



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**PODPORA PLÁNOVÁNÍ A ORGANIZOVÁNÍ
FESTIVALU**

SUPPORT FOR PLANNING AND ORGANIZING OF A FESTIVAL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

KAREL POKORNÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ZBYNĚK KŘIVKA, Ph.D.

BRNO 2017

Zadání bakalářské práce

Řešitel: **Pokorný Karel**

Obor: Informační technologie

Téma: **Podpora plánování a organizování festivalu**
Support for Planning and Organizing of a Festival

Kategorie: Informační systémy

Pokyny:

1. Seznamte se s organizačními procesy festivalu Animefest a jejich dosavadní počítačovou podporou, především informačními systémy postavenými nad CMS Drupal a možnostmi jejich rozšíření.
2. Procesy analyzujte a dle konzultací s vedoucím vytvořte specifikaci požadavků na podporu plánování a organizování festivalu (především organizování pořadatelských služeb a správu úkolů vedoucích sekcí a skupin) a navrhnete odpovídající řešení.
3. Dle návrhu tyto (pod)systémy implementujte.
4. Nové (pod)systémy podrobte testování běžnými uživateli, od kterých získáte zpětnou vazbu (např. formou dotazníků), na jejímž základě navrhnete případná vylepšení.

Literatura:

- Butcher, M.: *Drupal: programování modulů*. Computer Press, Brno, 2011, 360 s.
- Polzer, J.: *Drupal 7: podrobný průvodce tvorbou a správou webů*. Computer Press, Brno, 2011, 432 s.

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: **Křivka Zbyněk, Ing., Ph.D.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

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

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Cílem této práce je navrhnout a implementovat modul pro Drupal (systém pro správu obsahu). Návrh bude z velké části založen na již zavedených principech komunity, a proto bude potřeba využívat i data již existujících prvků systému. Modul bude pomáhat organizátorům neziskového festivalu s vytvářením nových směn, udržováním již vzniknutých směn a přehledem nad jednotlivými docházkami dobrovolníků s důrazem na pohodlné použití na mobilních platformách.

Abstract

The goal of this thesis is to design and implement an interface of a module for Drupal (content management system). The design is mostly created according to introduced principles of the community, thus there is a need to use some data from other parts of the system. The module is going to help to organizers of the festival with the shifts creation and management. The user interface is created with emphasis on fine usability on mobile devices.

Klíčová slova

Drupal, PHP, modul, festival, směna, jQuery, MySQL, CSS, JavaScript, responzivita

Keywords

Drupal, PHP, module, festival, shift, jQuery, MySQL, CSS, JavaScript, responsivity

Citace

POKORNÝ, Karel. *Podpora plánování a organizování festivalu*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Zbyněk Křivka, Ph.D.

Podpora plánování a organizování festivalu

Prohlášení

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

.....

Karel Pokorný
14. května 2017

Poděkování

Děkuji panu Ing. Zbyňku Křivkovi, Ph.D za cenné rady a pedagogický dohled při vedení mé bakalářské práce. Dále děkuji panu RNDr. Adamu Rambouskovi, Ph.D za technickou podporu při nesnázích s administrací Drupalu.

Obsah

| | |
|---|-----------|
| 1 Úvod | 3 |
| 1.1 Motivace | 3 |
| 2 Analýza | 4 |
| 2.1 Drupal 7 | 4 |
| 2.1.1 Architektura Drupalu | 5 |
| 2.1.2 Háčky v Drupalu | 6 |
| 2.2 Analýza současného řešení | 6 |
| 2.3 Specifikace klíčových požadavků | 6 |
| 2.3.1 Balíky směn | 7 |
| 2.3.2 Dávkové vytváření směn | 7 |
| 2.3.3 Kód směny | 7 |
| 2.3.4 Logování provedených operací | 7 |
| 2.3.5 Vypnutí možnosti přihlašování | 7 |
| 2.3.6 Responzivní design | 7 |
| 2.3.7 Součást Drupalu | 8 |
| 2.4 Analýza cílových skupin | 8 |
| 2.4.1 Vedení | 8 |
| 2.4.2 Samotní organizátoři | 8 |
| 3 Použité Technologie | 9 |
| 3.1 Drupal a PHP | 9 |
| 3.2 Databáze | 9 |
| 3.3 JavaScript | 10 |
| 3.4 HTML a CSS | 10 |
| 3.5 Responzivní web design | 11 |
| 3.5.1 ViewPort | 11 |
| 3.5.2 Media Queries | 11 |
| 4 Návrh | 12 |
| 4.1 Návrh databáze | 12 |
| 4.2 Uživatelské role | 14 |
| 4.2.1 Role reprezentující postavení | 14 |
| 4.2.2 Skupiny | 14 |
| 4.3 Koncept | 14 |
| 4.3.1 Vytváření směn | 15 |
| 4.3.2 Přihlašování se na směny | 15 |
| 4.3.3 Potvrzování směn | 16 |

| | | |
|----------|---------------------------------------|-----------|
| 4.3.4 | Pohlížení na data | 16 |
| 4.4 | Vzhled | 18 |
| 4.4.1 | Responzivita | 18 |
| 4.5 | Případy užití | 18 |
| 4.5.1 | Entity | 19 |
| 4.5.2 | Akce organizátora | 19 |
| 4.5.3 | Akce vedoucích skupin | 19 |
| 4.5.4 | Akce administrátora | 19 |
| 4.5.5 | Use Case Diagram | 19 |
| 5 | Implementace | 21 |
| 5.1 | Identifikátor směn | 21 |
| 5.2 | Soubory tvořící modul | 22 |
| 5.2.1 | Informační soubor | 22 |
| 5.2.2 | Instalační soubor | 23 |
| 5.2.3 | Soubor s modulem | 23 |
| 5.2.4 | Adresářová struktura modulu | 24 |
| 5.3 | Form API | 26 |
| 5.3.1 | Reakce formulářů | 28 |
| 5.3.2 | hook_form() | 29 |
| 5.3.3 | hook_forms() | 29 |
| 5.4 | Navigační panel | 29 |
| 5.5 | Bezpečnost a oprávnění | 30 |
| 6 | Testování a možná rozšíření | 31 |
| 6.1 | Průběžné testování | 31 |
| 6.2 | UAT - Akceptační testování | 31 |
| 6.3 | Návrhy na rozšíření | 32 |
| 6.3.1 | Zpětná vazba z dotazníků | 33 |
| 7 | Závěr | 34 |
| | Literatura | 35 |
| | Přílohy | 36 |
| | A Obsah CD | 37 |
| | B Dotazník pro organizátory | 38 |
| | C Dotazník pro vedoucí | 40 |

Kapitola 1

Úvod

Cílem této práce je vytvoření webové aplikace, kterou bude možno integrovat formou modulu na weby implementovány redakčním systémem Drupal 7. Modul bude využíván organizátory festivalu Animefest, kteří jsou současně i jeho zadavateli. Aplikace bude rozlišovat dva typy přístupu. Prvním bude přístup pro vedení, kam spadají především vedoucí jednotlivých skupin a administrátoři portálu, druhý přístup už bude pro samotné organizátory, kteří budou vykonávat zadanou práci. Na základě těchto přístupů bude pak aplikace umožňovat uživatelům vytvářet jednotlivé směny nebo celé balíky směn, přihlašovat se na vypsání směny, nebo potvrzovat příchody a odchody na směny.

1.1 Motivace

Motivací je starší nevyhovující aplikace, která je mimo jiné umístěna mimo doménu stránek festivalu a k implementačním detailům neexistuje žádný fyzický přístup. Navíc není implementována Drupalem a funguje na principu přetahování dat z databáze mimo servery Animefestu. Nový koncept zruší narozdíl od současného řešení závislosti na jiných než lokálních nástrojích, větší dynamičnost u vytváření více podobných směn a jejich potvrzování. Další výhodou bude možnost spravovat modul přímo v administraci portálu bez nutnosti přechodu na cizí doménu a skutečnost, že aplikace bude využívat systémová volání jádra Drupalu, čímž se stane jeho plnohodnotnou součástí. Jednota přinese portálu jak vyšší kompatibilitu, tak i vyšší úroveň zabezpečení.

Osobní motivací je pak touha po vytvoření širšího i hlubšího přehledu v oblastech informačních a redakčních systémů.

Kapitola 2

Analýza

Součástí analýzy bude seznámení se s CMS Drupal, specifikace požadavků ze strany zadavatele, hledání co nejhodnějšího řešení na tyto požadavky a zvolení vhodných nástrojů k efektivnímu řešení. Je nutno zjistit především co by měla nová aplikace umět, jaké nároky by měla splňovat, požadavky na implementaci, zjištění cílové skupiny uživatelů a také co zákazníkovi na starém nástroji nevyhovuje.

Prvotní analýza a návrhy aplikace byly mnohokrát měněny, a to především kvůli častým zpětným vazbám po prezentacích prototypů. To mělo sice za příčinu mnoho strávených hodin nad upravováním funkčnosti a vzhledu modulu, ale vzhledem k absenci zkušeností s vývojem webových aplikací a kontaktů se zákazníky tento postup ušetřil hodně práce při závěrečných úpravách a refaktorizaci.

2.1 Drupal 7

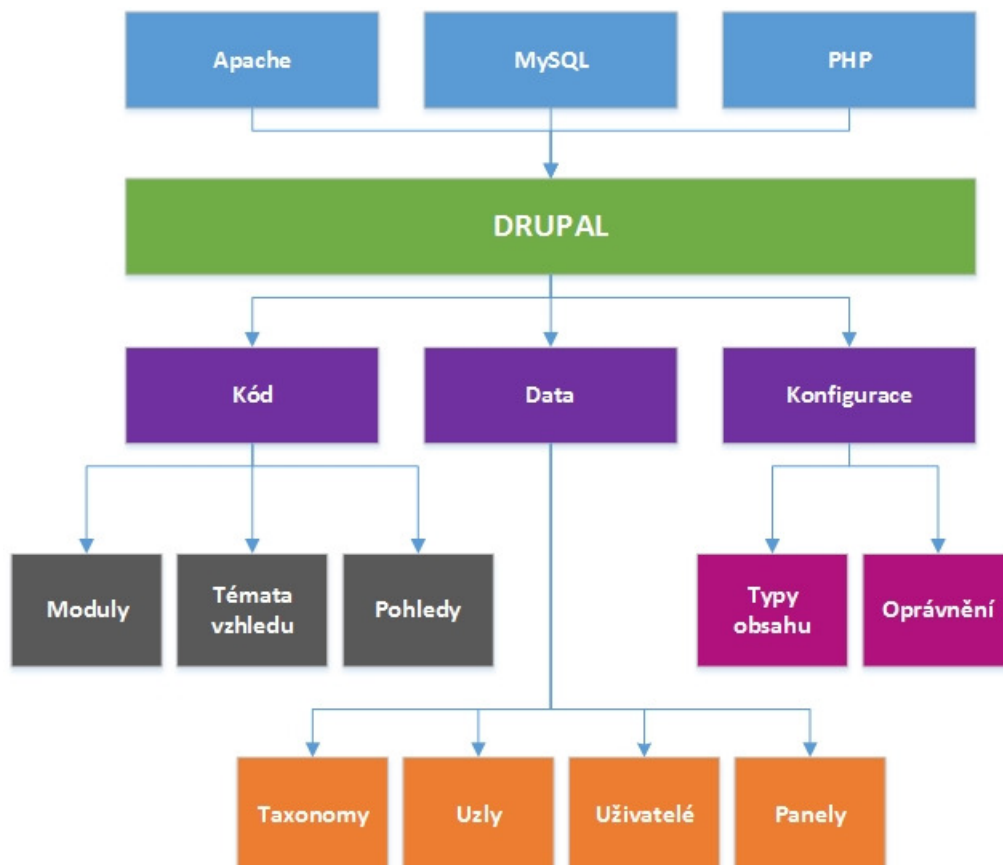
Drupal je typický webový systém pro správu obsahu, který se může navíc pyšnit open source statusem. Jako každý systém pro správu obsahu umožňuje svým návštěvníkům běžné věci, jako prohlížení zveřejněných informací na webu, navigaci na stránkách, vytváření účtů, psaní komentářů. Administrátorům zase nabízí možnost spravovat konfiguraci jejich portálu, úroveň oprávnění nebo přidávat nové obsahy[1].

Tohle všechno jsou sice úžasné funkce, které mnohdy stačí ke spokojenosti, avšak schopnosti, kvůli kterým je Drupal vyjímečný jsou především:

- Možnost rozšiřovat moduly
- Rozšiřování a vytváření témat vzhledu

Drupal je jednoduše systém, který se svojí enormní schopností rozšiřovatelnosti a modularitou dokáže adaptovat na požadavky každého. Ať už chce uživatel používat adresářový server pro autentizaci, exportovat data do souborů, nebo třeba podporu sdílení obsahu na sociální síti, odpověď je vždycky stejná. Existuje na to modul nebo několik modulů propojených k dosažení vysněného cíle.[1]

2.1.1 Architektura Drupalu



Obrázek 2.1: Lze si povšimnout, že se architektura Drupalu skládá ze tří hlavních skupin: kód, data, konfigurace. Každá z těchto skupin pak obstarává kus z celkové reže.

Takže jak to vlastně funguje? Obecně neexistuje úplně přesné pořadí operací, které se provádějí při obdržení požadavku a to především kvůli tzv. háčkovému systému 2.1.2, který dynamicky reaguje způsobem posluchače (který čeká jestli nedojde k určité události a teprve na základě této skutečnosti začne aktivně vykonávat svoji práci). Nicméně pro představu by se zjednodušeně dala reže obsluhy požadavku z pohledu Drupalu popsat následujícími body [1]:

- Zkontroluje jestli je potřeba inicializovat nějaké moduly
- Načte zdrojový kód, který zodpovídá za zpracování dané adresy
- Ověří, že uživatel je oprávněn k provedení vyžádané akce
- Vybídne systém uzlů, k nahrání požadovaného uzlu (podle URL)
- Systém témat vzhledu převezme data uzlu a aplikuje styly
- Stylovaná data se rozhodí do panelů a nabídek
- Vrátí vizuální reprezentaci dat čitelnou pro uživatele

Uzly v Drupalu

Uzly jsou asi to nejdůležitější, co by měl každý uživatel Drupalu pochopit. V Drupalské terminologii je uzel chápán jako kus nějakého publikovatelného obsahu. Typicky obsahuje název, tělo a nějaká pomocná data (údaje o autorovi, čas vložení,..). Uzly jsou jednoduše zodpovědné za obsahy na stránkách.

2.1.2 Háčky v Drupalu

Systém háčků v Drupalu dává modulům šanci zapojit se do životního cyklu požadavku. Jinými slovy, Drupal se snaží při obsluhování požadavků dát šanci modulům ovlivnit výsledek tohoto požadavku. Když bylo dříve zmíněno, že Drupal zjistí, jestli je potřeba inicializovat nějaké moduly, provede to právě touto háčkovací metodou, kdy se dívá, který modul má implementovanou metodu `hook_init()` [1]. Postupně tak volá háček za háčkem a modifikuje výsledná data podle zásahu modulů, které tyto háčky implementují.

Systém háčků je pravděpodobně nejdůležitější částí co se týče programování v Drupalu. Většina modulů nedělá nic víc, než jen implementací skupiny háčků, které Drupal volá, a tím pak ovlivňují finální verzi zaslanou uživateli jako odpověď.

2.2 Analýza současného řešení

Současné řešení docházkového systému má dva velké problémy:

- Nutnost přechodu na jiný portál
- Nemožnost jakkoliv modifikovat současnou aplikaci

V současnosti, pokud chce uživatel vytvořit nějakou směnu nebo se na směnu přihlásit, je nucen opustit stránky Animefestu a znovu se přihlásit na jiném webu (ačkoliv stále stejnými přihlašovacími údaji), kde tyto úkony může provádět. Aplikace funguje tak, že využívá data z databáze Drupalu, která potom používá pro vlastní účely. Tohle řešení je nevhodné především z hlediska bezpečnosti, jelikož se zcela obchází jádro Drupalu a zároveň se data duplikují na straně vedlejšího serveru.

Druhým zmíněným problémem je nemožnost modifikace současné aplikace a stejně tak nemožnost dostat se na server, který tuto aplikaci poskytuje, jelikož administrátoři Animefestu ztratili na tento server přístup. Bez přístupu na server se stává aplikace naprosto nevhodnou, jelikož hrozí např. zrušení hostingu a tím i zaniknutí této aplikace.

Mezi menší – avšak stále důležité – problémy se pak řadí:

- Absence možnosti vytvářet více směn najednou
- Nemožnost exportu/importu směn ze souboru
- Chybí zaznamenávání provedených operací
- Drobné nedostatky v uživatelském rozhraní

2.3 Specifikace klíčových požadavků

Další nezpochybnitelně důležitou součástí návrhu a studia problematiky je seznam věcí, které si zákazník přeje, aby jeho nová aplikace splňovala. Na tyto body musí být v návrhu

brán největší zřetel, jelikož se jedná o funkce, které organizátorům festivalu na starém systému chybí.

2.3.1 Balíky směn

První požadavek říká, že každá vytvořená směna musí mít svoji nadsekcí. Tato nadsekcce může obsahovat nula a více směn a bude nazývána jako balík směn. Všechny směny v balíku mají alespoň jeden společný rys. Typicky se jedná o typ práce, která bude vykonávána, ale není to podmínkou.

2.3.2 Dávkové vytváření směn

Dalším z důležitých požadavků je, aby aplikace poskytovala možnost vytvořit více směn najednou. Uživatel by si měl vybrat, jestli chce vytvořit samostatnou směnu nebo několik podobných směn. Tato možnost se řídí podle zadaného počátečního času, koncového času, a časovým údajem reprezentujícím délku trvání každé směny. Počáteční a koncový čas se tedy potom rozdělí na stejně dlouhé díly a vznikne nám více stejně dlouho trvajících směn, avšak navazujících na sebe.

2.3.3 Kód směny

Každá směna také musí mít svůj unikátní pětimístný kód. Podle tohoto kódu vedoucí skupiny vyhledá směnu při potvrzování, aby nemohlo dojít k omylu a směna tak nebyla potvrzena někomu jinému. Tento kód je vždy složen z pěti písmen, přičemž musí splňovat Hammingovu vzdálenost pro nejméně dvě místa v porovnání se všemi již dříve vygenerovanými kódy.

2.3.4 Logování provedených operací

Aplikace musí navíc zaznamenávat veškeré operace, které byly jejím prostřednictvím vykonány. To znamená, že vytvoření/potvrzení každé směny musí být evidováno, stejně tak i přihlašování a odhlašování.

2.3.5 Vypnutí možnosti přihlašování

Dále si zákazník přeje, aby bylo možno zmrazit přihlašování a odhlašování podle zadaného data a času. Usnadní to komplikace, kdy se lidé odhlašovali den před akcí a docházelo tak k nežádoucím stavům na vybraných směnách.

2.3.6 Responzivní design

Vzhledem k tomu, že má být aplikace používána i na potvrzování příchodů a odchodů na směny, je důležité, aby měla aplikace responzivní design. Tento požadavek je vysoce důležitý, protože na festivalu jsou vedoucí skupin dost pracovně vytížení a nemohou si dovolit tahat s sebou celý den laptop nebo dokonce kvůli každému potvrzení vyhledávat stolní počítač. Naopak mobilní telefon má dnes už téměř každý a potvrzování směn tak nebude zdržovat od práce a zároveň se dá dělat téměř odevšad.

2.3.7 Součást Drupalu

V návaznosti na problémy uvedené v sekci 2.2 je požadováno, aby byla aplikace zakomponovaná přímo v portálu Animefestu prostřednictvím Drupal modulu. Tímto požadavkem si zákazník slibuje odstínění nepříjemných problémů s možnou nekompatibilitou systémů, únikem dat nebo zbytečně eskalovanou režii při přesměrování na jiný web.

2.4 Analýza cílových skupin

Skupiny uživatelů se liší především v tom, jaké části aplikace budou příslušníci těchto skupin využívat.

Cílové skupina uživatelů lze tedy rozdělit do dvou podskupin:

- Vedení
- Samotní organizátoři

Zmiňuji zde pouze tyto dvě skupiny, protože bez vytvoření účtu na portálu není tento web, na kterém aplikace poběží, přístupný širší veřejnosti. Po vytvoření uživatelského účtu a schválení účtu už uvažujeme uživatele za organizátora, který může mít potencionálně zájem přihlašovat se na směny.

2.4.1 Vedení

Jak již název napovídá, do této skupiny se řadí administrátoři portálu a vedoucí skupin. Tato skupina se skládá převážně z technicky zdatných lidí, kteří disponují výbornou internetovou gramotností. Tito lidé budou pracovat primárně s nástroji pro správu a potvrzování směn, tudíž si do těchto nástrojů mohou dovolit zakomponovat abstraktnější prvky uživatelského rozhraní.

2.4.2 Samotní organizátoři

Naopak druhá podskupina se skládá z kombinace lidí, kteří jsou zvyklí na práci s počítačem a lidí, kteří naopak počítač, či mobilní a internetové technologie téměř nepoužívají. Tito uživatelé budou aplikace využívat pouze k prohlížení si jednotlivých směn nebo balíků a přihlašování/odhlašování se ze směn. Právě kvůli hojnému výskytu internetově negramotných uživatelů v této podskupině musí být dbáno na jednoduchost a přehlednost, především pokusit se vyvarovat složitým prvkům, které by mohly uživatele zmást a vést k pochybnostem.

Kapitola 3

Použité Technologie

Tato část textu pojednává o zvolených technologiích a důvodech jejich použití. V této kapitole bude jak úvod do technologií Drupalu, tak i popis technologií, které jsou používány volitelně. Přesto, že se bude aplikace programovat s využitím systémových volání Drupalu, je zapotřebí zakomponovat více technologií pro dosažení optimálního a uživatelsky přívětivého výsledku.

3.1 Drupal a PHP

Tato sekce bude probírat jazyk PHP v kontextu s Drupalem, jelikož bude aplikace psána formou Drupal modulu a používané techniky Drupalu jsou znatelně oříznuté od všech možností nativního jazyka PHP a probírání těchto zákoutí by tak nebylo v rámci tohoto projektu nijak přínosné. Následující text čerpá z [1].

PHP je oblíbený skriptovací programovací jazyk s širokou podporou a početnou uživatelskou základnou. Samotný Drupal je napsaný v programovacím jazyce PHP a jelikož je Drupal psaný v PHP, bude celá logika aplikace vyvíjena právě v programovacím jazyce PHP. Minimální požadovanou verzí jazyka PHP pro Drupal 7 je verze 5.2.

Drupal je zajímavý tím, že ve většině své implementace nepoužívá objektově orientovaný přístup, ale z řady historických a praktických důvodů používá procedurální programování. Nenajdeme zde moc příkladů použití tříd nebo rozhraní, co ale naopak v Drupalu najdeme jsou kolekce funkcí. Neznamena to však, že by Drupal nepoužíval objekty, některé jeho subsystémy jsou objektově orientované.

Pro navrhovanou aplikaci postačí základní konstrukce jazyka PHP s několika standardními funkcemi (např. `levensthein()` nebo funkce pro práci s řetězci) a interními globálními proměnnými Drupalu, které umožňují například zjistit data o aktuálním uživateli.

3.2 Databáze

Drupal ukládá veškeré informace v databázi. Uvnitř této databáze má potom každý typ informace svojí vlastní tabulku[5]. Jako příklad lze uvést tabulku, která sdružuje informace o všech komentářích na portálu (tuto tabulku – `comment` – vytváří spolu s dalšími Drupal automaticky). Nalezneme zde například číslo uzlu ke kterému se komentář vztahuje, nebo kým byl kdy napsán.

Stejně tomu tak bude i pro tento modul, kdy je potřeba ukládat informace o tom kdo se přihlásil na jakou směnu, časové rozpětí směn, které skupiny se mohou přihlašovat na danou

směnu a další. Pro tento účel jsem zavedl do databáze tabulky s prefixem `animefest_`, které obsahují všechna data pro modul.

Podle online dokumentace [5] je nejvíce používanou databází pro Drupal MySQL, ale lze použít i jiné databázové systémy. V našem případě byl využit databázový systém MariaDB.

Zajímavostí je, že Drupal 7 postrádá schopnost používat ve svém databázovém schématu cizí klíče. Je sice možno je nastavit v instalačním souboru modulu, ale tato činnost se nijak neprojeví v modifikování databáze a slouží jen k účelům dokumentace [4].

3.3 JavaScript

Podle knihy [11] je JavaScript multiplatformní objektově orientovaný jazyk, který je v současnosti základem pro vývoj interaktivních webových stránek a webových aplikací. JavaScript se provádí na klientské straně aplikace, což úzce souvisí s potřebou hostitelských programů – typicky webového prohlížeče. Výsledkem zakomponování JavaScriptu jsou například vyjíždějící navigační nabídky, transformace textu, dynamické přidávání elementů nebo dynamické ověřování správnosti zadaných údajů ve formulářích. Je důležité si uvědomit, že JavaScript je komplementární jazyk a tudíž není běžné, aby byla celá aplikace psána pouze v JavaScriptu, nýbrž doplňuje aplikace psané třeba v jazyku HTML (sekce 3.4).

V tomto modulu byl JavaScript použit především pro dynamickou kontrolu u zapisování do formulářů a pro interaktivnější vzhled, jako třeba rozklikávací menu nebo dynamickou změnu textu.

Výše zmíněná dynamická kontrola formulářů je zabezpečena konkrétně knihovnou AJAX (Asynchronous JavaScript and XML), která je schopna měnit obsah stránek bez nutnosti znovunačtení. Knihovna AJAX je napsaná právě v jazyce JavaScript.

Dále je použita javascriptová knihovna jQuery, která se stará o komunikaci mezi HTML a JavaScriptem. Tato knihovna je využita pro interaktivní prvky modulu, jako třeba rozklikávací položky.

3.4 HTML a CSS

V aplikaci byl také využit značkovací jazyk HTML (HyperText Markup Language) a jazyk kaskádových stylů (CSS). Jazyk HTML je základním kamenem pro vytváření webových stránek. Je to neproprietární, čistě textový jazyk, který lze upravovat v jakémkoli textovém editoru. Elementy kódu jazyka HTML se používají za účelem vytvoření struktury stránky, formátování, logického označení obsahu nebo generování interaktivity.

HTML je v modulu použito na uspořádání obsahu aplikace, využita byla především práce s tabulkami pro výpis existujících směn a balíčků, dále pak tagy na zvýraznění důležitých částí textu.

Kaskádové styly se starají o výsledný vzhled stránky. Toho se docílí upravováním elementů napsaných v tomto případě v jazyce HTML. Syntaxe jazyka CSS je založena v první části na definici selektoru, což může být třeba atribut třídy, id, nebo nějaký tag v jazyce HTML. Následuje souhrn deklarácí pravidel, které se mají na daný selektor aplikovat.

V tomto modulu byly kaskádové styly použity k dosažení uživatelsky přívětivého vzhledu aplikace tak, aby korespondoval s již zavedeným tématem vzhledu. Od běžných pravidel se aplikovaly i složitější algoritmy, například na rozdělení barev jednotlivých buněk tabulky ve stylu mozaiky. Každá stránka modulu má vlastní soubor se styly.

3.5 Responzivní web design

V souvislosti s HTML a CSS (sekce 3.4) je zde ještě jeden velice důležitý pojem, který úzce souvisí s vizuálním návrhem modulu. Řeč je o responzivitě.

Responzivní web design dovoluje webové stránce přizpůsobit se různým velikostem viewportů (podsekce 3.5.1) bez toho, aby programátor musel nastavovat speciální doménu/subdoménu pro uživatele s mobilními zařízeními. Vzhled stránky může být takto udržován s podobnými výsledky pro všechny velikosti zařízení. Toho lze docílit použitím meta tagu Viewport a CSS3 Media Queries (podsekce 3.5.2).[6]

Dalším faktorem ovlivňujícím flexibilitu stránky jsou měrné údaje v procentuálních jednotkách. Šířky a výšky tedy nejsou udávány v pixelech, ale v procentech. Tím může určitý element stránky reagovat na různé šířky displejů různých zařízení. Tato procenta se nezaokrouhlují.

3.5.1 ViewPort

Před mobilními zařízeními byly webové stránky navrhovány pouze pro obrazovky počítačů, takže bylo běžné mít statický design a fixní velikost. Viewport je plocha webové stránky, která je viditelná uživatelem na uživatelském zařízení. Hodnota viewportu závisí na velikosti zařízení, pro mobilní telefon bude viewport menší než pro obrazovku počítače.[13]

3.5.2 Media Queries

Media queries je technika představena v CSS3¹. Media queries definují styly pro specifické velikosti viewportů. Pro zahrnutí bloku vlastností, které budou aplikovány pouze, pokud jsou splněny určité podmínky, se používá pravidlo *@media*.

¹CSS3 je poslední standard CSS

Kapitola 4

Návrh

V této kapitole bude popsán základní popis architektury celého systému a jak bude touto prací rozšířen. Dále pak návrh jednotlivých funkčních komponent, které dohromady tvoří funkční modul, ale i návrh dalších neméně důležitých prvků celého systému. Prvky budou dále popsány v takovém pořadí, ve kterém skutečně vznikaly, aby bylo možno spatřit určitou analogii a vztahy mezi jednotlivými návrhy.

4.1 Návrh databáze

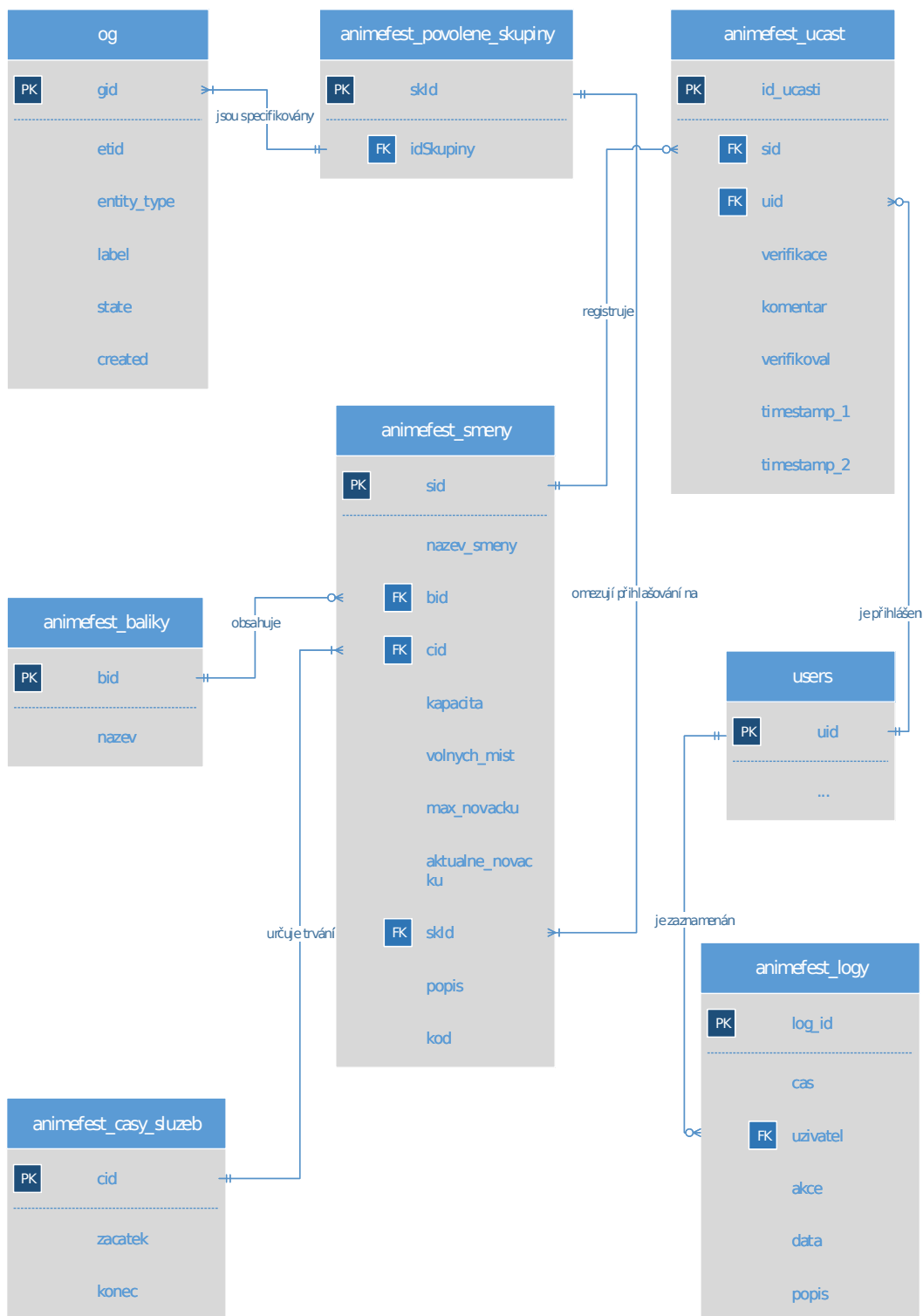
Schéma návrhu databáze je vyobrazeno na obrázku 4.1. Jak už bylo zmíněno v předešlé kapitole 3.2, data modulu jsou uloženy v tabulkách s prefixem `animefest_`. Zároveň se však v pozdější fázi implementace ukázalo jako vhodné použít některé již Drupalem vedené údaje, místo vytváření vlastních pseudo identifikátorů. To především z důvodu, že takto získané údaje jsou daleko mocnějším nástrojem pro přístupuování k vybraným informacím (například o jednotlivých uživateliých). Konkrétně se jedná o tabulky `og` a `users`.

Tabulka `og` obsahuje mimo jiné také identifikátory uživatelských skupin Animefestu a jejich slovní reprezentaci. Tahle informace se velice zužitkovala při implementaci formuláře na vytváření nových směn, kdy je potřeba dát vedoucímu skupiny na výběr ze skupin, kterým k těmto směnám umožní přihlašování viz sekce 4.2.2. Současně i ve všech dalších případech, kdy je potřeba pracovat s uživatelskými skupinami, například pro ověření těchto uživatelů při pokusech o přihlašování na směny.

Tabulka `users` zastřešuje obecné informace o jednotlivých uživateliých portálu. Tabulka opět obsahuje mnoho zajímavých informací, ale pro tento modul je důležitou informací především unikátní identifikátor každého uživatele, přes který se v pozdější implementaci lze přistoupit ke všem potřebným informacím (proto je také schéma ochuzeno o další atributy této tabulky). Jako příklad může posloužit potřeba, kdy chce aplikace zjistit, který uživatel se právě snaží odhlásit ze směny.

Co se týče vlastních tabulek modulu, tabulka `animefest_povolene_skupiny` uchovává informaci o tom, kterým skupinám z tabulky `og` je povolena registrace pro danou směnu. Tabulka `animefest_casy_sluzeb` uchovává informace o tom, ve který den a hodiny se daná směna vykonává. Tento údaj se nemusí vázat k pouze jedné směně, jelikož se připouští, že se může odehrávat více směn ve stejnou dobu, ale třeba na jiném místě. Tabulka `animefest_baliky` obsahuje názvy jednotlivých balíků, do kterých se sdružují směny z tabulky `animefest_smeny`. Tabulka `animefest_ucast` drží a aktualizuje data o orga-

nizátorech přihlášených na danou směnu a jejich docházkách. Nakonec zůstává tabulka `animefest_logy`, která uchovává záznamy o všech provedených operacích v modulu.



Obrázek 4.1: Schéma databáze

4.2 Uživatelské role

Samotný modul sice nevyžaduje definování dodatečných uživatelských rolí, ale je potřeba navrhnout podporu a integraci funkcí modulu pro systém rolí, na kterých je koncept festivalu postaven. Stávající systém uživatelských rolí se dá rozdělit na:

- Role reprezentující postavení
- Skupiny

Drupal samozřejmě nabízí i možnost vytvářet další speciální role podle možností správy obsahu nebo jednotlivých uzlů, tato možnost však nebyla pro návrh modulu důležitá, jelikož se pro takto vznikuté role nemusí vytvářet explicitní podpora.

Mimo výše zmíněné role používá komunita také rozdělení podle kádrovacího posudku. Pro každého uživatele se vede záznam, jestli se jedná o organizátora nováčka, nebo již spolehlivého a samostatného člena týmu. Tato informace již důležitá je, jelikož modul musí podle specifikace mezi těmito skupinami rozlišovat a poskytovat jim rozdílné možnosti.

4.2.1 Role reprezentující postavení

Tyto role zpravidla určují kompetence a odpovědnost uživatelů. Nejnižší úroveň reprezentuje standardní role Drupalu **přihlášený uživatel**, která jednoduše nemá žádné kompetence a slouží pouze k prohlížení povoleného obsahu. Opakem je role **Administrator**, která dovoluje přístup ke všem obsahům i konfiguraci. Další důležité role jsou pak **vedoucí** nebo **vedoucí skupin**, nositel má na starosti všechny organizátory příslušné skupiny. Seznam je zakončený ještě několika dalšími rolemi, které však pro účely projektu není třeba rozepisovat.

Vzhledem ke skutečnosti, že modul je vyvíjen na míru existujícímu systému, musí být brán zřetel i na některá specifická nastavení. Až na výjimky u tohoto typu rolí platí jistá hierarchická dědičnost mezi jednotlivými rolemi. Např. **Administrátor** dědí práva **vedoucích**, **hr**, i obyčejných uživatelů. Toto pravidlo platí obdobně i pro ostatní role.

Zmíněné rozdělení hraje v souvislosti s navrhovaným modulem důležitou roli především v bezpečnosti přístupů, protože podle těchto rolí se rozlišují oprávnění (viz sekce 5.5) jednotlivých uživatelů napříč celým modulem.

4.2.2 Skupiny

Dalším rozdělením jsou skupiny, které reprezentují činnost, kterou v rámci daného kruhu uživatelé vykonávají. Příkladem mohou být skupiny: *Promítači, Řidiči, Hlídky,..*

Tyto skupiny sice nezastávají tak výrazné postavení jako dříve zmíněné Role, ale návrh by se bez nich také neobešel. Pro samotné přihlášení se na směnu musí být přihlašující se uživatel členem některé ze skupin, které jsou pro danou směnu povolené. Neaplikuje se však striktní omezení na skupiny, takže uživatel, který není členem některé z potřebných skupin, si může bez problémů prohlížet všechny ostatní směny včetně cizích účastí. Vytváření nových skupin by podle specifikace nemělo mít negativní dopad na funkčnost modulu.

4.3 Koncept

Tato sekce bude popisovat návrh výsledného řešení a celkové rozložení stránek modulu včetně jejich návazností.

System je navržen tak, aby nevznikaly zbytečné duplicity (jak v datové části, tak i v logickém celku), mimo jiné i proto, že odstranění duplicit byl jeden z původních požadavků. Zároveň je kladen důraz na jednoduché a intuitivní ovládání. V mnoha případech se lze tedy z jednoho konce modulu jednoduše dostat na opačný konec pomocí přiložených odkazů. Tohle zlepšení ocení především vedoucí týmů, jelikož se mohou z prohlížení konkrétní směny pomocí jednoho odkazu dostat na její potvrzování, a tím tak ušetřit minimálně dva přechody mezi stránkami.

Modul jako celek se skládá z třinácti samostatných stránek, které jsou individuálně viditelné a konfigurovatelné na základě interních oprávnění modulu. Tyto stránky je možno opět sdružit podle vztahů, které spolu sdílejí:

- Vytváření směn
- Přihlašování se na směny
- Potvrzování směn
- Pohlížení na data

4.3.1 Vytváření směn

O vytváření směn se stará pouze jedna stránka typu formuláře. Tento formulář se skládá ze tří částí.

V první části se volí informace o směně kterou chce uživatel vytvořit. Povinná pole pro úspěšné vytvoření směny v této části jsou: **Název balíku** (slouží k zařazení výsledné směny, směna se po vytvoření objeví právě ve zvoleném balíku), **Název Směny** (nemusí být unikátní pro balík), **Popis** (krátký popis konkrétní směny, jediná nepovinná položka v této části), **Počet lidí** (celkový počet organizátorů, kteří se mohou na směnu přihlásit) a **Počet nováčků** (určuje, kolik organizátorů z předchozího údaje může mít posudek nováčka). Mimo tato pole je zde i možnost rychlého vytvoření balíku, pro případ, že se vytvářená směna nehodí do žádného již existujícího.

Druhá část formuláře blíže specifikuje datum a čas vytvářené směny. Mimo tyto povinné údaje se zde nachází i možnost generování více směn s totožnou dobou trvání, což byl jeden z hlavních požadavků.

Třetí a poslední část se věnuje specifikaci povolených skupin (viz sekce 4.2.2) výběrem ze všech existujících skupin. Součástí je i možnost vybrat všechny skupiny. Důležitou funkcí této sekce je i možnost přepsat povolené skupiny všem již existujícím směnám v balíku, do kterého je směna vytvářena. Tato funkce se může hodit třeba v případě, kdy je potřeba rychle doplnit stavy směn určitého balíku (typicky tato potřeba nastává v pozdějších fázích přihlašování, kdy jsou již rozebrány všechny lukrativní směny, nebo už jsou organizátoři přihlášení na jiné směny, což může vyústit v nenaplněnou kapacitu). V této fázi přihlašování již jdou preference skupin stranou a je potřeba zpřístupnit možnost přihlašování se na směny i organizátorům, kteří by se bez členství ve skupině na směnu nemohli dostat.

Stejný koncept je plikovaný i pro stránku na úpravu již vytvořených směn, která je vždy dostupná z detailu dané směny.

4.3.2 Přihlašování se na směny

Do této skupiny patří stránky, které jsou nějak spojeny s reprezentací balíků nebo směn a přihlašování na ně.

První takovou stránkou je stránka zobrazující již existující balíky, ať už obsahují nějaké směny, nebo nikoliv. Rozkliknutí balíku pak uživatele přesměruje na výpis směn daného balíku. Je pamatováno i na situaci, kdy je kromě vytvoření balíku potřeba také jeho smazání. V souvislosti s intuitivností zmíněnou v úvodním textu této podkapitoly, je tato funkce řešena přímo zde, v přehledu všech balíků, kde uživatelé s dostatečným oprávněním mohou spatřit miniaturní odkaz na smazání balíku.

Druhou stránkou je pak přímo stránka, na kterou se lze dostat po rozkliknutí některého z balíků. Tato stránka obsahuje všechny směny, které spadají pod daný balík směn. Důraz je kladen na přehledné oddělení jednotlivých směn od sebe, aby nedocházelo k nepříjemnému splynutí a ztráty orientace na stránce. Stránka poskytuje asi nejdůležitější funkci tohoto modulu, a to možnost přihlášení a odhlášení (sebe nebo i jiné uživatele). Tyto a další možnosti jsou opět v zásadě reflektovány oprávněními konkrétního uživatele, ale všichni uživatelé bez ohledu na přidělená práva směji minimálně prohlížet všechny směny včetně jejich detailů a již přihlášené uživatele.

4.3.3 Potvrzování směn

Potvrzování směn je ze všech čtyř částí asi nejobsáhlejší a nejkomplikovanější, jelikož bylo třeba navrhnout spolehlivý postup potvrzování a zároveň jeho jednotlivé kroky postupně provazovat s ostatními částmi modulu pro snadnou orientaci a přecházení (např. z prohlížení směn balíku na kontrolu účastí). Z těchto důvodů se tato část skládá ze tří stránek pro samotné potvrzování směn a jedné stránky, která slouží jako přehledná tabulka všech účastí. Všechny tyto stránky jsou opět přístupné jen s patřičnými oprávněními.

Všechny tři stránky potvrzování směn na sebe navazují. První stránkou ze sledu je samostatný formulář, který obsahuje pouze textové pole a tlačítko. Textové pole očekává pěti místný identifikátor směny (viz sekce 5.1), podle kterého nalezne směnu k potvrzení v dalším kroku.

Po zadání existujícího identifikátoru směny je uživatel přesměrován na druhou stránku, která už obsahuje detailnější popis dané směny. Formulář je zde rozdělen do dvou logických částí. První část obsahuje organizátory, kteří jsou na směnu přihlášení. Mimo to se zde pro každého z těchto organizátorů nachází i stav potvrzení a tlačítko **upravit**, které dává uživateli možnost podívat se detailněji na účast jednotlivce, nebo provést úpravu v potvrzení účasti. V druhé části formuláře se nachází větší textové pole pro případnou poznámku k potvrzení a konečně i samotná tlačítka pro provedení potvrzení. Tlačítka jsou celkem tři (**Příchod**, **Absence**, **Odchod**) a stisk některého z nich potvrdí směnu pro vybrané organizátory.

Poslední stránka potvrzování není nezbytně nutná pro plnohodnotné potvrzení účasti, ale slouží spíše k detailnímu vyobrazení dané účasti a opravy dříve provedeného potvrzení. Jak už bylo zmíněno, na tuto stránku se lze dostat přes tlačítko **upravit**. Po přesměrování se uživatel ocitne na stránce opět rozdělené do dvou logických částí. V první části se nacházejí detailní informace o směně, časové údaje jednotlivých potvrzení a uživateli, který jako poslední provedl potvrzení. V druhé části je již standardně textové pole pro případnou změnu poznámky a tlačítka identická s těmi z předchozího kroku plus nově tlačítko **Uložit změnu**, které uloží právě změnu v poznámce.

4.3.4 Pohlížení na data

Jako první je tu stránka s přehledem účastí, která sice není moc spojená se samotným potvrzováním, neobsahuje ani možnost dodatečně upravit potvrzení, ale má s předchozími

stránkami společnou cílovou skupinu uživatelů. Motivací k návrhu této stránky byla absence možnosti zjistit informace o docházkách organizátorů z více různých směn. Mimo to stránka poskytuje i možnost vyhledání všech směn pro daného organizátora, exportu dat a mnoho dalších užitečných operací pro vedoucí skupin a administrátory portálu.

Stránka je rozdělena do dvou logických částí a tabulky. Tabulka je složená ze šesti sloupců: *Jméno*, *Směna*, *Začátek*, *Konec*, *Příchod*, *Odchod*, *Absence*. Každý řádek pak reprezentuje právě jednu účast. Ve snaze dodržet i zde koncept snadné orientace napříč modulem jsou data v řádcích navíc obohacena o odkazy (např. údaj názvu směny obsahuje odkaz na samotnou směnu v přehledu směn). Kromě očekávaných dat se v levé části každého řádku nachází zaškrtačací políčko.

V počátečním stavu tabulka obsahuje data o všech účastech ve všech směnách. Z toho důvodu stránka nabízí možnost filtrování těchto dat dle libosti uživatele. Tento filtr se nachází v první části stránky a dává na výběr vyfiltrovat účasti s potvrzenou absencí, nepotvrzeným příchodem na směnu, nepotvrzeným odchodem ze směny, a nebo podle jména uživatele.

Posledním významným prvkem této stránky je možnost exportovat vyfiltrovaná data do některého z dostupných formátů. S touto operací se pojí dříve zmíněný pojem zaškrtačacích políček u každého vypsaného řádku. Uživatel si tedy může jednoduše zvolit, zda si přeje exportovat všechna vyfiltrovaná data, nebo pouze některé řádky.

Přidruženou stránkou, která nabízí další informace o směnách a účastech, ale tentokrát je určena spíše pro administrativní účely, je stránka *Sumář směn*. V minulosti nastávaly situace, kdy si organizátoři navolili více směn, než zvládali odpracovat, a vznikaly tak nepříjemné situace ohledně hledání náhradníků na poslední chvíli. Řešením je tato stránka, která nabízí uživateli pohled, kde nalezne informace o všech organizátorech, jež se zúčastní aktuálního festivalu, a to především:

- počet směn, na které jsou přihlášení
- součet časů, které uživatel stráví na směnách
- součet časů potvrzených směn

Modul nabízí také pohled na směny každého uživatele. Organizátor zde nalezne odkaz na směny, informace o jejich začátku i konci a nechybí ani záznam o potvrzení. Uživatel si tak může jednoduše zjistit, kdy a kde se má hlásit, a zároveň mít přehled o svých potvrzených či nepotvrzených účastech. Podobnou funkci poskytuje přehled *Aktuálně probíhající směny*, kde se nacházejí všechny právě probíhající směny. Tento přehled se aplikuje na všechny směny, ne pouze na daného organizátora.

Poslední stránkou pohledů na data je stránka obsahující logy, která do této skupiny bezesporu také patří. Pohled má vzhled tabulky s atributy: *Čas*, *Uživatel*, *Akce*, *Data* a *Popis*. *Čas* signalizuje dobu vyvolání nějaké události, která vede k zalogování. Atribut *Uživatel*, jak už název napovídá, označuje jedince, který stojí za vyvolanou událostí. *Akce* značí několikaslovný identifikátor, který slouží k obecnému značení provedené operace. Ve sloupci *Data* se nachází řetězec formátu CSV¹(comma-separated value), který obsahuje podrobné informace o okolnostech logované události a objektu, který byl operací ovlivněn. Sloupec *Popis* převádí hodnoty z řetězce *Data* do čitelné slovní reprezentace.

¹Řetězec je ve formátu CSV na přání zákazníka, aby měl možnost pracovat s těmito daty ve svém prostředí bez potřeby přístupu do databáze.

4.4 Vzhled

Hlavním cílem při navrhování vizuální stránky modulu bylo, aby bylo prohlížení, spravování a provádění operací modulu co nejpřívětivější jak pro vedoucí, tak i obyčejné uživatele. Důraz byl kladen na drobnosti, jako třeba ani malý, ani velký prostor mezi jednotlivými směnami z důvodu prevence optického splynutí směn.

Bylo potřeba zvolit kombinace příjemných barev, které by korespondovaly s celkovým vzhledem portálu. Naopak pro ikony u potvrzování směn byly zvoleny barvy výraznějšího rázu, aby bylo na první pohled patrné, kdo má, a kdo nemá potvrzenou účast na směně.

Jediná stránka bez výraznějšího stylování je stránka s přehledem účastí. Je to z toho důvodu, že všechny konfigurační stránky v Drupalu, které obsahují tabulku, mají podobný vzhled. Vzhledem k tomu, že možnost prohlížení účastí je uvažována pouze pro administrátory, byla snaha docílit podobného výsledku a zjednodušit tak práci uživateli, aby si nemusel zvykat na nový způsob práce s tabulkami. Navíc je tabulka velice přehledně stylovaná samotným Drupalem, takže nepůsobí strohým dojmem.

Za zmínku stojí barevné rozlišení hlaviček směn, kdy se pro dosažení rychlého přehledu o stavu přihlášených uživatelů tyto mění barvy. Hlavička prázdné směny má šedou barvu. V případě, že je na směnu již nějaký uživatel přihlášen, ale stále ještě zbývá volné místo, je barva změněna na růžovou a plná směna má barvu hlavičky červenou.

Jediný známý nedostatek je v případě našeptávání, kdy současné téma, které je zvolené na portálu překrývá výsledky našeptávání vrstvami, které by měly být zakryty. Nejedná se však o kritické omezení, nýbrž spíš o kosmetickou vadu tohoto jinak fungujícího tématu.

4.4.1 Responzivita

Vzhledem k tomu, že u některých částí modulu se předpokládá, že budou používány především na mobilních zařízeních, bylo potřeba navrhnout, jak má vypadat vizuální stránka aplikace pro zařízení s menšími displeji a rozlišeními. Tento návrh proběhl pro všechny stránky modulu.

Responzivní návrh modulu je založen na viewportech (podsekcce 3.5.1) a media queries (podsekcce 3.5.2). O velikost viewportů se pro tyto účely musí starat tématická šablona portálu. Modul jako takový je totiž pouze část stránky, kterou Drupal vyobrazuje ve svém obsahu. Proto i kdyby modul definoval viewport, nemělo by to očekávaný výsledek, neboť tento údaj musí být definován v samotné hlavičce zdrojového kódu. Naopak media queries již jsou implementovaná pro každou stránku podle potřeby.

Velikosti šířek jsou definovány v procentech, takže se responzivní vzhled dokáže adaptovat na větší množství zařízení s podobnými vizuálními výsledky.

4.5 Případy užití

Tato podkapitola se bude zabírat rozdílem mezi hodnotami a postavením uživatelů ve vztahu k modulu. V následujících odstavcích tedy bude probráno, jaké všechny operace jsou v modulu dostupné, jak spolu souvisejí a pro jaké rozdělení přístupových práv (přístupová práva jsou blíže popsána v sekci 5.5) byl modul navrhnout. Je důležité si uvědomit, že všechny operace, které modul umožňuje provést jsou dosažitelné pouze přes implementovaná oprávnění a tato oprávnění jsou přidělována administrátorem. Následující text pojednává o ideálním případě rozdělení práv, nicméně díky možnostem Drupalu přidělovat práva, jak se kompetentnímu uživateli zachce, nemusí vždy návrh korespondovat s realitou.

4.5.1 Entity

Z diagramu 4.2 lze vyčíst, že návrh uvažuje celkem tři entity - **Organizátor**, **Vedoucí skupiny** a **Administrátor**. Tyto entity na sebe navazují a navzájem od sebe dědí. V praxi to může znamenat, že zatímco obyčejný organizátor se může přihlásit na směnu, jeho vedoucí má kromě této sdílené možnosti také možnost přihlásit i jiné uživatele kromě sebe. Administrátor zastává nejvyšší post a počítá se s tím, že má i možnost rozdělovat oprávnění mezi kompetentní uživatele.

4.5.2 Akce organizátora

Organizátor je uvažován jako nejnižší možná úroveň. Uživatel, který nemá např. možnost zobrazovat balíky, by se totiž do dalších kroků ani nedostal a proto by byl pro model nezájímavý. Organizátoři mají k dispozici jen základní operace jako: Prohlížet balíky, Prohlížet směny, Přihlašovat se nebo Odhlašovat se ze směny a prohlížet si účasti na svých směnách.

4.5.3 Akce vedoucích skupin

Vedoucí skupin už jsou zajímavější entitou. Kromě výše zmíněných základních operací mají vedoucí navíc možnosti vytváření a upravování směn a balíků, jelikož se předpokládá, že to budou právě vedoucí, kteří se budou starat o rozvržení směn uvnitř svých skupin a využívat dostupné kapacity. Mimo to musejí mít také možnost potvrzovat účasti na svých směnách, aby existoval spolehlivý dohled nad docházkou zúčastněných organizátorů i po skončení směny. Vedoucí skupin navíc ještě mají možnost explicitně přihlašovat (nebo odhlašovat) jiné uživatele na své směny.

4.5.4 Akce administrátora

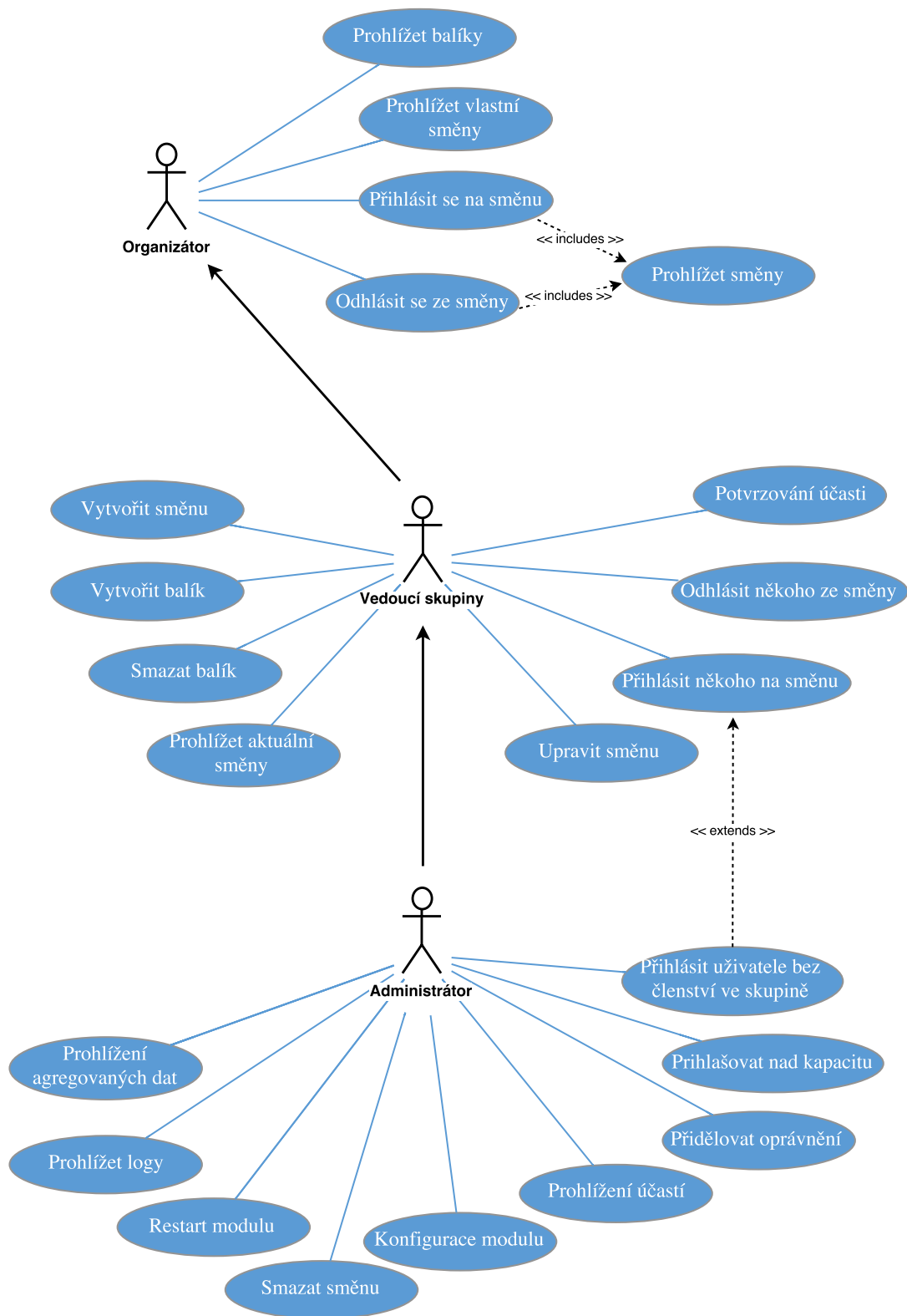
Mezi administrátory se řadí nejspolehlivější a nejzodpovědnější uživatelé. Proto mohou kromě dříve zmíněného provádět velice citlivé operace, které by mohly nenávratně poškodit, či úplně ztratit data modulu. Mezi takové operace se řadí například mazání směn, nebo restart modulu (který nenávratně smaže veškerá data). Zároveň se předpokládá, že nebude potřeba tyto operace provádět často. Mimo to mají administrátoři možnost různě konfigurovat omezení modulu podle svých preferencí a přidělovat oprávnění jednotlivým organizátorům.

Další méně citlivé operace jsou pak možnost přihlásit jakéhokoliv uživatele na jakoukoliv směnu, a to dokonce i když je naplněna kapacita dané směny.

Mezi samostatnou část se pak řadí možnost přehledného prohlížení všech účastí na směnách a možnost exportu těchto dat pro pozdější využití, nebo jako zálohu. Patří sem i možnosti prohlížení agregovaných dat databáze a logů modulu. Tyto operace se připouští i pro vedoucí skupiny, ale vzhledem k přístupu k citlivým datům je v návrhu tato operace uvažována pouze pro administrátory.

4.5.5 Use Case Diagram

Diagram případů užití je nedílnou součástí návrhu každého většího projektu. Obrázek 4.2 popisuje jak typy entit, tak i jejich přidružené akce. Lze si povšimnout hierarchičnosti jednotlivých případů užití, což navazuje především na dříve zmíněné rozdělení oprávnění.



Obrázek 4.2: Diagram případů užití

Kapitola 5

Implementace

Tato kapitola je asi nejvýraznější částí vývoje tohoto modulu. Budou zde popsána programátorská řešení dříve zmíněných problémů podle návrhů z předchozí kapitoly. Prvně budou popsány klíčové části implementace, následně pak jednotlivé komponenty v pořadí, ve kterém vznikaly. Bude zde snaha o vyzdvihnutí zajímavých pasáží a stejně tak i těch problematických, které se během implementace vyskytly a musely být řešeny.

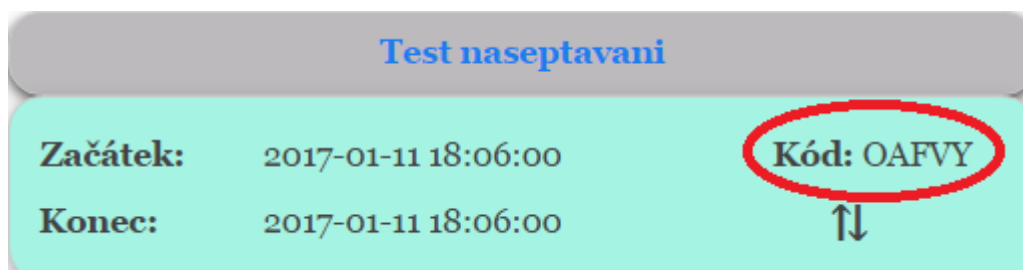
5.1 Identifikátor směn

Prvním problémem, který bylo potřeba vyřešit, byla implementace kódu směn. Tento kód slouží jako unikátní identifikátor směny, který nalezne své využití mimo jiné v případě, kdy existuje více směn se stejným názvem.

Jak už bylo zmíněno ve specifikaci 2.3.3, tento kód se musí skládat z pěti písmen a musí splňovat Hammingovu vzdálenost pro nejméně dvě místa. Modul tedy vytvoří náhodný řetězec z bezdiakritických písmen abecedy. Vytvořený řetězec následně porovnává s každým kódem, který již existuje v databázi. Porovnávání je založeno na algoritmu *levenshtein distance*, za použití funkce standardní knihovny PHP *levenshtein*. Levenshtein distance je algoritmus, který spočítá, jaký nejmenší počet změn v řetězci je nutno provést, aby se z něj stal druhý řetězec [8].

Tento proces je prováděn v nekonečné smyčce, dokud není vygenerován řetězec, který by vyhovoval požadavkům. Úspěšně vytvořený řetězec je potom převeden na velká písmena, aby nedocházelo k nechtěným záměnám některých podobně vypadajících písmen.

Na obrázku 5.1 můžeme spatřit srolovanou verzi směny *Test naseptavani*. Identifikátor směny se zde nachází uprostřed červeného kroužku.



Obrázek 5.1: Pohled na směnu, která se jmenuje Test naseptavani. Vlevo je specifikován začátek a konec směny, vpravo v červeném kroužku se nachází její unikátní identifikátor.

5.2 Soubory tvořící modul

Adresářová struktura Drupalu je velice pestrá. Z pozice kořenového adresáře lze spatřit hned několik podadresářů s povědomými názvy, které zastupují nějakou obecnou funkcionalitu nebo část jádra. Každá složka z tohoto pohledu má svůj význam a silně se nedoporučuje tuto strukturu jakýmkoliv způsobem narušovat. Hlavní adresáře Drupalu a jejich popis podle [12] jsou:

- *includes* - obsahuje knihovny funkcí Drupalu
- *modules* - obsahuje moduly jádra Drupalu
- *scripts* - podpora volání Drupalu z příkazové řádky, kontrola syntaxe, ..
- *themes* - původní témata Drupalu a interpretace šablon
- *sites* - data, která nejsou distribuována jako součást jádra Drupalu
- *profiles* - různé typy instalací portálu
- *misc* - obsahuje JavaScript, ikony a obrázky Drupalu

Ačkoliv by se mohlo intuitivně zdát, že bude náš modul umístěn v adresáři *modules*, není tomu tak. Jak si lze všimnout ve výše uvedeném seznamu, adresář *modules* slouží pouze pro moduly dodávané jako součást jádra Drupalu, tedy uživatel (ani vývojář) by zde neměl nikdy nic vytvářet, ani mazat. Místo toho bude náš modul – stejně jako všechny ostatní stažené moduly či témata vzhledu – umístěn v adresáři *sites*, s absolutní cestou `sites/all/modules/sluzby/`. Z absolutní cesty si lze navíc všimnout, že posledním adresářem cesty je složka `sluzby`, která se jmenuje stejně jako náš modul. Zde platí, že každý modul musí být uložen ve svém vlastním adresáři z důvodů, které budou v zápětí vysvětleny. Jakmile se ocitneme v kořenovém adresáři modulu, jsou zde typicky tři klíčové soubory, bez kterých by modul nemohl fungovat, v horších případech by ani nemusel být viditelný v administraci Drupalu. Řadí se zde informační, instalační a modulové soubory.


5.2.1 Informační soubor

Informační soubory jsou typické svojí příponou *info* (pro náš modul se informační soubor jmenuje `sluzby.info`). Tento soubor slouží k definování informací o modulu, tzn. jeho verze, autora, popisu, závislostí a dalších. [10] Na obrázku 5.2 lze spatřit informační soubor našeho modulu a obrázek 5.3 ukazuje, jak se informace z informačního souboru promítají do administrace.

```
name = AF shifts
description = Plugin pro správu organizátorských služeb Animefestu
core = 7.x
version = 1.1
dependencies[] = date:date_api
dependencies[] = date_popup:date_popup_timepicker
package = Animefest
```

Obrázek 5.2: Informační souboru našeho modulu, který se jmenuje *AF shifts*, patří do vlastního balíku *Animefest*, je dělaný na verzi jádra 7.x a k nainstalování je potřeba mít nainstalované některé moduly balíku *date*.

▼ **Animefest**

| Zapnuto | Jméno | Verze | Popis | Operace |
|-------------------------------------|------------------|-------|---|---|
| <input checked="" type="checkbox"/> | AF shifts | 1.1 | Plugin pro správu organizátorských služeb Animefestu Vyžaduje: Date API (povolen), Date popup timepicker (povolen), Libraries (povolen), System (povolen), Date Popup (povolen), Chaos tools (povolen) |  Oprávnění |

Obrázek 5.3: Informace o našem modulu *AF shifts* v administraci. Nachází se zde jak verze, tak i potřebné závislosti, bez kterých by modul nemohl být nainstalován. Vpravo se nachází rychlý odkaz k nastavení implementovaných oprávnění modulu (viz sekce 5.5).

Pro všechny PHP soubory modulu (včetně souborů modulových a instalačních), kromě těch, které jsou přímo určeny k posílání výstupu, je nepsaným pravidlem, že soubory sice začínají tagem `<?php`, ale nepoužívá se zavírací PHP tag.

Toto opatření předchází běžným problémům způsobeným bílými znaky za zavíracím PHP tagem. Ve formulářích (na kterých je celý modul postavený) by se takový problém promítnul formou chybových či varovných hlášek [9].

5.2.2 Instalační soubor

Pokud modul potřebuje někam ukládat svá data, typicky pro tento účel využívá tabulky v databázi. V tom případě je potřeba databázi na tuto činnost nějakým způsobem připravit a právě k tomu slouží instalační soubory. Tyto soubory obsahují informace o instalaci modulu (typicky se jedná o kód k vytvoření potřebných tabulek v databázi)¹. Kód přípravy tabulek pro modul musí být uveden ve funkci `hook_schema()` v souboru s příponou `install`, kde `hook` je nahrazen jménem modulu [10]. Pro náš modul tedy opět `sluzby.install`.

Instalační soubor zajišťuje ještě jednu úzce související funkci a tou je odinstalování modulu. V administraci Drupalu se nachází volba **Odinstalovat**. Při odinstalování modulu se Drupal opět podívá do instalačního souboru a provede smazání všech tabulek, které byly při instalaci tímto souborem vytvořeny [10].

5.2.3 Soubor s modulem

Modulový soubor lze poznat tak, že vždy končí příponou `module`. Tento soubor obsahuje tělo modulu a definuje jeho funkcionalitu. Mezi hlavní a nejčastěji používané funkce tohoto souboru se řadí vytváření příslušných odkazů napříč celým portálem, implementování přístupových oprávnění pro dosažení víceúrovňového přístupu k datům modulu, správa automaticky prováděných akcí v závislosti na cron², definovat náповědu, atd. [10]. Modulový soubor může obsahovat implementaci funkcionality, nebo se na tyto implementace může dále odkazovat do jiných souborů.

¹Ve starších verzích Drupalu instalační soubory neexistovaly a administrátor byl nucen tyto tabulky vytvářet ručně nebo pomocí SQL skriptů.

²V kontextu operačních systémů se jedná o démona, který se stará o automatizované spouštění nějakého příkazu, či procesu. Známy také jako plánovač.

5.2.4 Adresářová struktura modulu

Modul je z kompozičního hlediska rozdělen do čtyř celků, které se od sebe jazykově nebo typově liší. Podle obrázku 5.4, se v kořenovém adresáři modulu nacházejí tři modulové soubory, které byly popsány v předchozích sekcích a další tři adresáře *css*, *icons*, *source*.

Adresář *css* obsahuje soubory, které se starají o vzhled celého modulu. Každá stránka modulu využívá jeden ze souborů zde uložených a prezentuje tak svůj výstup koncovému uživateli. Soubory jsou psány striktně v jazyce CSS a jejich součástí jsou i implementace pro mobilní zařízení.

V adresáři *icons* se nacházejí stavové ikony. Tyto ikony se používají pro indikaci příchodů P, odchodů O a absencí A jednotlivých organizátorů při operaci potvrzování směny. Ikony příchodu a odchodu jsou zde v šedém (tedy neaktivním) a zeleném (aktivním) provedení. Ikona absence se v tomto liší, jelikož se rozlišuje mezi šedým a červeným provedením. Červená je zvolena kvůli indikaci alarmujícího stavu.

Dalším adresářem je adresář *dependency*, který se skládá ze dvou dalších podadresářů *wvega-timepicker* a *timepicker*. V *dependency* jsou umístěny soubory třetích stran, které podporují funkce modulu. Např. kdyby Drupal neobsahoval ve svém jádře jQuery knihovny, byly by uloženy v tomto adresáři. Jak už bylo dříve uvedeno, jednou ze základních operací, kterou modul organizátorovi poskytuje, je vytvoření směny. Každá směna musí nést časový údaj o svém trvání. Pro uživatelsky příznivější volení časových údajů jsou využity volně dostupné JavaScriptové timepickery, které jsou uloženy právě v těchto podadresářích.

Adresář *source* je posledním dílem kompozice a jednoduše obsahuje implementační soubory pro každou stránku modulu.



Obrázek 5.4: Adresářová struktura a kompoziční rozdělení modulových souborů.

5.3 Form API

Drupal poskytuje API (rozhraní pro programování aplikací) pro vytváření, validaci a zpracování formulářů. Programátorovi abstrahuje tyto formuláře jako vnořená pole s indexy a hodnotami. Toto pole pak drží strukturu a data formuláře a je důležitou částí procesu, když Drupal renderuje stránku obsahující formulář. [12]

Spouštěčem všech událostí souvisejících s operacemi nad formulářem je volání funkce `drupal_get_form()` s argumentem formuláře, který má být vytvořen. Obrázek 5.5 reprezentuje zjednodušený model vytváření formuláře. Lze si povšimnout, že Drupal, pokud je to možné, se snaží ulehčit si hned ze začátku práci a pokud již požadovaný formulář v poslední době pro uživatele renderoval¹, nebude formulář sestavovat znovu, ale vytáhne si jej z cache. Tato schopnost je jistě cenným pomocníkem při šetření strojového času, ale při vývoji je třeba si na toto dát pozor a v případě změny struktury formuláře se vždy doporučuje před testováním nového formuláře smazat obsah Drupal cache.

Pokud formulář není uložen v cache, Drupal se jej pokusí sestavit zřetěžením několika interními (volitelně i externími) voláními. Hned ze začátku celého procesu je nutno si uvědomit, že pro správné fungování formulářů, musí mít každý formulář na aktuální stránce své vlastní `build_id`. Toto `build_id` musí být v rámci stránky unikátní, a to z toho důvodu, že kdyby se na stránce vyskytovalo více formulářů s totožným `build_id`, Drupal by nebyl schopen adresovat požadavky od uživatele ke správnému naslouchači².

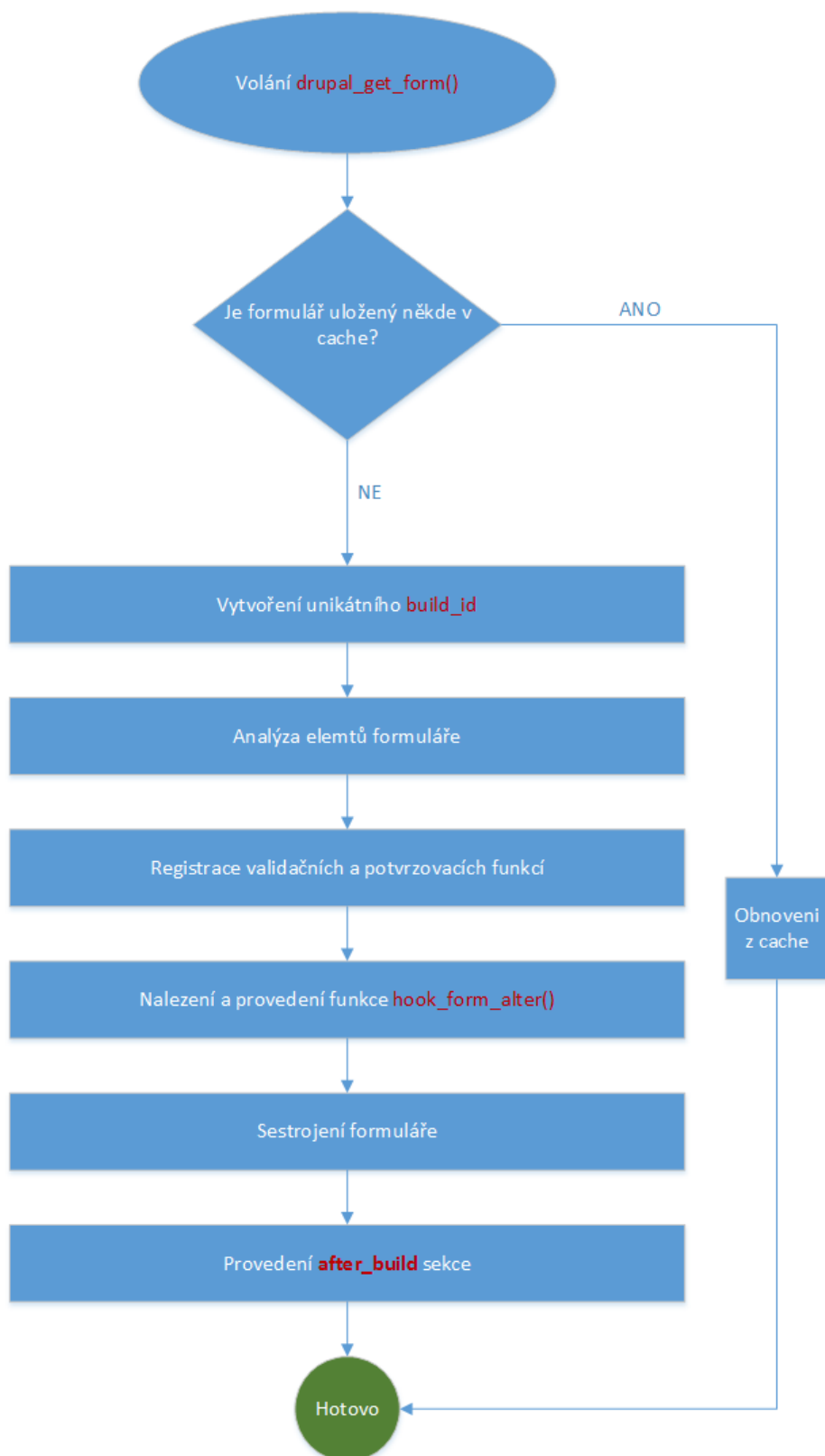
Tento problém může nastat, třeba v případě, kdy si přejeme generovat na stránce proměnný počet totožných formulářů. Při vývoji modulu tento problém nastal při implementaci přehledu směn, kdy se na jednu stránku generují všechny směny z daného balíku, pro každou směnu právě jeden formulář. Drupal, ač poněkud kostrbatě, poskytuje řešení tohoto problému použitím háčku `hook_forms`, který je blíže popsán v podsekcí 5.3.3.

Po vytvoření `build_id` se zanalyzuje struktura formuláře, proběhne prvotní mapování struktury formuláře, což zahrnuje kontroly hodnot, omezení jednotlivých elementů, uspořádání jednotlivých elementů do bloků, atd. Následuje registrace validačních a potvrzovacích funkcí, které budou po vytvoření formuláře potřeba pro uživatelské vstupy. Dalším krokem je úprava některých vlastností formuláře ve funkci `hook_form_alter()`. Tato funkce je důležitá hlavně při interakci uživatele s formulářem, kdy typicky uživatel provede nějakou volbu a formulář mu skryje nebo zobrazí dodatečné elementy.

Mezi poslední úkony pak patří vlastní sestavení formuláře na základě zjištěných indexů a hodnot a nakonec provedení sekce `after_build`, která je volaná potom, co je formulář sestaven k zobrazení. `After_build` se může použít například když chceme zkontrolovat nastavené výchozí hodnoty formuláře.

¹Renderování je v kontextu Form API tvorba a vykreslování daného formuláře na základě dříve nastavených indexů a hodnot.

²Nasloucháčem se rozumí háček, který čeká na své zavolání, aby mohl patřičně zareagovat na uživatelský vstup.



Obrázek 5.5: Pořadí v pozadí prováděných operací při vytváření formulářů.

5.3.1 Reakce formulářů

Reakce formulářů jsou důležitou částí správného pochopení nejen toho, jak Drupal formuláře vytváří, ale i jak s nimi pracuje. Drupal programátorovi nabízí hned několik možností interakce s uživatelem, v tomto modulu byly hojně využity především potvrzovací tlačítka.

V případě, že se uživatel rozhodl pro nějakou akci (třeba přihlásit se na směnu) nebo vyplnil všechny potřebné elementy formuláře a je připravený odeslat požadavek na server, přichází na řadu Drupal. Každé potvrzovací tlačítko má svůj úsek v kódu, kde jsou implementovány požadované operace. Toto místo v kódu není nic jiného, než další háček drupalu – `form_submit`. Drupal však před provedením operace poskytuje programátorovi ještě jeden háček, který se volá těsně před samotným provedením. Tento háček – `form_validate` – slouží typicky ke kontrole vstupních hodnot a zabezpečuje, aby proces v případě neočekávaných hodnot nedošel až k samotnému potvrzování, a tím byl včas zastaven před možným způsobením nekonzistence dat.

V obou těchto háčkách se přistupuje k datům formuláře přes pole `form_state`, mezi jehož nejdůležitější položky se podle dokumentace [3] řadí:

- *values* - obsahuje hodnoty jednotlivých elementů
- *redirect* - určuje co se bude dít po skončení procesu potvrzení
- *rebuild* - určuje, zda bude formulář přestavěn
- *clicked_button* - kopie elementu tlačítka, které bylo stisknuto

Další použitý způsob interakce s uživateli prostřednictvím formulářů je AJAX validátor, který poskytuje přímo Drupalské form API. V zásadě se jedná o rychlou reakci formuláře na základě zadané hodnoty bez nutnosti obnovení nebo znovuvytvoření stránky. Tohoto chování je docíleno právě využitím technologií AJAX, který využívá asynchronního zpracování stránek pomocí knihovny napsané v jazyce JavaScript. AJAX validace je použita u komplexnějších elementů formulářů na vytváření a úpravu směn. Nachází se zde totiž několik elementů, ve kterých by uživatelé mohli dělat často chyby a neustálé obnovování stránky za účelem zpětné vazby od formuláře by se časem mohlo stát velice nepohodlé. Na obrázku 5.6 lze spatřit ukázkou této validace.

2 Čas směny

| | | |
|----------------------------|-------------------------|-----------------------|
| Datum 04-05-2017 | Začátek 14:02 | Konec 13:02 |
|----------------------------|-------------------------|-----------------------|

Generovat po hodinách.

Začátek > Konec

Obrázek 5.6: Příklad použití AJAX validace. Text červené barvy značí uživateli, že zvolil začátek směny později, než je její konec, a že pokud tento vstup neupraví na korektní hodnotu, nebude proces vytváření směny úspěšně dokončen.

5.3.2 hook_form()

Hook_form je základní háček, ve kterém se vytváří jakýkoliv formulář. Tento háček tedy můžeme nazívat funkcí konstruktorem. Na formuláře se dá později přilepit mnoho různých funkcí, ale pokaždé, když je potřeba vytvořit nový formulář, musí být implementován tento háček (jinak by Drupal nedokázal formulář koretně sestavit). Definují se zde všechny výchozí elementy formuláře, včetně všech jejich omezení a vlastností. Tato funkce bere dva parametry:

- `$form` - reprezentuje strukturu formuláře a vlastnosti jeho elementů
- `&$form_state` - reprezentuje aktuální stav formuláře viz popis elementů 5.3.1

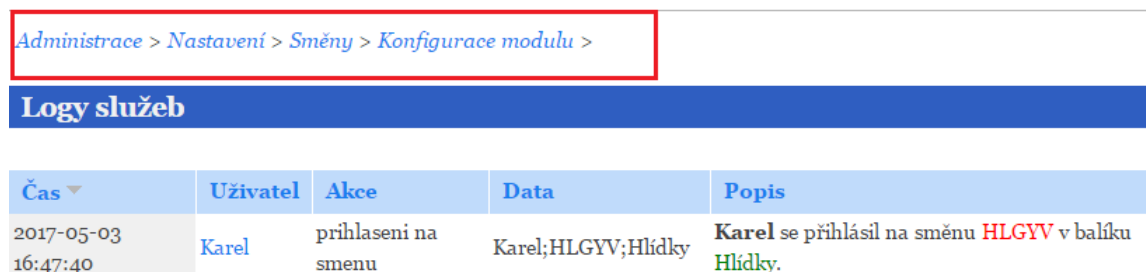
5.3.3 hook_forms()

Za normálních okolností když se zavolá funkce `drupal_get_form()`, Drupal najde funkci se stejným jménem jako je form ID a použije ji k sestavení formuláře. Pokud taková funkce neexistuje, volá se tento háček. Moduly, které implementují tento háček si vytvářejí své vlastní schéma pro mapování form ID do funkcí konstruktorem. Výsledkem je mapování více form ID na stejnou funkci konstruktorem (na kterou se Drupal odkazuje jako na základní formulář). Používá se k vyhnutí se duplicitnímu kódu tím, že se právě používá stejný konstruktorem pro různá form ID [2].

Tato technika hraje důležitou roli v implementaci stránky přehledu směn, kde se právě tento problém vyskytl a musel být řešen. Původně byla snaha řešit tento problém přes pevné nastavení unikátních identifikátorů pro každý formulář na stránce, ale s takto vytvořenými formuláři nebylo možno dále pracovat, jelikož Drupal neměl zaregistrovaná jejich form ID a choval se tak ke všem formulářům tak, jako by na stránce byl pouze jeden.

5.4 Navigační panel

Na obrázku 5.7 lze spatřit provedení navigačního panelu, jež je důležitou součástí modulu, která napomáhá uživateli lépe se orientovat napříč celým modulem. Tento panel se vždy týká kontextu daného zanoření, kde základním bodem je některá položka z nabídky v navigačním bloku. Panel se zobrazuje nad obsahem stránky vždy po vnořeném rozkliknutí (vždy, když vzniká možnost vrátit se o krok zpět).



Obrázek 5.7: Na obrázku se nachází krátký kontext výpisu logů a v levém horním rohu se nachází navigační panel, který je zvýrazněn červeným rámečkem.

5.5 Bezpečnost a oprávnění

Důležitou částí implementace každého projektu je zajištění bezpečnosti. Ať už se jedná o bezpečnost jednotlivých operací, modulu, nebo celého systému, musí být zajištěno, aby určité akce mohly být prováděny pouze ze strany důveryhodných uživatelů. Tuto problematiku Drupal řeší tím, že nabízí programátorovi možnost naimplementovat si své vlastní oprávnění. Tato oprávnění jsou pak aplikovatelná na úrovni uživatelských rolí, které jsou součástí celého systému a s implementací modulu nemají nic společného.

Na obrázku 5.8 je seznam oprávnění, která může administrátor rozdělit mezi uživatele a spravovat tak efektivně modul. Lze si povšimnout, že se na seznamu vyskytují i některá oprávnění, která by se dala považovat za samozřejmá (třeba právo na prohlížení směn). Nicméně je počítáno i se scénářem, kdy někteří uživatelé mohou chtít přistupovat např. jen k obsahu webu, ale nebudou se účastnit nynějšího festivalu jako organizátoři, a proto by neměli mít přístup k žádným datům souvisejícím s modulem, který slouží pouze a jen ke spravování směn.

| Oprávnění |
|---|
| AF shifts |
| Prohlížení dat o účastech |
| Prohlížení agregační tabulky |
| Prohlížení aktuálně běžících směn |
| Prohlížení mých směn |
| Konfigurace modulu |
| Mazání prázdných směn. |
| Mazání směn v konfiguraci. |
| Mazání balíku |
| Mazání balíků v konfiguraci |
| Prohlížení smen |
| Vytváření směn |
| Odhlásit jiné uživatele |
| Přihlásit jiné uživatele |
| Přihlašovat se na smenu |
| Odhlašovat se ze směny |
| Přihlašovat nad kapacitu |
| Přihlašování přes nastavený limit hodin |
| Přihlašovat kohokoliv i mimo směny ve kterých jsem členem |
| Ověřování směn. |

Obrázek 5.8: Seznam všech implementovaných oprávnění modulu.

V závislosti na oprávněních se uživatelům otevírají možnosti přistupovat na speciální stránky pro správu směn (příkladem může být právo *Ověřování směn*, které umožní přistupovat k nástrojům pro potvrzování směn), ale jsou zde i oprávnění, která pouze rozšiřují paletu uživatelských vstupů a možností (např. *Přihlásit jiné uživatele*).

Kapitola 6

Testování a možná rozšíření

Tato kapitola se bude věnovat testování vyvíjeného modulu a návrhům na rozšíření.

Testování je bezesporu velice důležitou součástí vývoje. Jedná se o disciplínu, jejíž náplní je hledání chyb v programu. Ať už se jedná o testování platform, vstupů nebo třeba funkčnosti, na konci testování by měl být program, jehož chování je shodné s jeho specifikací. Na světě existuje mnoho různých metod testování pro projekty různých velikostí. Součástí tohoto textu bude stručné seznámení s tím, jak a kým probíhalo testování, které bylo součástí vývoje tohoto modulu. Vzhledem k tomu, že zadavatelem (tedy i zákazníkem) byl vedoucí této práce, byla zpětná vazba zákazníka k dispozici daleko častěji než u klasického vývoje programu a v poměrně pravidelných časových intervalech. Tato skutečnost mnohdy napomohla k brzkému odhalení nevyhovujících částí modulu či dokonce úpravě specifikace.

6.1 Průběžné testování

Vzhledem k tomu, že modul byl vyvíjen pouze jedním programátorem, bylo mnohem jednodušší průběžně testovat vyvíjenou funkci. Dalo by se říci, že programátor testuje svůj program prakticky pokaždé, když se podívá na interpretaci svého kódu. Po každé úpravě zdrojového kódu a následné aktualizaci stránky se testuje chtěný stav dané komponenty. Tohle testování se doplňuje systematickým (ale i náhodným) proklikáváním stránky a sledováním, zda program odpovídá očekávaným způsobem. Testování se provádí typicky s mezními a neočekávanými hodnotami, jelikož je zde snaha odstínit konečného uživatele od neošetřených chybových stavů a nechtěného chování programu. Výsledkem průběžného testování by měl být program, který se chová podle specifikace a má spolehlivě ošetřeny své vstupy a výstupy. V praxi se toto testování označuje jako *Assembly tests*.

6.2 UAT - Akceptační testování

User acceptance tests (UAT) jsou akceptační testy na straně zákazníka. Tento typ testování je až poslední na řadě a z pravidla se k němu předává aplikace až po té, kdy všechny ostatní testy dopadly úspěšně [7]. Hlavním úkolem akceptačního testování je, aby výsledný program dosahoval původně smlouvené kvality a jeho funkčnost odpovídala představám zákazníka.

Testování je prováděno zákazníkem (nebo jím pověřeným týmem) a typicky se odehrává přímo v jeho prostředí, na jeho strojích. V případě nalezení chyby nebo nedostatku v pro-

gramu se tato událost reportuje zpět k programátorovi a ten by ji měl v co nejkratší době vyřešit.

V případě tohoto modulu probíhalo akceptační testování bez větších komplikací, jelikož byl zákazníkem sestaven akceptační tým, který zároveň představoval skupinu lidí, kteří by s modulem měli v budoucnu pracovat nejčastěji. Uživatelé měli za úkol simulovat ostrý provoz (tzn. vytvářet směny se standardními daty, přihlašovat se na ně, potvrzovat účasti, aj.) a zkoušet různé scénáře, které by programátora, jakožto člověka bez zkušeností v oblastech organizátorství, nenapadly. Zúčastnění měli rozumné připomínky k současnému stavu modulu a navrhli mnohá vylepšení. Všechny známe chyby v programu byly opraveny v dostatečném předstihu před nasazením na ostrý provoz. Většina návrhů na vylepšení byla dodatečně doimplementována jako součást modulu, jelikož se díky nim stal modul jednodušším a flexibilnějším. Mezi doimplementovávané vlastnosti patří informace o stavech směn v přehledu balíků, přehled vlastních směn každého uživatele, aktuálně probíhající směny, nastavení limitu pro možný počet hodin na osobu, export jmenovek a upravování vytvořené směny. Díky průběžným konzultacím se zákazníkem po čas vývoje modulu proběhlo akceptační testování bez větších problémů a modul bude nasazen v květnu tohoto roku.

6.3 Návrhy na rozšíření

Po dokončení fáze testování bylo návrhů na zlepšení poměrně dost (např. změna barev hlavičky směn podle počtu přihlášení, přidání pohledové stránky, kde by každý uživatel viděl své směny a účasti, aj.). Ve většině případů se tyto návrhy obratem implementovaly, jelikož byly rozumné a navrhované funkce by zvýšily použitelnost modulu. Několik nejzajímavějších návrhů je zde rozepsáno.

V současné době si organizátor, jenž chce vytvořit směnu, musí vyplnit a naklikat všechna pole pro vytvoření směny sám. Zlepšením by mohl být nějaký seznam předvyplněných často používaných intervalů a dat vytvářených směn. Tyto intervaly by se nejen předvyplňovaly, ale mohly by se i nabízet automaticky tak, že nabízený interval by byl časově disjunktní se všemi směnami daného balíku. Stejný princip by se dal uplatnit i pro potvrzování účastí na směnách, kdy se ve většině případů předpokládá, že poznámka buďto nebude žádná, nebo pouze opakující se a stručná.

Prioritním návrhem na rozšíření byla pohledová stránka, která by jednoduše ukazovala probíhající směny v reálném čase tak, aby vedoucí skupin, ale i organizátoři, věděli, co se právě děje a kde se případně nachází jejich kolegové. Tato funkce ušetří mnoho režie a zmatků během samotného festivalu, jelikož mají všichni organizátoři ten stejný rozvrh probíhajících směn. Tento návrh se ukázal během testování tak důležitým, že byla pohledová stránka přednostně naimplementována do modulu ještě před začátkem festivalu.

Dalším návrhem na rozšíření byl logovací systém napříč modulem, který by zaznamenával do logu všechny důležité operace. Součástí logu měl být časový údaj vzniku události a její popis. Opět se návrh ukázal jako věc, která by modulu mohla přidat na flexibilitě a zvýšit úroveň zabezpečení, takže i tento návrh byl implementován s časovým předstihem. Log je nakonec obohacen ještě o uživatele a několikoslovný název prováděné akce, za účelem jednodušší práce s logy při aplikaci vyhledávacích filtrů. Dodatečně byl pak přidán i řetězec s daty záznamu pro možné administrativní použití bez nutnosti přístupu do databáze.

6.3.1 Zpětná vazba z dotazníků

Po úplném dokončení implementace (tedy i po fázi rozšiřování) byly všem organizátorům, u kterých je předpokládané aktivní využívání modulu nabídnuto poskytnutí zpětné vazby na organizátorskou část modulu vyplněním dotazníku – viz příloha B. Celkem dotazník vyplnilo 47 respondentů, z nichž 4 vyplňování dotazníku nedokončili. Zaokrouhlené četnosti odpovědí v procentech jsou uvedeny v dotazníku u každé možné odpovědi. Slovní odpovědi byly méně četné a poměrně různorodé.

Na otázku *"Co vám v přehledu balíků směn chybí nebo byste udělali jinak?"* se objevilo několik praktických připomínek ohledně funkcionality (například možnost přesměrování pouze na neplně směny daného balíku), ale i námitek týkajících se estetické stránky pohledu, jako třeba změna písma na bezpatkové nebo zesvětlení zelené barvy.

Dále se objevili připomínky k hlavičce (nerozklikávací části) směny, kde by se podle respondentů mohl navíc objevit přímo údaj o počtu přihlášených organizátorů, indikace omezení pro nováčky, nebo třeba značení, že se jedná o směnu, na kterou jsou již přihlášení. Nápaditým poznatkem byl návrh ke změně číselné formy datumu na slovní zkratku, jenž by reprezentovala den (Pá, So, Ne), jelikož se délka festivalu vleze do jednoho týdne.

Speciální dotazník (viz příloha C) byl pak nabídnut vedoucím skupin a dalším uživatelům zastávajícím vyšší pozice. Na tento dotazník odpovídalo celkem šest respondentů. Stejně jak u dotazníku B jsou zaokrouhlené četnosti odpovědí uvedeny vedle každé z možností. Tento dotazník byl poznamenán tím, že v době vyplňování nebyla ještě možnost plně využít všechny dostupné funkce modulu, protože festival teprve proběhne, takže nebylo možno na některé otázky objektivně odpovědět.

Slovní odpovědi byly narozdíl od předchozího dotazníku častější a obsáhlejší. Na otázku *"Co vám chybí nebo byste změnili na formuláři vytváření směn?"* byla jedna odpověď, která opět zmiňovala upřednostnění výběru data formou dnů trvání festivalu, než číselné reprezentace úplného data. Další reakcí byl návrh na seřazení seznamu povolených skupin podle abecedy, nebo mít možnost definovat, odkdy se může na vytvářenou směnu přihlašovat.

Kapitola 7

Závěr

Cílem této bakalářské práce bylo prostudovat organizační procesy festivalu Animefest a jejich podporou v aktuálním informačním systému. Na základě těchto procesů a potřeb zákazníka sestavit specifikaci požadavků na podporu organizování pořadatelských služeb a navrhnout pro ně odpovídající řešení.

V úvodní části návrhu proběhly konzultace s budoucím uživatelem, kdy sdělil všechny známé nedostatky aktuálního řešení a požadavky, které by měl modul pro dosažení svého účelu splňovat. Na základě těchto požadavků bylo rozhodnuto, že systém bude implementován formou Drupal modulu a byl vytvořen jeho prvotní návrh. Po čas vývoje probíhaly pravidelné konzultace ohledně aktuálního stavu modulu, aby nevznikaly přílišné odchylky realizovaného díla od představ zákazníka. Tento přístup a průběžné testování prototypů (programátorem i zákazníkem) modulu přispělo k pravidelnému sjednocování představ, ale přinášelo i nové požadavky na jeho uživatelské rozhraní a fungování. Především se jednalo o přidávání funkcionalit do již existujících komponent za účelem usnadnění ovládání, ve vyjimečných případech pak vytváření funkcionalit nových.

Po dokončení implementace byl modul vydán zákazníkovi a jeho týmu na provedení akceptačního testování. Cílem akceptačního testování bylo simulovat podmínky skutečného provozu. Výsledky testů prokázali, že modul sice splňuje svoji funkci a je provozuschopný, ale bylo by přínosné, aby byly doimplementovány některé další funkcionality. Následovalo tedy dodatečné implementování navržených komponent a pohledů. Dokončením fáze rozšiřování funkcionalit byl modul předán zákazníkovi, který jej nasadil k ostrému provozu.

Modul byl cíleně vyvíjen především pro vedoucí skupin a jejich specifické potřeby při organizování festivalu. Z informací získaných z dotazníků lze pak vyzorovat, že vedoucí jsou s dosavadním výsledkem většinou spokojeni a shledávají vytvořené nástroje užitečnými, ačkoliv podle vznesených připomínek je zde stále prostor k vylepšování. Nejčastěji se jedná o rozšiřování stávajících vlastností. Co se týče části modulu, která bude převážně využívána obyčejnými organizátory, tak se většinové reakce ukázali také jako pozitivní. Na všechny tyto reakce je však nutno nahlížet s patřičným odstupem, jelikož byl modul doposud použit jen pro část přípravy na festival (vytváření směn a přihlašování) a aktivní použití všech jeho implementovaných funkcí ho terpve čeká.

Literatura

- [1] Butcher, M.; Garfield, L.; Wilkins, J.; aj.: *Drupal Programování modulů*. Computer Press, a.s., 2011, ISBN 978-80-251-3440-5.
- [2] Drupal: *function hook_forms*. [Online; navštíveno 18.04.2017].
URL https://api.drupal.org/api/drupal/modules!system!system.api.php/function/hook_forms/7.x
- [3] Drupal: *Community Documentation - \$form_state keys*. Říjen 2013, [Online; navštíveno 04.05.2017].
URL <https://www.drupal.org/node/1850410>
- [4] Drupal: *Drupal 7 - Schema API*. Říjen 2016, [Online; navštíveno 15.01.2017].
URL <https://www.drupal.org/node/146939>
- [5] Drupal: *Understanding Drupal*. Prosinec 2016, [Online; navštíveno 16.01.2017].
URL <https://www.drupal.org/docs/7/understanding-drupal/general-concepts>
- [6] Firdaus, T.: *Responsive Web Design by Example Beginner's Guide*. Packt Publishing, 2013, ISBN 978-1-84969-542-8.
- [7] Hlava, T.: *Fáze a úrovně provádění testů*. Srpen 2011, [Online; navštíveno 18.04.2017].
URL <http://testovanisoftwaru.cz/tag/akceptacni-testovani/>
- [8] Levenshtein: *The Levenshtein-Algorithm*. [Online; navštíveno 09.04.2017].
URL <http://www.levenshtein.net/>
- [9] Melancon, B.; Micka, A.; Scavarda, A.; aj.: *The Definitive Guide to Drupal 7*. Apress, první vydání, 2011, ISBN 978-1-4302-3136-3.
- [10] Polzer, J.: *Drupal 7 Podrobný průvodce tvorbou a správou webů*. Computer Press, a.s., první vydání, 2008, ISBN 978-80-251-1946-4.
- [11] Suehring, S.: *JavaScript Krok za Krokem*. Computer Press, a.s., 2008, ISBN 978-80-251-2241-9.
- [12] Tomlinson, T.; VanDyk, J. K.: *Pro Drupal 7 Development*. Apress, třetí vydání, 2010, ISBN 978-1-4302-2839-4.
- [13] W3Schools: *Responsive Web Design - The ViewPort*. [Online; navštíveno 04.04.2017].
URL https://www.w3schools.com/css/css_rwd_viewport.asp

Přílohy

Příloha A

Obsah CD

- písemná zpráva ve formátu pdf
- zdrojový kód a soubory pro překlad písemné zprávy
- zdrojové kódy a ostatní soubory modulu
- výsledky dotazníků

Příloha B

Dotazník pro organizátory

Zdá se vám přehled balíků přehledný?

- Ano (95%)
- Ne (5%)

Jak dlouho vám trvalo zjistit, který balík obsahuje směny pro vás?

- Ihned (57%)
- Během chvíle (38%)
- Trvalo mi, než jsem se zorientoval (2%)
- Dlouho (2%)

Jak moc se vám zdají informativní výpisy ("Prázdných směn 0/0;"...) užitečné?

- Jsou potřeba, usnadňují proces přihlašování (64%)
- Zajímavá vlastnost, ale nepoužívám je (34%)
- Zbytečná informace (2%)

Chybí vám v přehledu balíků absence výpisu skupin, pro které jsou balíky určeny?

- Ano (15%)
- Ne (57%)
- Je to nezvyk, ale nevadí mi to (28%)

Co vám v pohledu balíků směn chybí nebo byste udělali jinak?

Zdají se vám změny barev hlaviček podle obsazenosti směny užitečné?

Ano (100%)

Ne (0%)

Obsahuje podle vás sekce pod hlavičkou všechny na první pohled důležité informace?

Ano (91%)

Ne (9%)

Pokud ne, které byste přidali?

Zdá se vám rozklikávací funkce směn intuitivní?

Ano (72%)

Ne (28%)

Obsahuje podle vás rozklikávací sekce všechny na první pohled důležité informace?

Ano (100%)

Ne (0%)

Pokud ne, které byste přidali?

Zdá se vám způsob výběru mezi směny (v případě, kdy jste ve správném balíku a vybíráte si, na kterou z X směn se chcete přihlásit) v balíku intuitivní?

Ano (80%)

Ne (20%)

Příloha C

Dotazník pro vedoucí

Je pro vás formulář na vytváření směn snadno pochopitelný?

Ano (83%)

Ne (17%)

Je pro vás přínosná možnost vytvořit více směn najednou?

Ano (83%)

Ne (17%)

Jak se vám pracuje s pomocnými okny pro výběr data a času?

Dobře, usnadňuje to práci (50%)

Někdy používám, ale nemuselo by to tam být (0%)

Nepoužívám, datum a čas píšu ručně (50%)

Je pro vás způsob vybírání skupin přehledný?

Ano (100%)

Ne (0%)

Co vám chybí nebo byste změnil na formuláři vytváření směn?

Zdají se vám funkce odhlašování a přihlašování jiných organizátorů intuitivní?

Ano (100%)

Ne (0%)

Pokud ne, k čemu máte výhrady?

Usnadňují vám rychlé odkazy ("Upravit směnu", "Potvrdit účast", ..) v přehledu směny práci?

Ano (100%)

Ne (0%)

Jste spokojeni s celkovým konceptem systému potvrzování směn?

Ano (17%)

Ne (0%)

Zatím bahužel nemohu posoudit (83%)

Poskytuje vám možnost výběru mezi orgy dané směny při potvrzování dostatečnou flexibilitu?

Ano (0%)

Ne (0%)

Zatím bahužel nemohu posoudit (100%)

Co byste na systému potvrzování směn změnili/uvítali?

Je pro vás tabulka "Sumář směn" jako celek přínosná?

Ano (17%)

Ne (0%)

Ano s výhradami (33%)

Nevím, nemám práva na tento přehled (50%)

Co vám chybí v sumáři směn?

V tabulce "Přehled účastí", zdá se vám filtr a výpis dostatečně flexibilní?

Ano (33%)

Ne (0%)

Bohužel nemám práva na tento přehled. (33%)

Výpis ano, filtr ne (33%)

Výpis ne, filtr ano (0%)