou a fungují jako tlačítka, které po zvolení vypíší na horní straně stránky důvod zamítnutí žádosti.

Přehled dovolených

- Zjistit přehled všech schválených žádostí může provést kterýkoliv uživatel systému.
- Náhled schválených dovolených je možné zvolením sekce „Náhled dovolených“ v záložce horního menu „Správa dovolené“. Po zvolení se zobrazí kalendář s dovolenými jednotlivých zaměstnanců. V kalendáři je možné listovat jednotlivými měsíci.

Schválení žádosti o dovolenou

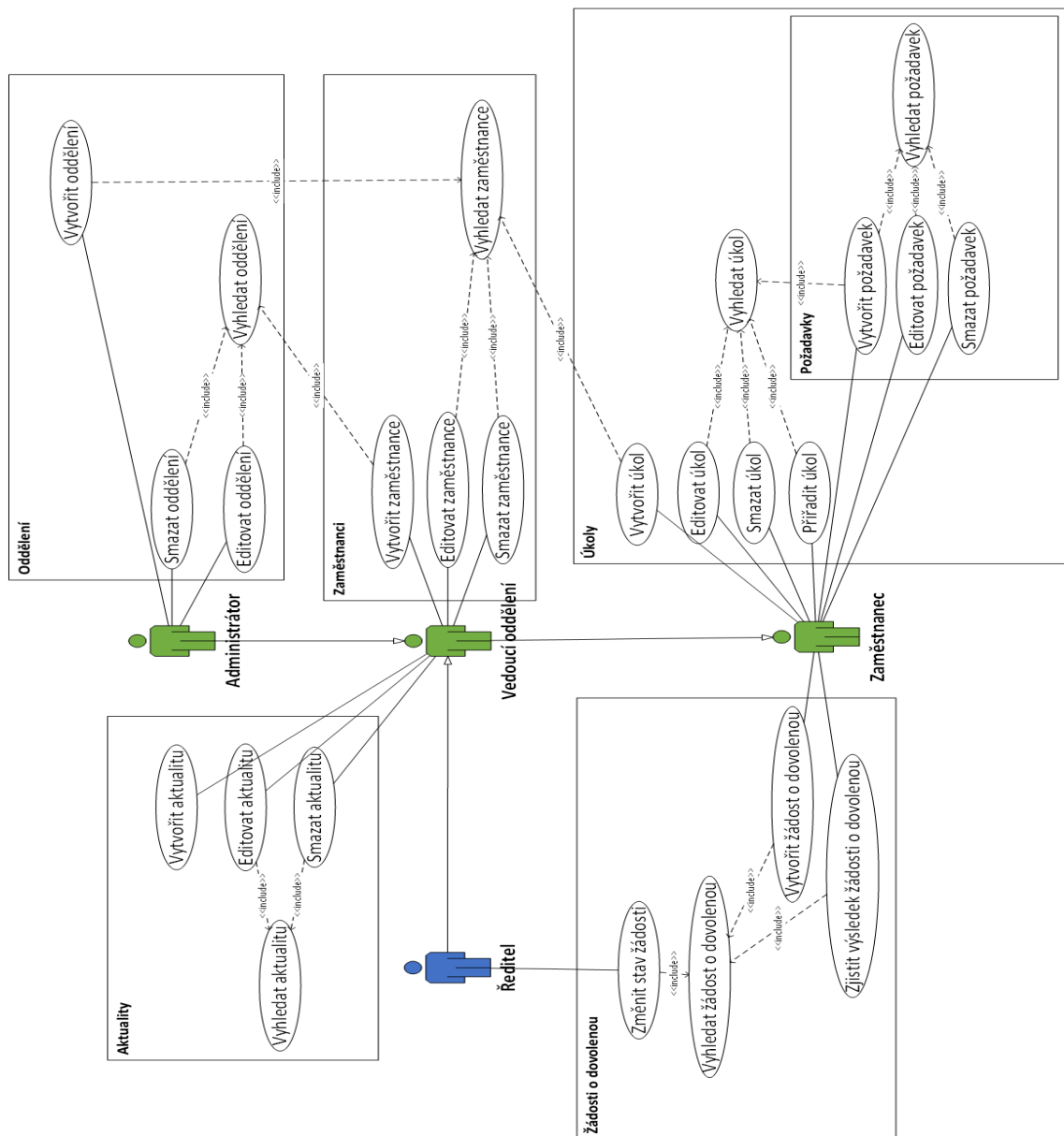
- Schválit žádost o dovolenou může provést ředitel.
- Proces schvalování dovolených je možné zvolením sekce „Administrace dovolených“ v záložce horního menu „Správa dovolené“. Po zvolení se zobrazí tabulka čekajících žádostí a kalendář s dovolenými jednotlivých zaměstnanců. Pro schválení dovolené je nutné u čekající žádosti provést akci „Změnit stav“, kde je potřeba zvolit variantu schválit.
- Schválená žádost je následně v sekci schválených žádostí. Do této sekce je možné se dostat přepnutím přepínače v horní části tabulky „Čekající žádosti“ do stavu „Schválené žádosti“.

Zamítnutí žádosti o dovolenou

- Zamítnout žádost o dovolenou může provést ředitel.
- Proces zamítnutí dovolené je možné zvolením sekce „Administrace dovolených“ v záložce horního menu „Správa dovolené“. Po zvolení se zobrazí tabulka čekajících žádostí a kalendář s dovolenými jednotlivých zaměstnanců. Pro schválení dovolené je potřeba u čekající žádosti provést akci „Změnit stav“, kde je potřeba zvolit variantu zamítnout.
- Zamítnutá žádost je následně v sekci zamítnutých žádostí. Do této sekce je možné se dostat přepnutím přepínače v horní části tabulky „Čekající žádosti“ do stavu „Neschválené žádosti“.

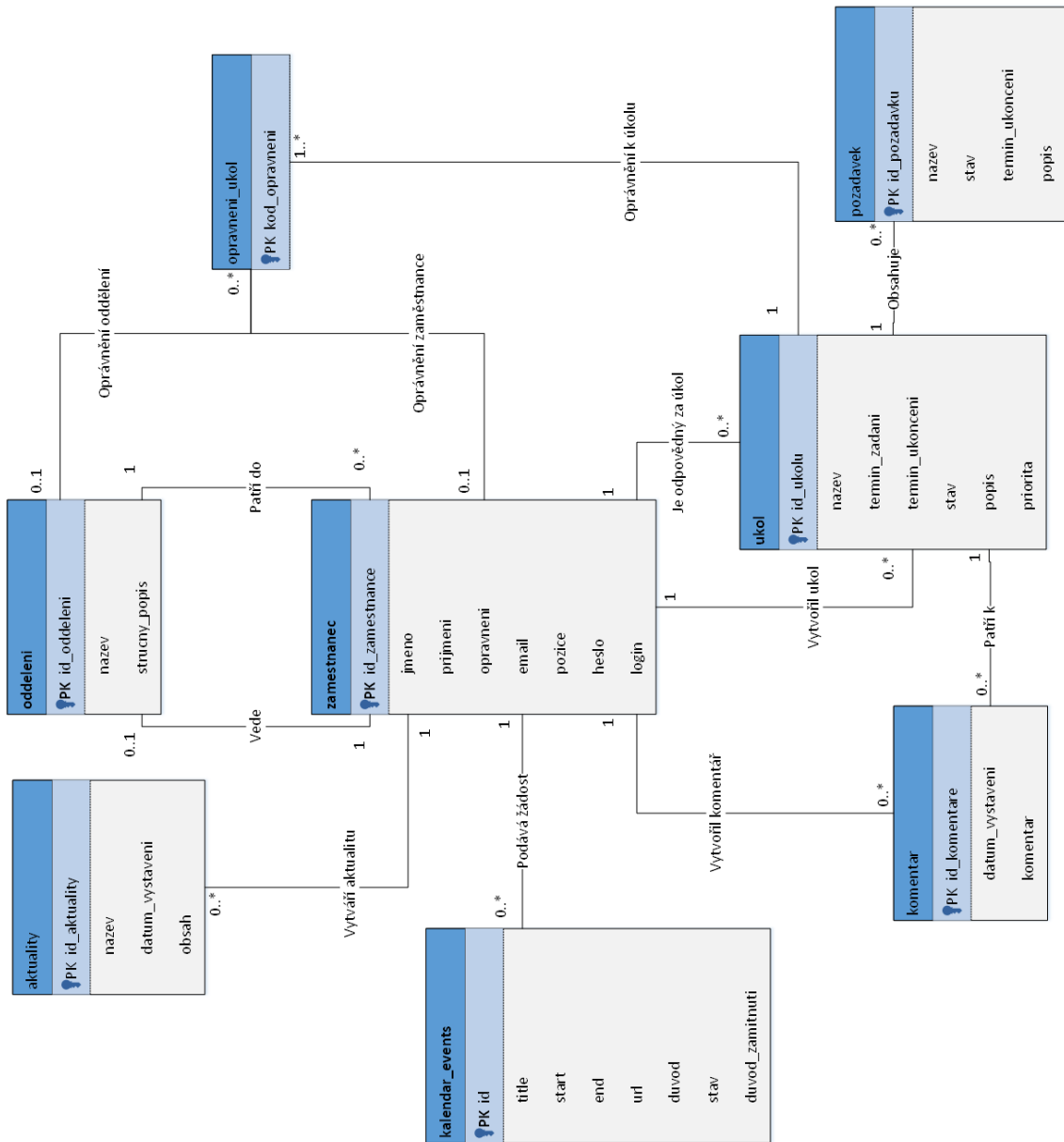
Příloha C

Use Case Diagram



Příloha D

ERD



Příloha E

Zhodnocení vedoucího IT Aqualandu Moravia.

Pan Měřínský se pustil do řešení zadání s patřičnou aktivitou.

Nejprve jsme se sešli a prodiskutovali, jak by měl nový informační systém pracovat a probrali jsme praktické fungování a specifika Aqualandu Moravia a její potřeby s ohledem na tento systém.

Po několikerém osobním setkání a doplnění informací začal vznikat samotný systém, do kterého pan Měřínský postupně přidával všechny funkce, na kterých jsme se dohodli. Funkce mi postupně představoval. Na spolupráci s panem Měřínským si cením zejména jeho nadšenému přístupu k práci. Výsledkem bude ulehčení administrativních úkonů spojených zejména s delegováním a přehledem úkolů všech oddělení. Zároveň jsme domluvení na budoucím přidávání dalších funkcí do systému.

Petr Procházka, Vedoucí IT Aqualandu Moravia.