



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název: ISST - IS Set Tracking
Student: František Koumar
Vedoucí: Ing. Jiří Hunka
Studijní program: Informatika
Studijní obor: Webové a softwarové inženýrství
Katedra: Katedra softwarového inženýrství
Platnost zadání: Do konce zimního semestru 2019/20

Pokyny pro vypracování

Cílem práce je navrhnout a implementovat informační systém pro krajskou sekci Českého svazu orientačních sportů, sloužící k evidenci majetku a výpůjček tohoto majetku.

Navrhněte a realizujte aplikaci, která by usnadnila sledování a správu společného majetku, který je zapůjčován jednotlivým oddílům na jimi pořádané závody, tréninky či soustředění. Aplikace bude sloužit správci ke správě sad dostupných k zapůjčení, udržování informací o jejich stavu (především míra nabití kontrol SportIdent) a přehledu o tom, kde se daná sada právě nachází. Aplikace umožní automatické vystavení účtu na úhradu výpůjčky. Dále bude sloužit i zástupcům oddílů, kteří budou jejím prostřednictvím zaznamenávali předání/převzetí sady a zadávali informace o jejím stavu (ať už ručně či např. nahráním logu o programování kontrol SI).

Aplikaci řádně otestujte.

Navrhněte možná budoucí vylepšení.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

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

V Praze dne 18. února 2018

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

ISST - IS Set tracking

František Koumar

Vedoucí práce: Ing. Jiří Hunka

15. května 2018

Poděkování

Tímto bych rád poděkoval vedoucímu mé práce panu Ing. Jiřímu Hunkovi za trpělivost, ochotu a cenné rady, které napomohly vzniku této práce. Dále bych také rád poděkoval panu Ing. Pavlu Nechanickému, který mě přivedl na myšlenku vytvoření aplikace, jejíž realizací se tato práce zabývá.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. května 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 František Koumar. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Koumar, František. *ISST - IS Set tracking*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato práce se zabývá realizací informačního systému pro oblastní sekci Českého svazu orientačních sportů, který usnadní a zpřehlední evidenci majetku a výpůjček tohoto majetku. V rámci práce bylo implementováno vlastní řešení v podobě pluginu pro CMS WordPress. Vzniklé řešení je primárně určeno pro krajský svaz ČSOS Vysočina, kde proběhne jeho pilotní testování a osvědčí-li se, bude poté šířeno do ostatních oblastí, které o něj projeví zájem.

Klíčová slova evidence vybavení, výpůjčka, webová aplikace, WordPress, plugin

Abstract

Subject of this thesis is realization of information system for regional section of Czech Union of Orienteering Sports, which should make it easier to manage property of this organization and track rentals of that property. Outcome of this thesis is plugin for CMS WordPress, which provides required functionality. As such was this plugin designed primarily to be used in region Vysočina where it will be pilot tested and subsequently it will be provided for use in other regions.

Keywords equipment management, rental, web application, WordPress, plugin

Obsah

Úvod	1
1 Cíl práce	3
2 Vymezení pojmů	5
3 Rešerše existujících řešení	7
3.1 Snipe-IT	7
3.2 Sharing club	7
3.3 Appointment, Reservation and Rental Booking for Woocommerce	8
3.4 Závěr	8
4 Analýza	9
4.1 Uživatelské role	9
4.2 Rozbor požadavků	9
4.3 Případy užití	11
5 Volba platformy a hlavních technologií	21
5.1 Volba platformy	21
5.2 Volba implementačních technologií	23
6 Návrh	27
6.1 Doménový model	27
6.2 Reprezentace entit v systému	27
6.3 Uživatelské rozhraní	29
7 Implementace	33
7.1 Použité technologie	33
7.2 Propojení se systémem ORIS	34
7.3 Uživatelské rozhraní	34

8 Testování	37
9 Další rozvoj aplikace	39
9.1 Filtry a akce	39
9.2 Automatizované přiřazování plateb	39
9.3 Uchování historie atributů	40
9.4 Kalendář událostí	40
Závěr	41
Bibliografie	43
A Seznam použitých zkratek	45
B Obsah příloženého CD	47

Seznam obrázků

6.1	Logické schéma vytvořené v rámci BI-DBS	30
6.2	UML diagram entit	31
6.3	Databázová tabulka – vztah mezi vypůjčenou položkou a výpůjčkou	32
6.4	Databázová tabulka – vztah mezi předmětem a sadou	32
6.5	Databázová tabulka – kluby	32

Seznam tabulek

5.1	Hodnocení jednotlivých možností podoby aplikace	23
-----	---	----

Úvod

Mnoho společností či firem se dnes potýká s problémem, jak jednoduše a přehledně evidovat jimi vlastněné movité statky a jejich pohyb po různých pracovištích či jejich přidělení konkrétním zaměstnancům.

Stejnou situaci je ale možné pozorovat i u mnohých sportovních organizací, které disponují množstvím vybavení potřebným pro výkon daného sportu či zajištění zázemí při pořádání závodů, tréninků a dalších sportovních akcí. Příslušné vybavení také není mnohdy levné a proto je často zapůjčováno členům dané sportovní organizace, kteří tak nejsou nuceni si pořizovat vlastní, nebo je danou organizací zapůjčováno jí podřízeným sportovním oddílům či spolkům v rámci úspory peněžních prostředků. Vzniká tak potřeba evidovat dané pohyby a stav tohoto vybavení.

Jinak tomu není ani v případě krajského svazu ČSOS Vysočina, kde byl za tímto účelem ustanoven správcem vybavení pan Ing. Pavel Nechanický. Byl to právě on, kdo se rozhodl oslovit autora této práce, který se sám již delší dobu podílí na zajištění chodu výše zmíněné organizace, zdali by nebylo možné vytvořit aplikaci, která by celý proces evidence zjednodušila a zřehlednila.

Autor se tak v této práci zabývá návrhem a realizací informačního systému implementujícího požadovanou funkcionalitu, který bude možné nasadit v následující sportovní sezóně v oblasti Vysočina, kde proběhne jeho otestování v ostrém provozu a doladění případných nedostatků ať již po straně funkcionality či uživatelské přívětivosti. Poté bude aplikace zpřístupněna i ostatním krajským svazům, které o ní již zájem projevíly nebo ho v budoucnu projeví.

Cíl práce

Autor této práce si klade za cíl navrhnout a implementovat aplikaci, která usnadní a zpřehlední proces evidence společného vybavení a výpůjček tohoto vybavení v rámci oblasti Vysočina, a kterou bude možné nasadit v následující sportovní sezóně v právě zmíněné oblasti, kde dojde k jejímu otestování v ostrém provozu a doladění případných nalezených nedostatků.

V rámci této práce se pak autor hodlá zabývat také návrhem případných budoucích vylepšení této aplikace.

Vymezení pojmů

V následujícím seznamu jsou k nalezení pojmy, které jsou použity v rámci textu této práce a jejich vysvětlení.

Systém, aplikace – synonyma pro aplikaci, která je výstupem této práce

Kontrola SI, kontrola SPORTident – jedná se o elektronické zařízení sloužící pro záznam průběhu závodníka určitým místem. Závodník k němu za tímto účelem přiloží SPORTident čip, na který je kontrolou zapsán čas průběhu závodníka.

Sady obsahující kontroly SI představují hlavní součást společného vybavení každé oblastní sekce ČSOS.

ORIS – ORIS je páteřní informační systém Českého svazu orientačních sportů, v němž jsou obsaženy údaje o jednotlivých oddílech, registrovaných závodnících a pořádaných sportovních akcích.

Oddíl – v rámci textu této práce je tento pojem použit k označení některého ze sportovních oddílů, které spadají pod Krajský svaz ČSOS Vysočina.

Oblast, oblastní sekce ČSOS, krajský svaz ČSOS – synonyma označující organizace podřízené Českému svazu orientačních sportů, které mají na starosti zajištění organizace dlouhodobých sportovních soutěží v daném kraji

Rešerše existujících řešení

Tento projekt se zabývá typem problému, který je poměrně běžný a to nejen v případě sportovních organizací, ale i dalších spolků, firem a atd. Ve všech případech potřebují tyto subjekty sledovat pohyb jimi vlastněného vybavení, které je přesouváno mezi různými pracovišti či přidělováno/zapůjčováno jednotlivým členům či zaměstnancům. Není tedy divu, že vskutku existují aplikace, které se snaží tuto problematiku řešit.

V rámci této práce budou blíže představeny následující existující řešení.

3.1 Snipe-IT

Snipe-IT [1] představuje open-source aplikaci určenou pro evidenci a sledování hmotného vybavení, softwarových licencí a dalších předmětů, která navíc disponuje poměrně moderním a přehledným uživatelským rozhraním.

Jako takový tento systém umožňuje také správu uživatelů, kterým je poté možno přiřazovat evidované vybavení. Uživatelé mohou dokonce sami požádat o přidělení daného předmětu.

Nevýhodou tohoto systému je jeho instalace, která není zcela triviální a vyžaduje alespoň základní znalost práce s příkazovou řádkou. Toto je problém vzhledem k tomu, že do budoucna je počítáno s rozšířením aplikace také do dalších oblastí. Systém by měl tedy pokud možno být co nejjednodušeji nainstalovatelný.

Také šíření případných aktualizací by v tomto případě bylo problematictější.

3.2 Sharing club

Sharing Club [2] je poměrně jednoduchý plugin určený pro CMS WordPress, který umožňuje vytvoření objektů, které si poté mohou uživatelé vypůjčit.

Celkově plugin umožňuje správci, aby si udržel přehled o pohybu jednotlivých objektů. Téměř veškeré úkony ovšem v tomto případě musí provádět právě správce. Ostatní uživatelé pouze vytváří požadavky na zapůjčení daného předmětu.

Tyto požadavky navíc není možné nijak upřesnit, což by mohlo představovat problém v případě, kdy by o daný předmět požádalo více uživatelů zároveň. Správce by v takovém případě musel nejdříve zjistit, kdy jednotliví uživatelé daný předmět reálně potřebují, aby jim ho mohl zapůjčit v odpovídajícím pořadí.

3.3 Appointment, Reservation and Rental Booking for Woocommerce

Appointment, Reservation and Rental Booking for Woocommerce [3] představuje taktéž plugin pro CMS WordPress, který ovšem v tomto případě rozšiřuje funkcionalitu e-commerce řešení Woocommerce.

Plugin umožňuje vytvoření nového typu produktu, který si uživatel může zapůjčit na jím zvolené časové období. Výpůjčka je tedy časově omezena, což eliminuje problém, který byl popsán u pluginu Sharing Club.

Správce pak může definovat cenu tohoto produktu za jednotku času, což v tomto případě může být hodina, den či týden.

3.4 Závěr

Ačkoliv zmíněné aplikace řeší problém se sledováním sdíleného vybavení, neodpovídá ani jedna z nich specifickým požadavkům problémové domény tohoto projektu.

Všechna tato řešení předpokládají, že zapůjčované předměty jsou jednotlivým uživatelům předány přímo správcem vybavení a po skončení výpůjčky jsou uživatelem správci opět vráceny. V případě problémové domény tohoto projektu ovšem dochází k předávání předmětů přímo mezi jednotlivými uživateli, kteří si sami v závislosti na okolnostech určí způsob a čas předání.

U výpůjček evidovaných v systému je pak zapotřebí uchovávat informaci o tom, na kterou sportovní akci byla realizována. Tímto způsobem je přesně určeno datum, kdy jsou zapůjčené předměty uživatelem reálně využívány.

První dvě zmíněná řešení navíc neumožňují jak vypůjčení daného předmětu za úplaty tak ani zapůjčení více předmětů v rámci jedné výpůjčky.

S ohledem na výše zmíněné skutečnosti se tak autor této práce rozhodl vytvořit vlastní řešení, které by bylo implementováno s ohledem na zmíněná specifika problémové domény.

Analýza

4.1 Uživatelské role

V rámci analýzy byly identifikovány tyto typy uživatelů.

Správce – uživatel představující správce vybavení. V rámci systému je mu umožněno provádět veškeré administrátorské úkony.

Běžný uživatel – uživatel, jemuž je umožněno pouze realizovat výpůjčky a zadávat informace o stavu jednotlivých evidovaných předmětů

4.2 Rozbor požadavků

Specifikace požadavků nebo také SRS (z anglického Software Requirements Specification) představuje první reprezentaci schopností a vlastností dodávaného software. Jako taková představuje základ pro veškeré další úkony na softwarovém projektu, což z ní ovšem činí silný zdroj potenciálních rizik, které mohou negativně ovlivnit průběh vývoje a podobu výsledného software. Vzhledem k této kritické důležitosti specifikace požadavků je nutné věnovat jí zvýšenou pozornost, protože sebemenší chyba v ní obsažená může představovat potenciální důvod selhání softwarového projektu. [4]

V rámci specifikace je rozlišováno mezi dvěma typy požadavků:

Funkční požadavky definují, jakou funkcionalitu má systém implementovat.

Nefunkční požadavky představují doplněk k požadavkům funkčním a popisují omezení kladená na systém jako celek. Jedná se například o požadavky na výkon, dostupnost či bezpečnost systému. [5]

4.2.1 Funkční požadavky

- **Správa uživatelů:** Systém umožní správci administraci uživatelských účtů a veškerých souvisejících údajů o uživateli.
- **Kontaktní informace:** Vypůjčené vybavení často není vráceno přímo správci, ale je místo toho aktuálním držitelem po domluvě předáno zástupci oddílu, který pořádá následující akci a u níž indikoval, že bude dané vybavení potřebovat. Z tohoto důvodu musí systém umět poskytovat kontaktní informace jednotlivých uživatelů.
- **Správa oddílů:** Běžní uživatelé uskutečňují výpůjčky pro oddíl, jehož jsou členy. Systém tedy umožní načtení oddílů ze systému ORIS a jejich přiřazení danému uživateli.
- **Správa jednotlivých předmětů a sad:** Systém bude umožňovat do systému přidat a spravovat předměty. Dále umožní definovat sady, do kterých bude možné tyto předměty přiřadit.
- **Správa ceny za vypůjčení:** Systém musí správci umožnit zvolit způsob nacenění výpůjčky jednotlivých předmětů/sad (cena za výpůjčku, cena za závodníka s nastavenou maximální výší) a nastavit případné atributy zvoleného způsobu.
- **Stav kontrol SI:** Systém umožní sledování aktuálního stavu kontrol SI a bude schopen upozornit správce na nízkou hladinu nabití kontrol SPORTident.
- **Vystavení účtu na úhradu:** Systém bude umožňovat automatické vystavení účtu na úhradu dané výpůjčky.
- **Získání informací o sportovních akcích:** Systém bude schopen načítat potřebná data o sportovních akcích z informačního systému ORIS.
- **Obrazovka typu dashboard:** Systém bude schopný prezentovat nej důležitější data na obrazovce typu dashboard.

4.2.2 Nefunkční požadavky

- **Snadné nasazení a šíření aktualizací:** Vzhledem k tomu, že vytvořená aplikace s největší pravděpodobností nebude používána pouze v oblasti Vysočina, ale i v dalších oblastech, mělo by být možné pro správce v daných oblastech aplikaci snadno zprovoznit. Ze stejného důvodu by také bylo vhodné zajistit bezproblémové šíření případných aktualizací.

4.3 Případy užití

Use cases neboli česky případy užití představují techniku zachycení funkčních požadavků z pohledu uživatele software prostřednictvím strukturovaných scénářů. [6]

V případě této práce byly při sepisování případů užití využity poznatky získané z [7].

4.3.1 Seznam případu užití

Případy užití společné pro oba typy uživatelů

- **U1.1** Přihlášení
- **U1.2** Změna vlastního hesla
- **U1.3** Obnovení zapomenutého hesla
- **U1.4** Vytvoření nové výpůjčky
- **U1.5** Přidání předmětu k výpůjčce
- **U1.6** Převzetí předmětu v rámci výpůjčky
- **U1.7** Nahrání informací o stavu kontrol SI v rámci výpůjčky

Případy užití pro správce

- **U2.1** Vytvoření nového uživatele
- **U2.2** Úprava kontaktních údajů uživatele
- **U2.3** Vygenerování nového hesla uživatele
- **U2.4** Načtení oddílů ze systému ORIS
- **U2.5** Změna oddílu uživatele
- **U2.6** Vytvoření nového předmětu
- **U2.7** Vytvoření nové sady
- **U2.8** Přidání předmětu do sady
- **U2.9** Přidání kontrol SI do sady pomocí logu o jejich programování
- **U2.10** Změna ceny zapůjčení daného předmětu či sady
- **U2.11** Změna hranice pro hlášení vybité baterie kontroly SI
- **U2.12** Vygenerování faktury

- **U2.13** Záznam zaplacení výpůjčky
- **U2.14** Načtení informací o sportovních akcích ze systému ORIS
- **U2.15** Nahrání informací o stavu kontrol SI

4.3.2 Případy užití společné pro oba typy uživatelů

U1.1 Přihlášení

Předpoklady: Uživateli byl správcem vytvořen účet a předány přihlašovací údaje.

Cíl: Zamezit neautorizovanému přístupu do systému.

Hlavní scénář:

1. Uživatel přistoupí na přihlašovací obrazovku aplikace
2. Uživatel vyplní požadované informace tj. jemu přidělené uživatelské jméno a heslo.
3. Jestliže jsou údaje správné, je uživateli umožněn vstup do systému.

U1.2 Změna vlastního hesla

Předpoklady: Uživatel je přihlášen do systému.

Cíl: Uživateli bude umožněn vstup do systému s použitím nově zvoleného hesla.

Hlavní scénář:

1. Uživatel přistoupí na obrazovku editace svého účtu
2. Uživatel vyplní nové heslo a potvrdí jeho uložení.
3. Systém uloží nové heslo.

U1.3 Obnovení zapomenutého hesla

Předpoklady: Uživateli byl správcem v systému vytvořen účet.

Cíl: Uživatel bude opět schopen se přihlásit do systému.

Hlavní scénář:

1. Uživatel přistoupí na přihlašovací obrazovku aplikace
2. Uživatel zvolí možnost obnovení hesla
3. Uživatel je systémem přesměrován na obrazovku pro obnovení hesla
4. Uživatel zadá své uživatelské jméno nebo emailovou adresu

5. Je-li zadaný údaj možné spojit s existujícím uživatelským účtem, pošle systém příslušnému uživateli na email vygenerované nové heslo

U1.4 Vytvoření nové výpůjčky

Předpoklady: Uživatel je přihlášen do systému.

Cíl: V rámci systému bude evidována nová výpůjčka.

Hlavní scénář:

1. Uživatel přejde na obrazovku s výpisem výpůjček.
2. Uživatel zvolí možnost vytvoření nové výpůjčky.
3. Systém ho přesměruje na obrazovku pro vytvoření nové výpůjčky.
4. Uživatel zvolí ze seznamu událost, k níž se má výpůjčka vztahovat a potvrdí její vytvoření.
5. Systém vytvoří příslušný záznam.

Poznámky:

1. Správci bude navíc umožněno zvolit uživatele, který výpůjčku realizuje.
2. Uživatel (s výjimkou správce) je ve výběru událostí omezen na ty, které pořádá jeho oddíl.
3. Výpůjčku na danou událost může běžný uživatel vytvořit nejpozději x dní před datem konání vlastní události, kde x je hodnota definovaná správcem v nastavení aplikace.

U1.5 Přidání předmětu k výpůjčce

Předpoklady:

1. Uživatel je přihlášen do systému.
2. Daná výpůjčka není uzavřená.

Cíl: K Dané výpůjčce bude přiřazen další předmět.

Hlavní scénář:

1. Uživatel přejde na obrazovku pro editaci výpůjčky.
2. Uživatel vybere daný předmět a zvolí možnost přidání předmětu.
3. Systém přidá předmět k výpůjčce ve stavu „rezervováno“.

Poznámky:

4. ANALÝZA

1. Běžný uživatel může přidat předmět k výpůjčce do nejzazšího data, do kterého je ještě možné vytvořit novou výpůjčku (viz U1.4).
2. Správce může předmět přidat i k výpůjčce, jejíž událost již proběhla. V takovém případě je předmět přidán k výpůjčce ve stavu „předáno“.

U1.6 Převzetí předmětu

Předpoklady:

1. Uživatel je přihlášen do aplikace.
2. Daná výpůjčka se vztahuje k události, u níž již došlo k přesažení nejzazšího data pro vytvoření výpůjčky (viz U1.4).

Cíl: Stav předmětu v rámci příslušných výpůjček budou aktualizovány.

Hlavní scénář:

1. Uživatel přejde na obrazovku pro editaci dané výpůjčky.
2. Uživatel zvolí možnost převzít daný předmět.
3. Systém změní stav předmětu v rámci dané výpůjčky na „vypůjčeno“ a v rámci výpůjčky, jejíž součástí byl předtím, na „předáno“.

Poznámka: Správci bude navíc umožněno převzít libovolný zapůjčený předmět prostřednictvím obrazovky dashboard.

U1.7 Nahrání informací o stavu kontrol SI v rámci výpůjčky

Předpoklady:

1. Uživatel je přihlášen do aplikace.
2. Daná výpůjčka se vztahuje k události, která již proběhla.

Cíl: V systému budou obsažena aktuální hodnoty atributů kontrol SI.

Hlavní scénář:

1. Uživatel přejde na obrazovku pro editaci dané výpůjčky.
2. Uživatel zvolí příslušný log soubor z programu SPORTident Config+.
3. Uživatel potvrdí, že chce informace z poskytnutého souboru načíst.
4. Systém zpracuje soubor a aktualizuje příslušné hodnoty. Dále u výpůjčky zaznamená, že v rámci ní byl log soubor nahrán.

4.3.3 Případy užití pro správce

U2.1 Vytvoření nového uživatele

Předpoklady: Správce je přihlášen do aplikace.

Cíl: V systému bude vytvořen nový uživatelský účet.

Hlavní scénář:

1. Správce zvolí možnost vytvořit nový uživatelský účet
2. Správce zadá všechny požadované informace:
 - uživatelské jméno,
 - emailovou adresu uživatele,
 - příslušný oddíl uživatele.
3. Jestliže jsou údaje správné, vytvoří systém nového uživatele a na uvedenou emailovou adresu zašle přístupové údaje.

U2.2 Úprava kontaktních údajů uživatele

Předpoklady: Správce je přihlášen do aplikace.

Cíl: V systému budou uvedeny aktuální kontaktní informace daného uživatele.

Hlavní scénář:

1. Správce zvolí ze seznamu uživatelských účtů příslušný účet.
2. Systémem je přesměrován na obrazovku pro editaci daného účtu.
3. Správce změní potřebné údaje o uživateli a potvrdí jejich změnu.

U2.3 Vygenerování nového hesla uživatele

Předpoklady: Správce je přihlášen do aplikace.

Cíl: Existujícímu uživateli bude vygenerováno nové heslo.

Hlavní scénář:

1. Správce přejde na obrazovku pro editaci daného účtu.
2. Správce zvolí možnost vygenerovat nové heslo.
3. Systém vygeneruje nové heslo.
4. Správce potvrdí změnu hesla.

U2.4 Načtení oddílů ze systému ORIS

Předpoklady:

1. Správce je přihlášen do aplikace.
2. Správce v nastavení aplikace vyplnil název oblasti.

Cíl: V systému budou dostupné oddíly dané oblasti.

Hlavní scénář:

1. Správce přejde na obrazovku s výčtem oddílů.
2. Správce zvolí možnost načtení oddílů ze systému ORIS.
3. Systém načte potřebné informace o oddílech a zaznamená je.

U2.4 Změna oddílu uživatele

Předpoklady: Správce je přihlášen do aplikace.

Cíl: U daného uživatele bude aktualizována příslušnost k oddílu.

Hlavní scénář:

1. Správce přejde na obrazovku pro editaci daného účtu.
2. Správce zvolí ze seznamu nový oddíl uživatele a potvrdí provedenou změnu.

U2.6 Vytvoření nového předmětu

Předpoklady: Správce je přihlášen do aplikace.

Cíl: Do systému bude vložen nový samostatný předmět.

Hlavní scénář:

1. Správce zvolí možnost vytvoření nového předmětu.
2. Systémem je přesměrován na obrazovku pro vytvoření nového předmětu.
3. Správce vyplní požadované údaje (může se lišit v závislosti na typu předmětu).
4. Správce potvrdí vytvoření předmětu.

U2.7 Vytvoření nové sady

Předpoklady: Správce je přihlášen do aplikace.

Cíl: Do systému bude vložena nová sada.

Hlavní scénář: Průběh je stejný jako v případě U2.6, správce pouze jako kategorii předmětu zvolí sadu.

U2.8 Přidání předmětu do sady

Předpoklady: Správce je přihlášen do aplikace.

Cíl: V rámci dané sady bude evidován nově přidaný předmět.

Hlavní scénář:

1. Správce přejde na obrazovku pro editaci daného předmětu.
2. Správce zvolí sadu z nabídky a změnu potvrdí.

U2.9 Přidání kontrol SI do sady pomocí logu o jejich programování

Předpoklady: Správce je přihlášen do aplikace.

Cíl: V rámci dané sady budou evidovány kontroly, které byly obsaženy v daném logu, ale nebyly evidovány v systému.

Hlavní scénář:

1. Správce přejde na obrazovku pro editaci dané sady.
2. Správce zvolí příslušný log soubor z programu SPORTident Config+.
3. Správce potvrdí, že chce informace z poskytnutého souboru načíst.
4. Systém zpracuje soubor a přidá nově vytvořené kontroly do dané sady.

U2.10 Změna ceny zapůjčení daného předmětu či sady

Předpoklady:

1. Správce je přihlášen do aplikace.
2. Daný předmět či sada byl označen jako vypůjčitelný.

Cíl: U předmětu či sady bude aktualizován způsob nacenění jejich zapůjčení a případné související parametry vybraného způsobu.

Hlavní scénář:

1. Správce přejde na obrazovku pro editaci daného předmětu či sady.
2. Správce zvolí způsob nacenění.
3. Systém v závislosti na vybraném způsobu nacenění zobrazí editovatelné parametry tohoto způsobu.
4. Správce vyplní parametry způsobu nacenění a potvrdí provedené změny.

U2.11 Změna hranice pro hlášení vybité baterie kontroly SI

Předpoklady: Správce je přihlášen do aplikace.

Cíl: Změna hranice úrovně nabití, při které systém začne varovat správce, že je daná kontrola téměř vybitá.

Hlavní scénář:

1. Správce přejde na obrazovku s nastavením systému.
2. Správce změní příslušnou hodnotu.
3. Správce potvrdí provedenou změnu.

U2.12 Vygenerování faktury

Předpoklady:

1. Správce je přihlášen do aplikace.
2. Daná výpůjčka se vztahuje k události, která již proběhla.
3. Daná výpůjčka není označena jako uzavřená.
4. Správce v nastavení vyplnil fakturační údaje.

Cíl: Systém vygeneruje fakturu a přiřadí ji k dané výpůjčce.

Hlavní scénář:

1. Správce přejde na obrazovku pro editaci dané výpůjčky.
2. Správce zvolí možnost generovat fakturu.
3. Systém vygeneruje fakturu a přiřadí ji k dané výpůjčce.

Poznámka: Systém bude provádět generování faktur také automaticky.

U2.13 Záznam zaplacení výpůjčky

Předpoklady:

1. Správce je přihlášen do aplikace.
2. Daná výpůjčka má přiřazenou fakturu.

Cíl: Výpůjčka, u které uživatel zaplatil požadovanou částku, bude v systému vedena jako uzavřená.

Hlavní scénář:

1. Správce vybere ze seznamu příslušnou výpůjčku.
2. Správce zvolí možnost uzavřít výpůjčku.
3. Systém změní stav výpůjčky.

U2.14 Načtení informací o sportovních akcích ze systému ORIS

Předpoklady:

1. Správce je přihlášen do aplikace.
2. Správce v nastavení aplikace vyplnil zkratku oblasti.
3. V systému jsou načteny oddíly.

Cíl: V systému budou obsažena aktuální data o sportovních akcích pořádaných oddíly v oblasti.

Hlavní scénář:

1. Správce přejde na obrazovku s výpisem událostí.
2. Správce zvolí možnost načtení dat o sportovních akcích.
3. Systém načte požadovaná data ze systému ORIS.

U2.15 Nahrání informací o stavu kontrol SI

Předpoklady: Správce je přihlášen do aplikace.

Cíl: V systému budou obsažena aktuální hodnoty atributů kontrol SI.

Hlavní scénář:

1. Správce přejde na obrazovku s nastavením systému.
2. Správce zvolí příslušný log soubor z programu SPORTident Config+.
3. Správce potvrdí, že chce informace z poskytnutého souboru načíst.
4. Systém zpracuje soubor a aktualizuje příslušné hodnoty, pokud při zpracování narazí na kontrolu, která ještě v systému není evidována, přidá jí.

Volba platformy a hlavních technologií

5.1 Volba platformy

Z provedené analýzy je patrné, že výsledná aplikace nebude používána pouze správcem společného vybavení, který bude jejím prostřednictvím evidovat jednotlivé předměty, ale také zástupci jednotlivých oddílů v oblasti, kterými bude využívána pro realizaci výpůjček sad a předmětů na jimi pořádané sportovní akce. Je tedy potřeba umožnit všem těmto uživatelům bezproblémový přístup k funkcionalitám aplikace a datům v rámci ní uložených.

Data, nad kterými aplikace operuje, je tak nutné uchovávat takovým způsobem, aby k nim mohlo přistupovat více uživatelů zároveň, tedy na nějakém serveru.

Dále je nutné si ujasnit, jakou podobu bude mít vlastní aplikace. V tomto ohledu se nabízí následující možnosti.

1. Aplikace pro mobilní telefon či tablet
2. Desktopová aplikace
3. Webová aplikace

Tyto možnosti budou hodnoceny z hlediska uživatelské přívětivosti, rozšířenosti jim příslušné platformy, složitosti šíření aplikace a případných aktualizací a zkušeností autora s vývojem pro danou platformu.

1. Aplikace pro mobilní telefon či tablet

Chytrá mobilní zařízení jsou dnes poměrně běžnou záležitostí. Dle [8] používá chytrý mobilní telefon v České republice zhruba 65 % lidí a 29 % pak užívá zařízení typu tablet.

Šíření aplikace a jejích případných aktualizací je díky existenci distribučních služeb, jako jsou App Store či Google Play, poměrně snadné.

Hlavní problém tohoto řešení představuje uživatelská přívětivost. Jak bylo uvedeno výše, tablety používá méně jak třetina české populace, což znamená, že většinu mobilních zařízení představují chytré telefony. Ty oproti počítačům či tabletům disponují nepoměrně menšími displeji, což by práci s aplikací činilo pro koncového uživatele nepohodlnou a to především pro správce, který musí mít v rámci systému přehled o větším množství záznamů.

Pro uživatele aplikace by bylo také složitější nahrání log souboru o programování kontrol SI, které je v rámci výpůjčky povinné. Tento log soubor je výstupem desktopové aplikace, což znamená, že by byl uživatel nucen tento soubor nejdříve přenést do svého mobilního zařízení a teprve potom ho nahrát do systému.

Další nevýhodu představuje existence různých operačních systémů mobilních zařízení. Ačkoliv se dnes jedná v podstatě pouze o operační systémy Android a iOS, může tato skutečnost představovat problém. U aplikace je potřeba, aby byla přístupná pro všechny zástupce oddílů v oblasti, aby správce nebyl zbytečně nucen realizovat výpůjčky v systému jménem jiného uživatele. To ovšem znamená, že by bylo nutné vytvořit aplikaci pro oba operační systémy a vzhledem k tomu, že autor práce momentálně nedisponuje zkušenostmi s vývojem mobilních aplikací pro libovolný OS, jeví se tato možnost jako značně problematická.

2. Desktopová aplikace

Ačkoliv poslední dobou roste čím dál tím více oblíbenost mobilních zařízení, představuje osobní počítač stále hojně využívanou platformu a to především díky větší přehlednosti prezentovaných dat, která je umožněna větším zobrazovacím zařízením.

V případě této možnosti ovšem vyvstává problém se šířením případných aktualizací, neboť aplikace by byla v tomto případě lokálně instalována na osobních počítačích jednotlivých uživatelů.

Zkušenosti autora s vývojem desktopových aplikací jsou výraznější než u ostatních zkoumaných možností, neboť se jedná o typ aplikace, se kterým se v rámci studia setkal nejčastěji.

3. Webová aplikace

Poslední možností je zpřístupnění funkcionality aplikace prostřednictvím webového klienta.

Tato možnost není závislá na platformě, neboť k využití plné funkcionality systému stačí uživateli v celku libovolný webový prohlížeč nainstalovaný na ním vybraném zařízení. Uživatel tedy může k aplikaci přistupovat jak z osobního počítače, který umožní zobrazit požadovaná data velmi přehledně, tak

i z mobilního telefonu či tabletu, kde může například provádět některé jednodušší úkony. Aplikaci je možné v tomto ohledu optimalizovat využitím tzv. responsivního designu.

V případě webové aplikace taktéž odpadá starost s aktualizací aplikace, neboť veškerý kód aplikace je přítomen pouze na serveru, na kterém je aplikace zprovozněna. Není tak nutné zajišťovat distribuci aktualizací k jednotlivým uživatelům.

Také znalosti autora jsou v tomto případě na poměrně obстойné úrovni.

Závěr

Nyní je možné přistoupit k vyhodnocení jednotlivých možností a volbě konečné podoby řešení.

Jak je zřejmé z následující tabulky, vychází v tomto případě jako nejlepší řešení webová aplikace, která netrpí nedostatky ostatních možností.

	Uživatelská přívětivost	Rozšířenost platformy	Šíření aplikace	Znalosti autora
Aplikace pro mobilní telefon či tablet	-1	1	1	-2
Desktopová aplikace	1	1	-2	1
Webová aplikace	2	2	2	0

Tabulka 5.1: Hodnocení jednotlivých možností podoby aplikace

5.2 Volba implementačních technologií

Poté co byla vyjasněna konečná podoba výsledné aplikace, je možné začít uvažovat o technologiích, které budou použity při její implementaci.

V případě této práce budou zvažovány následující technologie.

1. C#
2. PHP + WordPress
3. Apex

V případě prvních dvou možností se jedná o technologie, s nimiž se již autor této práce při tvorbě webových aplikací setkal.

Poslední možnost pak představuje technologii, která byla v počátečních fázích projektu navržena zadavatelem.

1. C#

C# představuje programovací jazyk vyvinutý společností Microsoft, který se řadí mezi momentálně nejoblíbenější programovací jazyky jak dle TIOBE indexu [9] tak i dle PYPL [10]. Jako takový lze C# využít pro tvorbu širokého spektra aplikací od desktopových, přes mobilní (Xamarin) až po webové (ASP.NET, ASP.NET MVC).

C# je na tom také velmi dobře co se týče podpory, neboť disponuje kvalitní dokumentací přímo od Microsoftu a díky většímu množství uživatelů i mnoha dalšími tutoriály. Rovněž na stackoverflow.com má dle každoročního průzkumu [11] obstojnou uživatelskou základnu.

C# ovšem pokulhává, co se týče nákladů vynaložených za hosting, které jsou výrazně vyšší oproti například PHP.

2. PHP + WordPress

PHP rovněž můžeme najít na předních příčkách TIOBE [9] i PYPL [10] a jeho uživatelská základna na stackoverflow.com je srovnatelná s jazykem C#. [11]

Taktéž disponuje poměrně přehlednou dokumentací a díky početné komunitě i množstvím tutoriálů, rad a dalších tipů, které potenciálnímu vývojáři pomohou překonat případné problémy, se kterými se může v průběhu vývoje aplikace setkat.

Jako takové představuje PHP nástroj pro snadnou a rychlou tvorbu webových aplikací a i přes odpor části programátorské komunity k tomuto jazyku se stále jedná o nejčastěji využívanou technologii v oblasti vývoje webových aplikací. Díky tomu není u PHP rozhodně nouze o rozšiřující knihovny či různé frameworky. Také náklady na provoz jsou relativně nízké, díky více než dostatečné nabídce poměrně levných web hostingů.

V rámci tohoto projektu se navíc v případě PHP nabízí možnost realizovat požadovanou aplikaci jako plugin pro CMS WordPress, který je postaven právě na PHP. Tato možnost se jeví jako výhodná především z toho důvodu, že web oblasti Vysočina (jako i weby většiny ostatních oblastí) využívá právě zmíněný CMS. Pro CMS WordPress navíc plánuje autor této práce vytvořit také další plugin, který by umožňoval snazší zpracování bodového hodnocení dlouhodobých soutěží oblastí. Všechny nástroje usnadňující organizační úkony v oblasti by tak byly sjednoceny v rámci jednoho systému, díky čemuž by bylo dosaženo snadnější udržitelnosti těchto nástrojů. Tato možnost by byla navíc také přínosem z pohledu uživatelů aplikace, kteří by k aplikaci mohli přistupovat přímo na webu oblasti.

Další výhodou tohoto řešení je také jednoduchost nasazení výsledné aplikace a šíření případných aktualizací.

Autor této práce rovněž disponuje zkušenostmi jak s programovacím jazykem PHP, tak i se zmiňovaným systémem WordPress.

3. Apex

Apex je nástroj určený pro vývoj webových aplikací, který je úzce integrován s databází společnosti Oracle a umožňuje rychle vytvořit bezpečnou a snadno rozšiřitelnou aplikaci a to pouze s využitím webového prohlížeče a určitou znalostí jazyka SQL, případně PL/SQL. [12]

Jako takový je Oracle Apex v porovnání s jinými technologiemi pro tvorbu webových aplikací poměrně málo využívaný. Disponuje tedy také menší komunitou a omezeným množstvím tutoriálů.

Autor práce v tomto případě nemá žádné znalosti či zkušenosti týkající se vývoje aplikací s využitím tohoto nástroje a vzhledem k malému množství návodů a dalších materiálů zabývajících se problematikou vývoje s tímto nástrojem, se jeví možnost volby tohoto nástroje pro realizaci tohoto projektu jako nepříliš vhodná alternativa.

Závěr

Nyní je tedy možné přistoupit ke konečné volbě implementační technologie. Oracle Apex se v případě tohoto projektu jeví jako nepříliš vhodná volba a to především kvůli chybějícím zkušenostem autora s touto technologií a omezeným množstvím materiálů zabývajících se vývojem aplikací s pomocí tohoto nástroje.

C# se oproti tomu zdá jako relativně dobrá volba, ovšem výrazně vyšší cena za hosting a menší zkušenosti autora s tímto jazykem v porovnání s PHP vrhají na tuto technologii poněkud nepříznivé světlo.

Jako nejlepší možná volba se v případě tohoto projektu jeví druhá možnost, tedy realizovat požadovanou aplikaci s pomocí PHP jako plugin pro CMS WordPress. U této možnosti jsou minimalizovány náklady díky skutečnosti, že oblast Vysočina již disponuje hostingem, na kterém provozuje svůj web poháněný právě systémem WordPress, a také znalosti autora jsou v tomto případě nejvýraznější.

Návrh

Nyní, když je zvolena jak platforma výsledné aplikace, tak i hlavní technologie, které budou použity při její implementaci, je možné přistoupit k vlastnímu návrhu aplikace.

6.1 Doménový model

Prvním krokem v tomto případě bude identifikace jednotlivých entit problémové domény a vztahů mezi nimi.

Při vlastní identifikaci entit systému byly částečně využity poznatky získané v rámci předmětu BI-DBS, ve kterém autor vytvářel jako semestrální práci návrh databázového úložiště, jehož doména byla inspirována právě problémovou doménou tohoto projektu. Vzniklé logické schéma databáze je vyobrazeno na obrázku 6.1 na straně 30.

Vlastní identifikované entity jsou pak znázorněny pomocí UML diagramu na obrázku 6.2 na straně 31.

Za povšimnutí stojí atribut Status u entity výpůjčky. Ten může v rámci systému nabývat hodnot „Rezervace“, „Částečná“, „Kompletní“ a „Uzavřená“. První tři hodnoty jsou výpůjčce přiřazovány na základě statusu jednotlivých předmětů, které k ní jsou přiřazeny. Ve stavu „Rezervace“ setrvává výpůjčka až do doby, než je uživatelem převzat první v rámci ní rezervovaný předmět, kdy je status výpůjčky změněn na „Částečná“. V okamžiku, kdy jsou všechny předměty rezervované v rámci výpůjčky převzaty, je pak výpůjčka označena jako „Kompletní“. Status „Uzavřená“ je pak výpůjčce přiřazen ručně správcem vybavení po obdržení platby za danou výpůjčku.

6.2 Reprezentace entit v systému

Vzhledem ke skutečnosti, že je aplikace implementována jako plugin pro systém WordPress, lze pro reprezentaci entit v systému využít vlastních typů

příspěvků. Takto vytvořené typy příspěvků umožňují využívat veškeré funkcionality, které je možné pozorovat u výchozích typů, tedy u příspěvků a stránek.

Především je možné v systému vytvářet nové příspěvky a upravovat, či mazat již existující.

Pro potřeby tohoto projektu byly v rámci systému vytvořeny následující typy příspěvků:

1. `isst_item`,
2. `isst_event`,
3. `isst_rental`.

Tyto přidané typy příspěvků reprezentují popořadě předměty a sady, které sdílejí stejný typ příspěvku, události a nakonec výpůjčky.

Těmto typům příspěvků jsou pak s pomocí pluginu Advanced Custom Fields (ACF) a vlastních taxonomií přidány potřebné atributy.

WordPress již sám o sobě umožňuje správu uživatelů, čímž odpadá potřeba definovat nový typ příspěvku reprezentujícího uživatele.

Dále ve výčtu chybí typ pro oddíl. Ten nebyl vytvořen, neboť informace o všech oddílech jsou dostupné prostřednictvím REST API informačního systému ORIS. Jejich načtení do systému tak může proběhnout zcela automaticky a není tak potřeba umožnit vytvářet v rámci systému nové oddíly.

Využití vlastních typů příspěvků navíc řeší z velké části problém, jak informace obsažené v systému ukládat do databáze. Veškeré informace ohledně těchto typů jsou totiž ukládány do WordPressem definovaných databázových tabulek.

Zbývá ovšem problém, jak v rámci databáze reprezentovat vztahy výpůjčka-předmět a sada-samostatný předmět. Pro tyto vztahy tak byly vytvořeny nové databázové tabulky, jejichž struktura je zobrazena na obrázcích na straně 32.

Tabulka na obrázku 6.3 představuje tabulku pro uložení vztahu výpůjčka-předmět. Jako taková kromě `id` výpůjčky a `id` předmětu obsahuje také položky `status` a `change_date`. `Status` představuje stav předmětu v rámci výpůjčky a v rámci systému může nabývat hodnot „rezervováno“, „vypůjčeno“ nebo „předáno“. `Change_date` poté obsahuje datum poslední změny statusu.

Druhá tabulka vyobrazená na obrázku 6.4 umožňuje uložení vztahu sada-samostatný předmět.

Poslední vyobrazená tabulka (obrázek 6.5) pak umožňuje ukládat do databáze jednotlivé oddíly.

6.3 Uživatelské rozhraní

Vzhledem k tomu, že aplikace bude realizována jako plugin pro WordPress, nabízí se zde možnost zapojení uživatelského rozhraní výsledné aplikace přímo do uživatelského rozhraní daného CMS.

Uživatelské rozhraní se v případě tohoto projektu bude skládat převážně z obrazovek obsahujících tabulkové výpisy jednotlivých doménových entit a obrazovek pro editaci jednotlivých položek v systému.

Obrazovky pro editaci příspěvků vytváří WordPress automaticky. Při využití pluginu ACF pak v rámci této aplikace nebude zapotřebí tyto obrazovky upravovat vůbec, nebo bude zapotřebí spíše menších změn.

WordPress také disponuje třídou `WP_List_Table`, která je využívána pro generování výchozích tabulek WordPressu, a kterou je možné využít k nadefinování tabulek vlastních. Plusem takového přístupu je fakt, že nově definované tabulky disponují stejným vzhledem jako ty výchozí. Uživatelské rozhraní tak zůstává v rámci systému konzistentní.

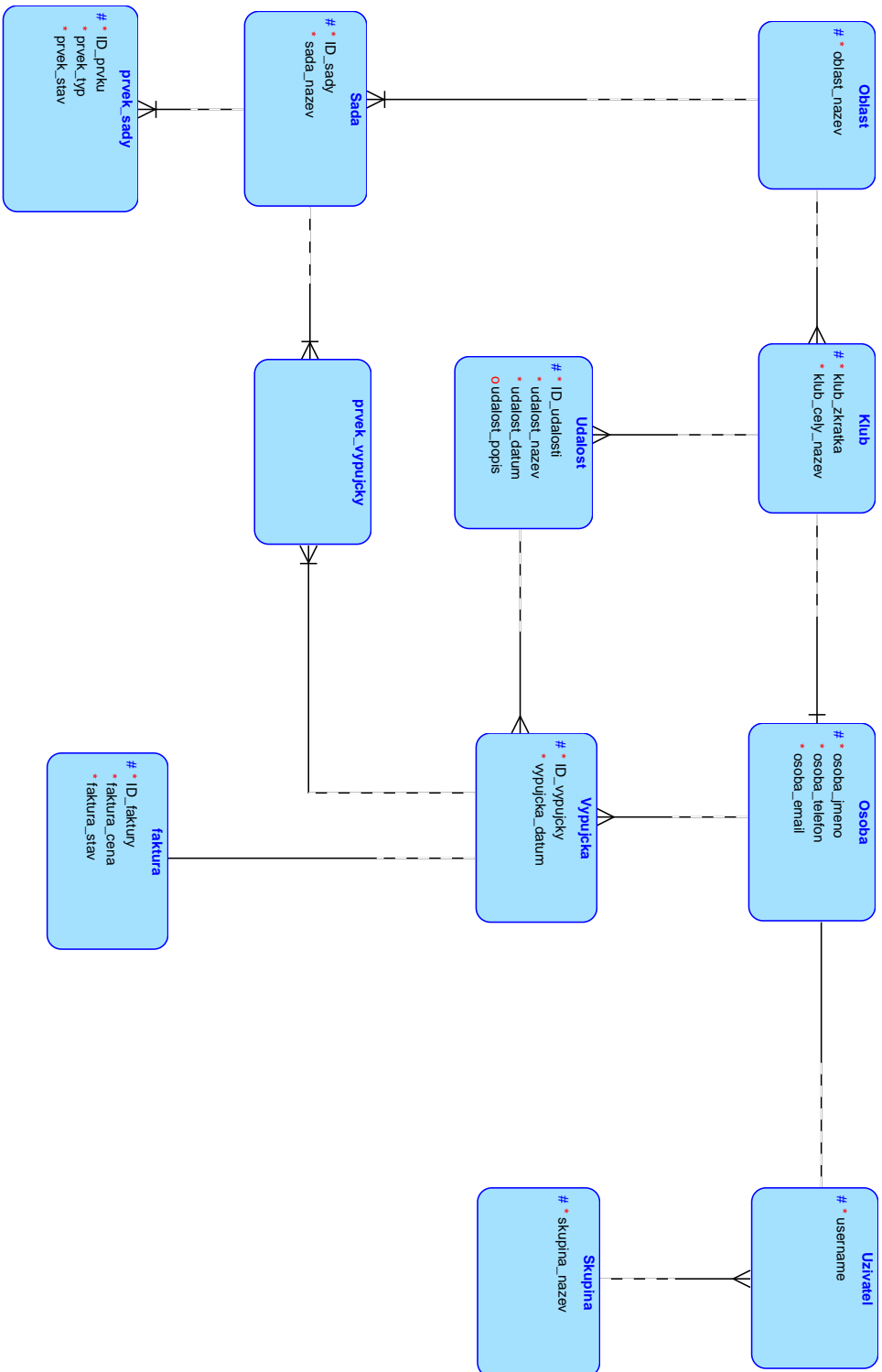
Obrazovka typu dashboard

Jedním z požadavků kladených na systém je existence obrazovky typu dashboard, která by zobrazovala nejdůležitější informace relevantní pro daného uživatele.

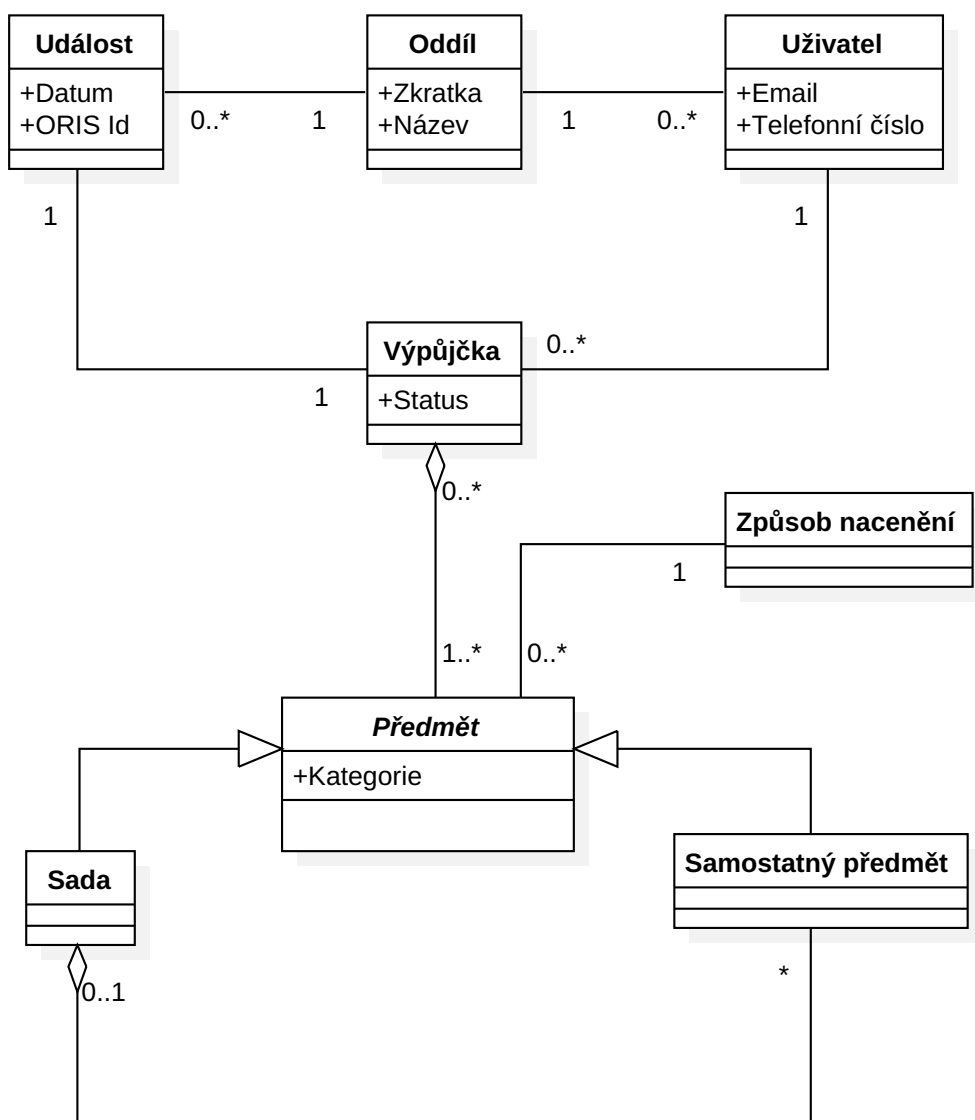
Pro potřeby správce vybavení tak na této obrazovce budou zobrazeny tabulky obsahující informace o aktuálně vypůjčených předmětech, výpůjčkách, u kterých je očekávána platba, a seznam kontrol, jejichž míra nabití je nízká.

Běžný uživatel pak na této obrazovce prozatím nalezne pouze seznam svých výpůjček, u kterých je očekáváno jejich zaplacení.

6. NÁVRH



Obrázek 6.1: Logické schéma vytvořené v rámci BI-DBS



Obrázek 6.2: UML diagram entit

isst.wp_jsst_rental_items		
P *	rental_id	INTEGER
P *	item_id	INTEGER
*	status	VARCHAR2 (50)
*	change_date	DATE
PRIMARY (rental_id, item_id)		
rental_id (rental_id)		

Obrázek 6.3: Databázová tabulka – vztah mezi vypůjčenou položkou a výpůjčkou

isst.wp_jsst_set_items		
P *	item_id	INTEGER
*	set_id	INTEGER
PRIMARY (item_id)		
set_id (set_id)		

Obrázek 6.4: Databázová tabulka – vztah mezi předmětem a sadou

isst.wp_jsst_clubs		
P *	abbr	VARCHAR2 (3)
*	name	VARCHAR2 (150)
PRIMARY (abbr)		

Obrázek 6.5: Databázová tabulka – kluby

Implementace

7.1 Použité technologie

WordPress Plugin Boilerplate

WordPress Plugin Boilerplate [13] představuje standardizovaný, organizovaný a objektově orientovaný základ pro tvorbu pluginů pro CMS WordPress.

Jako taková přináší tato šablona předdefinovanou přehlednou adresářovou strukturu pluginu a základní strukturu vlastního kódu pluginu, která následuje zásady objektově orientovaného návrhu. Šablona tak poskytuje velmi přehledný základ pro vývoj nového pluginu.

Advanced Custom Fields

Advanced Custom Fields je plugin pro CMS WordPress, který umožňuje přidání vlastních polí k příspěvkům v rámci systému. Tato jeho funkcionalita navíc není přístupná pouze prostřednictvím uživatelského rozhraní, ale je také možné přidat požadovaná pole pomocí PHP kódu. Lze ho tedy snadno využít k přidání polí i v rámci šablon či pluginů.

Existence uživatelského rozhraní pro přidávání polí a možnost exportu PHP kódu pro jejich přidání pak umožňuje snadné přidání a otestování funkcionality těchto polí v rámci procesu vývoje pluginu či šablony.

V rámci tohoto projektu byly mimo typy polí, které jsou již v rámci ACF definovány, využity také další, které jsou odvozeny z oněch již definovaných a mírně upravují jejich funkcionalitu. Příkladem může být například typ pole „Taxonomy on new“, které se od výchozího typu, který umožňuje přidání taxonomie k příspěvku, liší tím, že umožňuje tuto taxonomii přidat pouze při vytváření nového příspěvku. Při editaci příspěvku tedy již není možné hodnotu tohoto pole změnit.

Dále byl využit také typ pole „Relationship“ a to pro realizaci prvku uživatelského rozhraní na obrazovce pro editaci výpůjčky, který slouží pro přidání

předmětů k dané výpůjčce. Tento prvek jako takový nepředstavuje ACF pole, ale využívá uživatelského rozhraní definovaného pro výše zmíněný typ pole.

TCPDF

TCPDF představuje PHP nástroj pro tvorbu PDF souborů bez nutnosti instalace externí knihovny (např. PDFLib), který výsledná aplikace využívá pro generování faktur na úhradu jednotlivých výpůjček zaznamenaných v systému.

7.2 Propojení se systémem ORIS

Jak již bylo v této práci zmíněno, ORIS představuje páteří informační systém Českého svazu orientačních sportů. Jako takový obsahuje informace o jednotlivých oddílech, jimi pořádaných sportovních akcích, registrovaných závodnících atd. Pro potřeby tohoto projektu bylo zapotřebí zařídit, aby aplikace byla schopna z tohoto systému získávat data potřebná pro svůj chod. Jedná se především o

1. údaje o jednotlivých sportovních akcích,
2. výsledky dané sportovní akce,
3. údaje o jednotlivých oddílech.

Pro získání vlastních dat bylo využito veřejně přístupné REST API systému ORIS, které umožňuje přístup k těmto datům pomocí GET požadavku obsahujícím požadované parametry. Seznam možných parametrů je přístupný na [14]. Dle doporučení v [15] pak byla při provádění vlastního požadavku v aplikaci použita funkce `wp_remote_get`.

7.3 Uživatelské rozhraní

Jak již bylo zmíněno v kapitole zabývající se návrhem, WordPress disponuje třídou `WP_List_Table`, která může být v rámci pluginů využita pro definici jejich vlastních tabulek. Tato třída byla označena jako soukromá, což znamená, že jako taková není určena, aby byla přímo využívána vývojáři v pluginech, neboť v budoucích aktualizacích WordPressu může dojít k jejím změnám. Z toho důvodu je potřeba danou třídu zahrnout přímo do pluginu. [16]

V rámci této práce pak byla s využitím [17] vytvořena odvozená tabulka, která umožňuje zobrazení dat s využitím technologie AJAX. Tento krok byl potřeba například u tabulek, které zobrazují předměty na obrazkách pro editaci výpůjčky a sady, neboť přidání těchto tabulek je možné pouze s využitím API, které WordPress pro editační obrazovky příspěvků definuje. Prvky tohoto API pro editační obrazovky se ovšem všechny nalézají uvnitř HTML

formuláře dané editační obrazovky. To má za následek, že po přidání tabulky, která musí být rovněž obalena v HTML formuláři, dojde k zanoření dvou formulářů do sebe, což není dle [18] podporováno.

Autor práce by se pak také rád zmínil o zajímavém problému, se kterým se při vývoji uživatelského rozhraní setkal.

Daný problém měl za následek to, že uživatel byl po zvolení možnosti pro vytvoření nového příspěvku (např. výpůjčky) na obrazovce s autorem definovanou tabulkou, systémem WordPress upozorněn, že pro vytvoření nového příspěvku nemá dostatečná oprávnění, ačkoliv mu tato oprávnění byla přiřazena při aktivaci pluginu.

Toto chování je dle [19] způsobeno tím, že byly autorem práce z menu WordPressu odstraněny při registraci nového typu příspěvku defaultně generované položky, které kromě odkazu na zobrazení výčtu příspěvků daného typu obsahují také odkaz pro přidání nového příspěvku tohoto typu. Právě nepřítomnost tohoto druhého zmíněného odkazu v menu má za následek, že ačkoliv má uživatel dostatečná oprávnění, není schopný přidat nový příspěvek.

Ve výsledné aplikaci tak bylo zapotřebí uplatnit v [19] navrhované řešení, které umožní ošálit systém WordPress tak, aby si myslel, že v menu požadované položky jsou.

Testování

Testování představuje nedílnou součást vývoje software, která má zajistit, že funkcionality a vlastnosti systému odpovídají požadavkům, které byly na počátku projektu definovány zadavatelem.

Testování aplikace v tomto případě probíhalo, stejně jako její vývoj, lokálně na počítači se systémem Windows a to s pomocí nástroje XAMPP. Vlastní testy pak byly prováděny přímo autorem práce a následovaly případy užití, které byly popsány dříve.

Vzhledem ke skutečnosti, že jednotlivé funkcionality systému byly do vyvíjené aplikace přidávány iterativně, byly autorem této práce prováděny také občasně regresní testy, jejichž smyslem je ověření, že provedené změny či implementace nových vlastností v aplikaci neměly žádný vliv na stávající funkce a vlastnosti vyvíjeného software. [20]

V rámci této práce nebylo provedeno uživatelské testování, které ovšem autor této práce zamýšlí provést před vlastním nasazením aplikace. Princip této metody testování spočívá v tom, že je pozorováno chování samotných uživatelů, čímž mohou být odhaleny chyby, které zůstaly vývojářům skryté. [21]

Další rozvoj aplikace

9.1 Filtry a akce

Akce a filtry představují dva typy takzvaných háků, které v rámci systému WordPress umožňují určité části kódu interagovat s / modifikovat jinou část kódu. [22]

Akce umožňují zavolání definované funkce při běhu programu v místě, kde je definován příslušný hook.

Filtry slouží k modifikaci zpracovávaných dat prostřednictvím jim přiřazené funkce, která přijme určitou proměnnou jako parametr, provede její transformaci a následně předá jako svojí návratovou hodnotu.

Přidáním hooks do pluginu je tedy umožněno dalším vývojářům upravit funkcionality pluginu bez přímého zásahu do samotného kódu pluginu.

Využití hooks by v rámci výsledné aplikace bylo vhodné například v rámci třídy `Isst_Pricing_Helper`. V této třídě by bylo možné definovat filtr pro úpravu pole obsahujícího způsoby nacenění výpůjček, což by umožnilo jednoduše přidat nový způsob, či odebrat již existující.

V rámci `Isst_Pricing_Helper` by bylo dále možné nadefinovat akci, která by byla provedena po vygenerování faktury. Na tuto akci by pak šlo navěsit funkci, která by například zaslala danou fakturu příslušnému uživateli na email.

9.2 Automatizované přiřazování plateb

Jedním z hlavních cílů aplikace vyvinuté v rámci této práce je usnadnit správci co nejvíce evidenci předmětů a výpůjček těchto předmětů.

S výslednou aplikací již není správce nucen si ručně vést záznamy o výpůjčkách, stavu kontrol SPORTident či vystavovat fakturu na úhradu dané vý-

půjčky. Zůstává tu ovšem potřeba kontrolovat, jestli byla fakturovaná částka daným uživatelem skutečně zaplacená. Z toho důvodu se jeví jako poměrně vhodné vylepšení aplikace přidání možnosti automatického přiřazování plateb k výpůjčkám.

Po implementaci této funkcionality by systém byl schopný využitím API vystaveného příslušnou bankovní organizací, získávat data o platbách a na základě např. variabilního symbolu přiřadit platbu k dané výpůjčce a následně ji uzavřít (označit jako zaplacenou).

9.3 Uchování historie atributů

Dalším možným vylepšením aplikace by mohlo být ukládání historie atributů kontrol SPORTident. Uložené hodnoty by pak mohly být statisticky zpracovány. Tímto způsobem by bylo například možné zjistit, jak rychle kontroly spotřebovávají energii v závislosti na nastaveném módu ražení.

Uložení vlastní historie by mohlo být realizováno více způsoby.

Prvním z nich je možnost jednoduše ukládat log soubory, které uživatelé do systému nahrávají. Druhým možným řešením by pak mohlo být uložení historie atributů do samostatné databázové tabulky.

9.4 Kalendář událostí

Toto potenciální vylepšení spočívá v zobrazení kalendáře na dashboardu aplikace, kde by byly zaneseny jednotlivé události. Pro vlastní realizaci tohoto vylepšení by bylo možné využít například open-source jQuery plugin FullCalendar.

Závěr

Cílem této práce bylo navrhnout a implementovat informační systém pro krajskou sekci Českého svazu orientačních sportů, sloužící k evidenci majetku a výpůjček tohoto majetku.

V rámci této práce provedl autor analýzu požadavků zadavatele a řešerši existujících řešení, z nichž vyplynulo, že bude zapotřebí implementovat řešení vlastní.

Dále bylo tedy v práci přistoupeno k volbě platformy, pro kterou bude výsledná aplikace realizována, a zvolení technologií vhodných pro její implementaci. Následně autor přistoupil k návrhu a implementaci vlastního řešení v podobě pluginu pro CMS WordPress.

V závěru práce se pak autor zabývá procesem testování a návrhem možných budoucích vylepšení implementovaného řešení.

Výstupem této práce je především samotná aplikace, která umožní zjednodušení a zpřehlednění procesu evidence výpůjček vybavení v rámci oblasti Vysočina. Další důležitou součástí výstupu práce pak představují navržená potenciální vylepšení výsledné aplikace, jejichž implementací se autor může v budoucnu zabývat.

Bibliografie

1. *Snipe-It – Free Open Source Asset Management Software [online]*. 2018. Dostupné také z: <https://snipeitapp.com>.
2. *Sharing Club – WordPress Plugins [online]*. 2018. Dostupné také z: <https://snipeitapp.com>.
3. *Appointment, Reservation and Rental Booking for Woocommerce – WordPress Plugins [online]*. 2018. Dostupné také z: <https://wordpress.org/plugins/sharing-club/>.
4. WILSON, William M. Writing Effective Natural Language Requirements Specifications. *Crosstalk: The Journal of Defense Software Engineering [online]*. 1999, s. 16 [cit. 2018-03-27]. Dostupné z: <http://profinit.eu/wp-content/uploads/2016/03/WritingEffectiveSRS.pdf>.
5. *Nefunkční požadavky – PM Consulting*. [online] [cit. 2018-03-27]. Dostupné z: <http://www.pmconsulting.cz/slovníkovy-pojem/nefunkcni-pozadavky/>.
6. OCHODEK, Mirosław; KORONOWSKI, Krystian; MATYSIAK, Adam; MIKLOSIK, Piotr; KOPCZYŃSKA, Sylwia. *Sketching Use-Case Scenarios Based on Use-Case Goals and Patterns*. 2017. ISBN 978-3-319-43605-0. Dostupné také z: https://www.researchgate.net/publication/307089799_Sketching_Use-Case_Scenarios_Based_on_Use-Case_Goals_and_Patterns.
7. COCKBURN, Alistair. *WRITING EFFECTIVE USE CASES [online]*. Addison-Wesley, 2001 [cit. 2018-05-01]. ISBN 02-017-0225-8. Dostupné z: <http://alistair.cockburn.us/get/2465>.
8. *Consumer barometr [online]* [cit. 2018-04-23]. Dostupné z: https://www.consumerbarometer.com/en/graph-builder/?question=M1&filter=country:czech_republic.

9. *TIOBE Index* [online] [cit. 2018-04-24]. Dostupné z: <https://www.tiobe.com/tiobe-index/>.
10. *PYPL PopularitY of Programming Language* [online]. 2016 [cit. 2018-04-24]. Dostupné z: <http://pypl.github.io/PYPL.html>.
11. *Developer survey results 2018* [online] [cit. 2018-05-01]. Dostupné z: <https://insights.stackoverflow.com/survey/2018#most-popular-technologies>.
12. Co je vlastně Oracle APEX? [online] [cit. 2018-04-24]. Dostupné z: <http://www.digitec.cz/cs/co-je-vlastne-oracle-apex>.
13. *Wordpress Plugin Boilerplate* [online] [cit. 2018-05-12]. Dostupné z: <http://wppb.io>.
14. *ORIS API* [online]. Dostupné také z: <https://oris.orientacnisporty.cz/API>.
15. ROSS, Elliot. 7 Simple Rules: WordPress Plugin Development Best Practices [online]. 2011 [cit. 2018-04-29]. Dostupné z: <https://code.tutsplus.com/articles/7-simple-rules-wordpress-plugin-development-best-practices--wp-22996>.
16. *Class Reference/WP List Table* [online] [cit. 2018-05-12]. Dostupné z: https://codex.wordpress.org/Class_Reference/WP_List_Table.
17. *wp-list-table-ajax-sample* [online] [cit. 2018-05-12]. Dostupné z: <https://github.com/debba/wp-list-table-ajax-sample>.
18. *Forms* [online] [cit. 2018-05-12]. Dostupné z: <https://www.w3.org/TR/html5/forms.html#the-form-element>.
19. *User cannot Add New for Custom Post Type* [online] [cit. 2018-05-12]. Dostupné z: <https://wordpress.org/support/topic/user-cannot-add-new-for-custom-post-type/>.
20. HLAVA, Tomáš. Progresní a regresní testy. *Testování softwaru* [online]. 2011 [cit. 2018-05-12]. Dostupné z: <http://testovanisoftwaru.cz/metodika-testovani/druhy-typy-a-kategorie-testu/progresni-a-regresni-testy/>.
21. KRUG, Steve. *Nenutte uživatele přemýšlet!: Praktický průvodce testováním a opravou chyb použitelnosti webu*. Computer Press, 2010. ISBN 9788025129234.
22. *Hooks* [online] [cit. 2018-05-12]. Dostupné z: <https://developer.wordpress.org/plugins/hooks/>.

Seznam použitých zkratek

- CMS** Content Management System
- API** Application Programming Interface
- UML** Unified Modeling Language
- ČSOS** Český svaz orientačních sportů
- SI** SPORTident
- SRS** Software Requirments Specification
- OS** Operating System
- ACF** Advanced Custom Fields
- AJAX** Asynchronous JavaScript and XML
- HTML** HyperText Markup Language

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src	
├── isst.....	zdrojové kódy implementace
├── thesis.....	zdrojová forma práce ve formátu L ^A T _E X
silogs	ukázkové log soubory z programu SPORTident Config+
text	text práce
├── thesis.pdf.....	text práce ve formátu PDF