



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**MODUL PRO SLEDOVÁNÍ MILNÍKŮ PRO
KANBOARD.ORG**

MILESTONE TRACKING PLUGIN FOR KANBOARD.ORG

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PATRIK HOMA

VEDOUcí PRÁCE

SUPERVISOR

Ing. FRANTIŠEK ŠČUGLÍK, Ph.D.

BRNO 2019

Zadání bakalářské práce



21905

Student: **Homa Patrik**
Program: Informační technologie
Název: **Modul pro sledování milníků pro Kanboard.org**
Milestone Tracking Plugin for Kanboard.org

Kategorie: Web

Zadání:

1. Seznamte se s nástrojem Kanboard.org a s technologiemi PHP a SQLite.
2. Nastudujte možnosti tvorby vestavných modulů pro Kanboard.org.
3. Navrhněte modul pro vytváření a sledování stavu projektových milníků. V návrhu zohledněte možnost návázat milníky na existující úkoly a jejich vizualizaci na zvolené časové ose.
4. Po konzultaci s vedoucím navržené řešení implementujte a otestujte. Dosažené výsledky zhodnoťte.

Literatura:

- Dokumentace Kanboard.org [online], Dostupné z: <https://docs.kanboard.org/en/latest/>
- Welling, L., Thomson, L.: PHP a MySQL Kompletní průvodce vývojáře, CPress, 2017
- Kroenke, D., M., Auer, D. J.: Databáze, CPress, 2014

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Ščuglík František, Ing., Ph.D.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 30. října 2018

Abstrakt

Cílem této bakalářské práce je vytvoření vestavěného modulu pro Kanboard.org, který si klade za cíl implementaci časových či finančních milníků do již existujícího pole úkolu a rozvržení práce. V práci je použita metoda Kanban. Implementace je provedena pomocí technologií PHP a SQLite.

Abstract

The aim of this thesis is to create a built-in module for Kanboard.org, which aims to implement time or financial milestones into an existing task field and work layout. The Kanban method is used in this work. Implementation is done using PHP and SQLite technologies.

Klíčová slova

Kanboard.org, Kanban, Kanboard modul, PHP, SQLite, Databáze, Webová aplikace, time management, milníky

Keywords

Kanboard.org, kanban, Kanboard plugin, PHP, SQLite, Databasis, Web application, time management, milestones

Citace

HOMA, Patrik: *Modul pro sledování milníků pro Kanboard.org*, Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. František Ščuglík, Ph.D.

Modul pro sledování milníků pro Kanboard.org

Prohlášení

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

.....

Patrik Homa

29. července 2019

Poděkování

Zde bych chtěl poděkovat svému vedoucímu panu Ing. Františku Ščuglíkovi Ph.D. za poskytnuté konzultace a rady

Obsah

Obsah	1
1 Úvod	3
2 Kanban a aplikace Kanboard	4
2.1 Kanban	4
2.2 Kanban Board	5
2.2.1 Instalace	5
2.2.2 Použití	6
3 Využívané technologie	8
3.1 PHP	8
3.1.1 Model – View – Controller	8
3.2 Databázové technologie	9
3.2.1 SQLite	10
3.2.2 MySQL	10
3.2.3 Postgres	10
3.3 CSS	11
4 Milníkový modul	12
4.1 Návrh	12
4.1.1 Návrh databáze	12
5 Implementace modulu Milestones	15
5.1 Způsob vytváření modulů	15
5.1.1 Adresářová struktura modulu	15
5.1.2 Registrace modulu	16
5.1.3 Jazykové mutace	17
5.1.4 Databáze	17
5.2 Implementace	17
5.2.1 Registrace modulu	17
5.2.2 Zobrazování obsahu	18
5.2.3 Šablony	20
5.2.4 Modely	21
6 Ovládání	23
6.1 Seznam milníků	23
6.2 Přehled milníků a projektů	23
7 Testování	26
8 Možnost rozšíření	27

8.1	Hodinová mzda.....	27
9	Závěr.....	28

Kapitola 1

Úvod

V dnešním moderním světě nemůže žádný větší projekt vzniknout bez systematického řízení. Každá skupina osob spolupracujících na dosažení stejných cílů je ve společnosti nějak organizována. Řízení projektů vyžaduje okamžitý přehled o cílech úkolů, které jsou představovány pomocí milníků. Milníků je třeba dosáhnout, abychom se přiblížili cíli. Těchto významných událostí v průběhu projektu musí být dosaženo, co možná nejefektivněji. K tomu by měl také sloužit tento modul.

Když v 70 - tých a 80 - tých letech začala Kanbanu přibývat pozornost vědců a společností z celého světa, a to navzdory tomu, že už byl pár desítek let na světě, stalo se z něj časté téma i příklad pro moderní řízení projektů. Není divu, produkční úspěchy Japonska a to zvláště v oblasti nových technologií, inspirovali mnoho západních společností k implementaci kontrolních systémů kanban a díky tomu pak zefektivnění řízení zdrojů. Již v té době bylo zřejmé jaký má kanban potenciál. Mnoho empirických studií se začalo zabírat možnostmi upgradů a modelů pro produkční softwary na bázi kanbanu[1]. Aby společnost mohla těžit z výhod kanbanu, musí dojít k nastavení provozního systému, blížíciho se optimu.

Kapitola 2

Kanban a aplikace Kanboard

V následující kapitole bude popsána metodika Kanban, na které je tato práce založena. Dále se bude věnovat aplikaci Kanboard.org.

2.1 Kanban

Kanban je základní metoda pro řízením a vedením práce v lidských systémech. Je založena na řízení poptávky po práci a kapacitě lidského kapitálu. Koncept Kanbanu pochází od Japonského autora Taiichiho Ohna [2], který jej vytvořil pro automobilový koncern Toyota již v padesátých letech minulého století. Kanban měl sloužit jako opora pro výrobní systém „just-in-time“ (dále také jako JIT či TPS). Původním cílem JIT bylo snížení časových úseků při komunikaci produkce a zákazníků. Pro tuto práci je zásadní uvědomit si roli jakou hraje Kanban při vývoji softwaru. Metodologie lze využít při postupném vývoji, při změně parametrů projektu a nabízí tak interaktivní řešení vývoje softwaru. Metodika Kanban může být ale použita i v jiných odvětví než pouze v automobilovém průmyslu. Velké světové společnosti následně přišly s různými alternativami pro tento produkční software. V dnešní době existuje i v České republice mnoho společností zabývajících se tzv. „Lean manufacturing“ neboli implementací a tvorbou produktů vysoké kvality s co nejnižšími náklady a za co nejkratší dobu [3]. Právě JIT je jeden z pilířů „Lean manufacturing“ druhým je tzv. „Jidoka“, což je růst kvality a design procesů nastavených tak aby lidé nebyli upoutaní u strojů a mohli tvořit přidanou hodnotu při zpracovávání více úkolů záraz [4].

Kanban staví na dvou důležitých aspektech, jimiž je neustálé přiřazování práce a doplňování „pracovního materiálu.“ Aby bylo vůbec možné mluvit o Kanbanu a stavět na této metodice celou tuto práci je nezbytné jej definovat. Cimorelli jej ve své knize definuje jako „method of just in time production that uses standard containers or lot sizes with single card attached to each“ [4] Tato definice ukazuje výrobní aspekt kanbanu a jeho tzv. „pull effect“, kdy materiál je tažen celým procesem pomocí synchronizovaného toku práce [4].

Pro vývoj softwaru za použití metodologie Kanban je třeba si na počátku stanovit cíl případně funkce samotného projektu. Až v návaznosti na stanovení tohoto cíle je dále možné rozvíjet požadavky, které k němu vedou. V této fázi hraje velkou roli komplexnost informací k jednotlivým požadavkům. Aby vše fungovalo, jak má je třeba spojit jak vědomosti osob, které plní dané úkoly, tak požadavky zákazníků a konečně i informace z třetích stran. Za pomoci této kombinace je pak možné stanovit priority pro jednotlivé fáze vývoje softwaru.

Existuje mnoho aplikací, které využívají metodu Kanbanu, případně z ní vycházejí. V poslední době např. nabývá na popularitě aplikace Trello nebo Wekan, Tato práce se věnuje aplikaci Kanboard.org o které pojednává následující kapitola.

2.2 Kanban Board

V této kapitole bude popsáno, co to Kanboard je a jak funguje. Následně zde bude popsána možnost rozšíření Kanboardu o moduly.

Kanboard nebo také Kanban Board je volně přístupný software. Kanboard se zaměřuje na jednoduchost a minimalismus dle toho také vypadá jeho uživatelské rozhraní [5]. Kanboard je vizualizací pomocí níž můžeme zjistit v jaké fázi se projekt nachází. Kanboard si zakládá na jednoduchosti a je velmi uživatelsky přívětivý, pro jeho pochopení není třeba žádných speciálních školení [5]. Nabízí také velké množství jazykových verzí a jako je třeba i čeština. Vzhledem k jeho jednoduchosti je třeba uzpůsobit jej potřebám uživatelů, pro které může být dosavadní prostředí nevyhovující. Právě proto, je zde možnost rozšíření původní aplikace pomocí modulů. Zde má uživatel k dispozici dvě varianty. Pokud žádný z nabízených modulů nevyhovuje požadavkům, není nic jednoduššího než si napsat vlastní modul, který si uživatel vytvoří podle svých potřeb, anebo si uživatel vybere některý z modulů nabízených přímo Kanboard.org, které jsou obvykle vyvíjené komunitou a jsou k dispozici zdarma. Zde je na místě upozornit na to, že moduly neprochází žádnou kontrolou a schvalovacími procesy Co se týče funkcionalit, tak Kanboard.org nepoužívá online knihovny a je tak příjemnou změnou, že je možné jej používat i bez přístupu k internetu.

2.2.1 Instalace

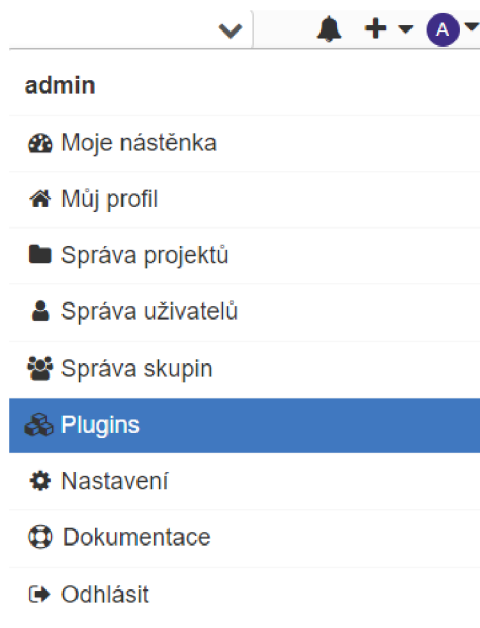
Instalace Kanboard

Aplikace Kanboard je napsaná v jazyce PHP, proto je potřeba mít tento jazyk na serveru nainstalovaný v minimální verzi 5.6.0, nicméně je doporučovaná alespoň verze 7.0. Serverová část Kanboardu podporuje jak některé Linuxové a Unixové operační systémy, tak i serverový operační systém Microsoft Windows 2012 R2 a Microsoft Windows 2016. Musíme mít nainstalovaný také webový server. Z databáze můžeme využít Sqlite, který je doporučovaný pro práci na menších projektech v malých týmech, nebo Mysql či Postgres pro práci na větších projektech. Uživateli poté stačí některý z webových prohlížečů, jako je Safari, Google Chrome, Mozilla Firefox Microsoft Internet Explorer verze 11 a vyšší a Microsoft Edge.

Instalace je velmi jednoduchá. Na nainstalovaný webserver stačí nakopírovat složku s kanboard.

Instalace modulů do aplikace Kanboard

Instalace modulů do aplikace je velmi jednoduché. Stačí modul stáhnout z webu, vytvořit nebo stáhnout modul z gitu. Složku s modulem přesuneme do složky plugins a o zbytek se postará aplikace. Následně se modul zobrazí v přehledu nainstalovaných modulů v nastavení Kanboardu viz *Obrázek 1*. Je doporučeno si číst popis modulu, ve kterém se může nacházet minimální požadovaná verze Kanboardu.

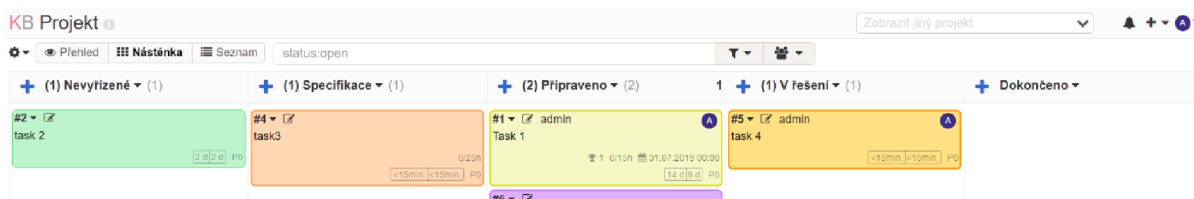


Obrázek 1 Menu pro správu aplikace Kanboard

2.2.2 Použití

Při prvním spuštění aplikace uvidíte přihlašovací obrazovku. V tuto chvíli existuje pouze jeden uživatel. Tento uživatel má stejné přihlašovací jméno i heslo, a to je **admin**. Další uživatele si můžete vytvořit v pravém horním rohu viz *Obrázek 1*. Uživatele poté můžete rozředit do týmů, a to vytvořením nového týmu a přiřazením jednotlivých uživatelů. K projektům poté můžeme přiřadit takto vytvořené týmy. Přiřazovat do projektu můžeme i jednotlivé uživatele

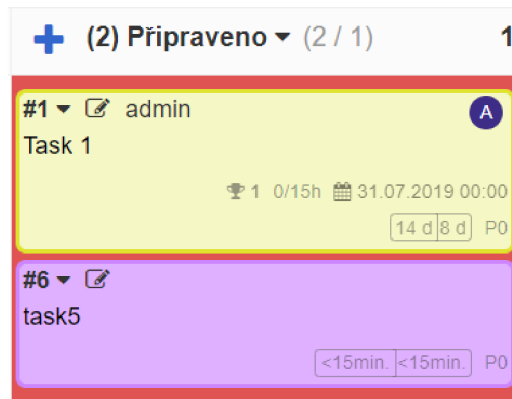
Nyní můžeme přejít na projekty, které jsou dvojího druhu. Buďto běžný projekt nebo soukromý projekt. Se soukromým projektem může manipulovat pouze vlastník a nelze do něj přiřazovat jiné



Obrázek 2 Příklad nástěnky projektu

uživatelé. Po vytvoření projektu mu můžeme nastavit různé parametry, jako jsou kategorie, štítky, sloupce nebo další horizontální linie. Nové úkoly můžete přidávat pomocí plus u každého sloupce horizontální linie. Na obrázku níže můžete vidět příklad nástěnky projektu.

Výše uvedený projekt má čtyři etapy vývoje, kde každý je reprezentován jedním sloupce. V hlavičce tabulky je uvedeno jméno etapy a počet položek. Každému sloupci také můžeme nastavit limit pro maximální počet položek. Když bude tento limit překročen zbarví se červeně pozadí



Obrázek 3 Překročení limitu úkolů

viz *Obrázek 3*. Úkoly mezi sloupce lze přesouvat pomocí myši tzv. „drag and drop“.

Kapitola 3

Využívané technologie

Logika modulu stejně jako logika aplikace Kanboard je napsaná v jazyce PHP. Modul je psaný ve verzi 7.0.32. Grafické rozhraní využívá HTML a CSS.

3.1 PHP

PHP je skriptovací, netypovaný, objektivě orientovaný programovací jazyk, který se především používá k vytváření dynamických internetových stránek. Veškeré výpočty jsou prováděny na straně serveru a v současné době je to nejpoužívanější skriptovací jazyk na straně serveru. PHP je poměrně starý programovací jazyk, byl vytvořen v roce 1995. To přináší řadu výhod, nicméně i nevýhod. Mezi výhody patří:

- rozsáhlá a přehledná dokumentace
- vysoká podpora u webhostingů
- množství funkcí ve standardní knihovně

Jako nevýhody se uvádí¹:

- nekonzistentní logika pořadí parametrů funkcí
- nekonzistentní pojmenování funkcí
- `array()` není objekt

Aplikace Kanboard je napsaná v jazyce PHP. Je napsaná bez použití rámce jazyka, nicméně využívá koncept MVC popisovaný níže.

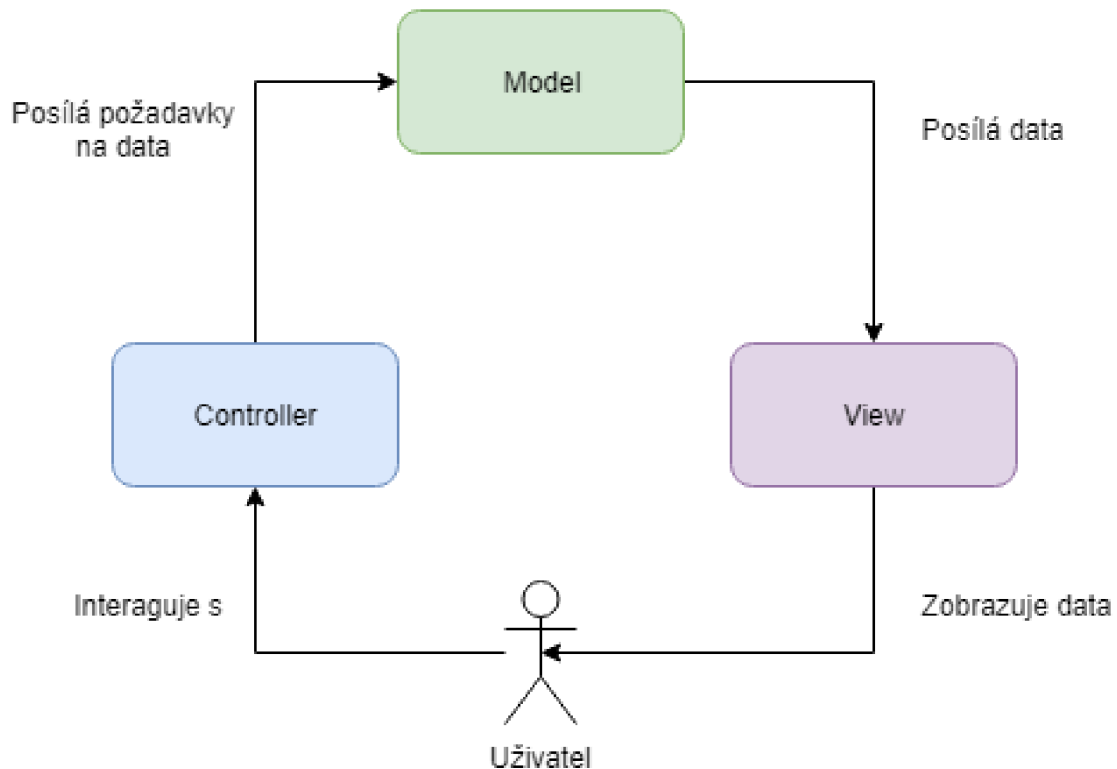
3.1.1 Model – View – Controller

Model – View – Controller se často označuje za návrhový vzor, nicméně je softwarová architektura nejčastěji používaná na vývoj webových aplikací. Rozděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé komponenty má pouze minimální vliv na ostatní komponenty. Těmi komponentami jsou:

- Model – definuje data aplikace a jejich chování, ale není zodpovědný za prezentování nebo formátování těchto dat.

¹ https://www.zizka.ch/pages/programming/php/nevyhody_php_v_praktickyh_prikladech.html

- View – zobrazuje data, která jsme dostali od modelu, uživateli. Nejčastěji se jedná phtml šablonu, obsahující HTML stránku a tagy nějakého značkovacího jazyku, který umožňuje provádět iterace a podmínky nebo vkládat proměnné.
- Controller – tzv. „prostředník“, který komunikuje s uživatelem, modelem, i view a propojuje všechny komponenty.



Obrázek 4 Princip architektury MVC

View komponenta zobrazuje data uživateli, mezitím akce uživatele zpracovává komponenta *controller*. *Controller* následně posílá požadavky na data (uložení, smazání, úprava nebo zobrazení). Komponenta *model* následně požadavky zpracuje a data vrátí.

3.2 Databázové technologie

Aplikace Kanboard umožňuje pracovat se třemi druhy databáze:

- SQLite
- MySQL
- Postgres

K práci na menších projektech stačí pracovat s SQLite, ale při práci na větších se doporučuje použít MySQL nebo Postgres².

² https://docs.kanboard.org/en/latest/admin_guide/requirements.html

3.2.1 SQLite

Oproti ostatním SQL databázím je SQLite pouze malá knihovna, která po přilinkování k aplikaci, je k dispozici pomocí jednoduchého rozhraní. Nepotřebuje tedy vlastní proces na straně serveru (přistupuje k uloženým souborům přímo). Jelikož je databáze uložena v jednom souboru, je velice snadno přenositelná. Kanboard tento databázový systém doporučuje na menší projekty s menším objemem dat. SQLite je v současnosti nejpoužívanější databázový systém na světě³.

3.2.2 MySQL

Je to relační databáze, která běží na straně serveru. Využívá více vláken současně, což vede ke zvýšení rychlosti databáze⁴. Každému připojení je přiřazeno nové vlákno. MySQL umožňuje i správu uživatelů. Autentizace uživatelů je prováděna buď pomocí uživatelského jména, hesla a hostitele nebo pomocí certifikátu přes SSL připojení. Kanboard doporučuje u větších projektů využít tento typ databáze nebo Postgres oproti SQLite, protože MySQL je navržena, aby zvládla větší databáze.

Data jsou oproti SQLite uložena na disku na serveru. Každá databáze má svou složku, ve které každý soubor reprezentuje jednu tabulku. Z důvodu uložení souborů na serveru je dobré dbát na pojmenování tabulek a databází. Na unixových serverech záleží na velkých a malých písmenech, ale na serverech s operačním systémem Windows na velikosti nezáleží.

Pro správu dat databáze a databáze můžeme využít mnoho nástrojů. Příklady nástrojů, které jsou uloženy na serveru jsou phpMyAdmin nebo Adminer.

3.2.3 Postgres

Poslední databázový systém, který můžeme v Kanboardu používat je Postgres. Postgres je open-source objektově-relační databázový systém dostupný pod licencí MIT. Umožňuje jako většina databázových systémů vytvářet funkce, indexy, triggerů atd. Tabulky mohou být vytvořeny děděním vlastností z rodičovské tabulky. Data uložená v takto vytvořených tabulkách se tváří, že jsou uložena v rodičovské tabulce

³ <https://sqlite.org/mostdeployed.html>

⁴ <https://dev.mysql.com/doc/refman/8.0/en/features.html>

3.3 CSS

Kaskádové styly je jazyk, který popisuje způsob zobrazení elementů na stránkách napsaných v jazycích HTML, XHTML nebo XML. Zpřehledňuje HTML kód, protože nemusíme styl psát „inline“ tj. přímo do atributu HTML značky. Pokud chceme nějaké značce nastavit styl můžeme k tomu využít buď přímo název značky nebo její třídu či id. Když nějaký element nacházející se v dokumentu nemá definovaný styl, dědí ho většinou od svých rodičovských elementů. Pokud tedy chceme definovat aby stránka využívala jeden font o jedné velikosti a jedné barvě, nadefinujeme styl nějakému hlavnímu elementu, nejčastěji je to `<body>`.

Kapitola 4

Milníkový modul

Účelem milníku je většinou kontrola postupu projektu nebo převzetí dílčí části, či hotového produktu. Při využití milníku jako kontrola postupu je součástí i nějaký druh rozhodnutí, jako třeba zastavení, pozastavení nebo pokračování v projektu. Právě těchto milníků se zabývám v této práci a tento modul by měl pomoci projektovému manažerovi mít větší kontrolu nad postupem a vývojem produktu.

4.1 Návrh

Na začátku bylo potřeba se zamyslet, jak tento modul přidat do již existující aplikace, aby bylo dodržoval nastavený styl uživatelského rozhraní aplikace. Bylo důležité, aby byl modul přehledný, ale zároveň jednoduchý a nezahlcoval vedoucího týmu zbytečnou náročností.

4.1.1 Návrh databáze

Při návrhu databáze bylo potřeba si uvědomit, jaká data bude vedoucí týmu, či projektový manažer potřeba u milníků uchovávat. Po nastudování databáze Kanboardu a nastudování problematiky jsem došel k prvnímu návrhu databáze viz. *Obrázek 5*. Z důvodu, že tabulka *projects* má 20 sloupců a tabulka *tasks* 32 sloupců, a že to není nová tabulka uvádím u nich pouze vybrané sloupce

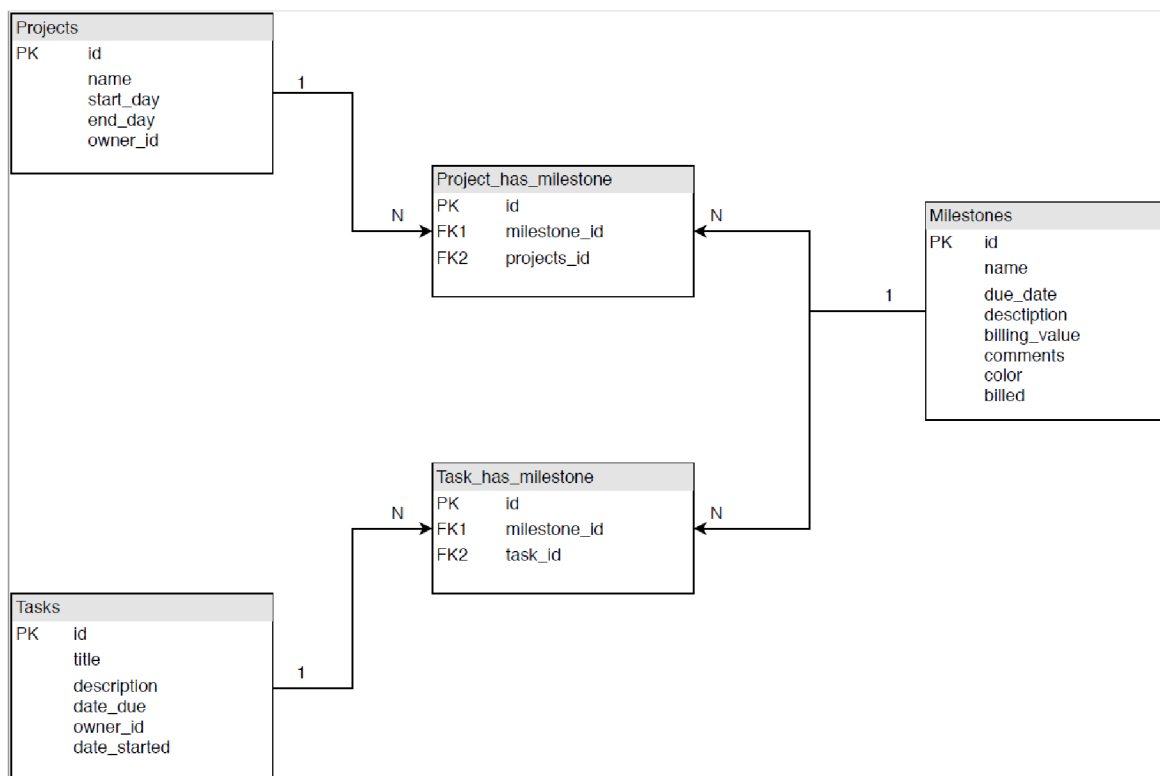
V tomto návrhu jsem počítal s tím že milník bude moct být přiřazen jak k více úkolům, tak i k více projektům. Modul bude mít tři tabulky.

První tabulka se jmenuje *Milestones* se sloupci:

- *id* – primární klíč
- *name* – jméno milníku
- *due_date* – datum, kdy milník nastává
- *description* – podrobný popis milníku
- *billing_value* – který nabývá hodnoty *NULL* pokud bude milník hodinový nebo částka, která byla dosažena na ukončení milníku
- *comments* – komentář

- *color* – barva, kterou bude mít pozadí milníku na výpise
- *billed* – nabývá hodnoty 1 nebo 0, podle toho zda byl milník již fakturován nebo ne

Tabulky *project_has_milestone* a *task_has_milestone* jsou spojivé tabulky pro propojení milníku s úkoly a projekty.



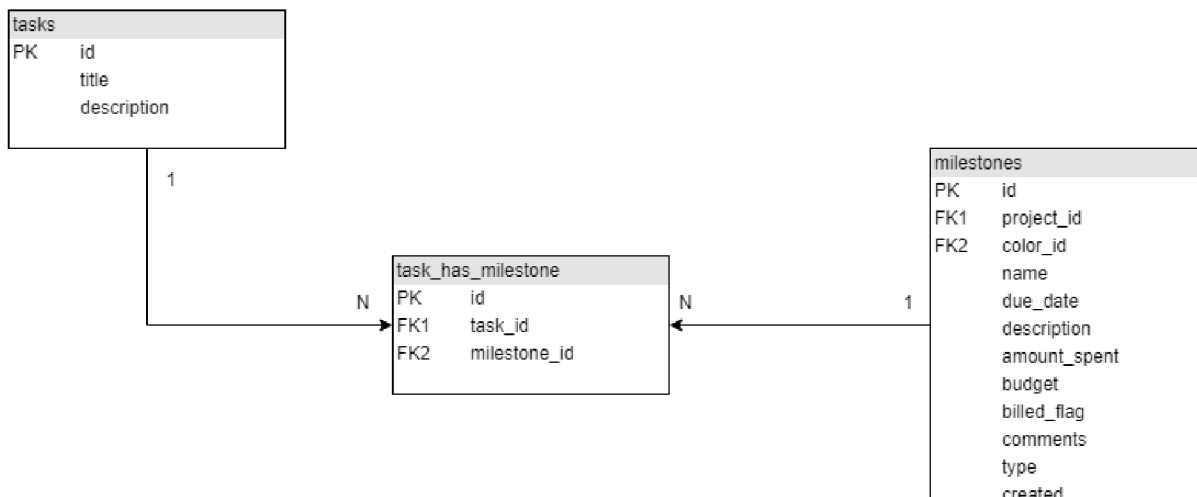
Obrázek 5 Původní návrh databáze

Po implementaci a otestování se ukázalo, že mít možnost přiřazení milníku k více projektům je špatné řešení. Toto řešení je nepřehledné pro uživatele. Další problém byl z rozpoznáním typu milníku. Z těchto důvodů vznikl nový návrh. Ten se skládá pouze ze dvou tabulek viz *Obrázek 6*. Hlavní tabulka *milestones* s těmito sloupci:

- *id* – primární klíč, INTEGER
- *name* – jméno milníku, TEXT
- *due_date* – datum, kdy milník nastává, INTEGER – počet sekund od 1. 1. 1970
- *description* – popis milníku, TEXT
- *budget* – rozpočet milníku, REAL
- *amount_spent* – hodnota utracená, REAL
- *color_id* – barva milníku, INTEGER
- *billed_flag* – hodnota, zda je milník vyfakturován, či ne, INTEGER
- *comments* – komentář milníku, TEXT
- *project_id* – id projektu na který je milník navázán, INTEGER

- *type* – typ milníku, TEXT
- *created* – datum vytvoření milníku, TEXT

Tabulka *task_has_milestone* je pomocná tabulka pro vazbu N:N a slouží pro navázání úkolů na milník, kde *task_id* a *milestone_id* jsou cizí klíče a pokud tedy bude smazán celý milník, smažou se i záznamy v této tabulce, které jsou na něj navázané.



Obrázek 6 Implementovaný návrh databáze

Kapitola 5

Implementace modulu Milestones

V následující kapitole se budu zajímat o implementaci modulu. Pro jeho vytvoření byly použity jazyky PHP, HTML, CSS. Zároveň byla dodržena pravidla, které platí pro celý Kanboard a to z důvodu zajištění kompatibility modulu se samotnou aplikací Kanboard

5.1 Způsob vytváření modulů

Jedna ze základních částí tvorby je nastudování vytváření modulu a následné napojení do aplikace. Toto řeší kapitola s názvem „Plugin development“ v dokumentaci aplikace Kanboard ⁵. Je nutno dodržovat principy, které jsou v dokumentaci aplikace, a to z důvodu správného napojení do Kanboardu.

5.1.1 Adresářová struktura modulu

Adresářová struktura modulu pro Kanboard je pevně daná viz **Obrázek 7** **Chyba! Nenalezen zdroj odkazů.** Není nic jednoduššího než tuto strukturu vytvořit pomocí nástroje `cookiecutter`. Tento nástroj můžeme nainstalovat pomocí tohoto příkazu:

```
pip install -U cookiecutter
```

Po instalaci nástroj spustíme příkazem:

```
cookiecutter gh:kanboard/cookiecutter-plugin
```

⁵ <https://docs.kanboard.org/en/latest/plugins/index.html>

```

plugins
├── JménoModulu
│   ├── Asset
│   ├── Controller
│   ├── LICENSE
│   ├── Locale
│   │   └── cs_CZ
│   ├── Model
│   ├── Plugin.php
│   ├── README.md
│   ├── Schema
│   ├── Template
│   └── Test

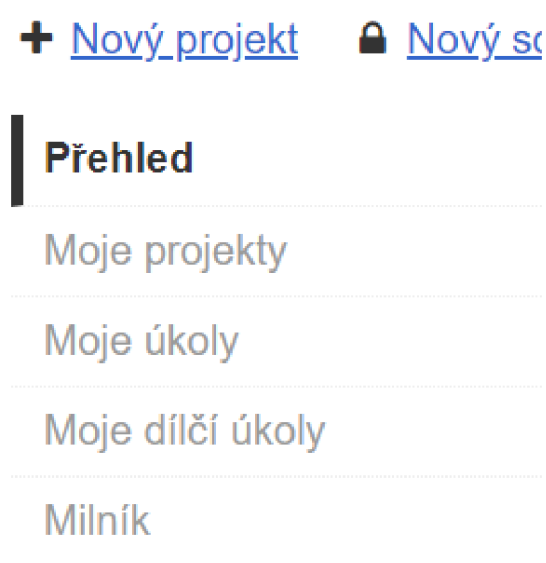
```

Obrázek 7 Adresářová strukturu modulu

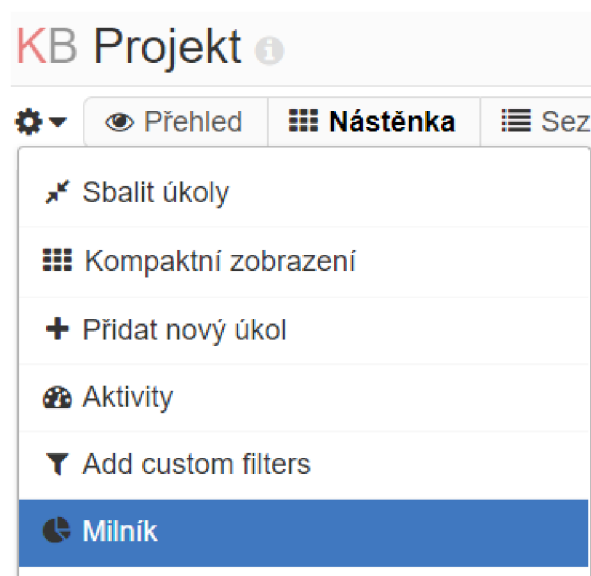
Nástrojem budete vyzváni na vyplnění jména modulu, jméno autora a další údaje, které popisují modul. Všechny údaje vyplňovat nemusíte, pokud je vyplňovat nechcete nebo tyto informace neznáte, dají se totiž vyplnit později. Po vygenerování adresářové struktury, musíte složku přesunout do složky plugins v aplikaci.

5.1.2 Registrace modulu

Aby mohl být modul v aplikaci použit, musí být zaregistrován. Třída `Plugin` dědí ze třídy `Base` se stará o tuto registraci. Třidu nalezneme v souboru `Plugin.php`, který se nachází v kořenovém adresáři modulu. Třída má mnoho metod, ale důležité jsou pouze tři. Hlavní metodou je



Obrázek 8 Menu na hlavní stránce



Obrázek 9 Menu projektu

`Initialize()`. V této metodě je zařízeno vkládání obsahu šablon, jako vložení odkazu do menu hlavní stránky aplikace a vložení odkazu v menu projektu, viz **Obrázek 8** respektive **Obrázek 9**.

Vkládají se zde také případné scripty a css soubory, které jsou využívány v modulu a z důvodu bezpečnosti jdou vkládat pouze tímto způsobem. Dále zde můžeme přidávat vlastní cesty nebo definovat přístupové práva uživatelů. Druhou důležitou metodou je metoda `getClasses()`. Tato metoda registruje třídy do Dependency Injection kontejneru, tím se zajistí, že se vytvoří pouze jedna instance třídy a může být využívána v celém programu. Poslední méně důležitou metodou je metoda `onStartup()`. Tato metoda načítá jazykové mutace.

5.1.3 Jazykové mutace

V současné době vyvíjený modul podporuje pouze angličtinu a češtinu. Pro novou jazykovou mutaci vytvoříme složku a pojmenujeme jí podle zkratky daného jazyka (`cs_CZ` pro češtinu). V této složce vytvoříme soubor `translations.php`, který obsahuje asociativní pole `klíč=>hodnota`. Překlad vyvoláme pomocí funkce `t($key)`, kde parametrem je klíč asociativního pole lokalizačního souboru. Po vytvoření daného souboru, přesuneme celou složku s překladem do složky `Locale`, která se nachází v kořenovém adresáři modulu.

5.1.4 Databáze

Pro správnou funkci modulu je potřeba vytvořit dvě tabulky viz. **Obrázek 6**. Jedna je pro ukládání milníků a druhá ukládá úkoly přiřazené k milníku. Tabulka milníků je propojená pouze s naší druhou tabulkou a druhá tabulka je propojená s tabulkou milníků a úkolů. Vytváření nových tabulek v Kanboardu snadné. Do složky `Schema` umístíme soubor, jehož název bude stejný jako název používaného databázového systému. Kanboard potom tento soubor najde a provede SQL příkazy uložené v souboru.

5.2 Implementace

5.2.1 Registrace modulu

Po vytvoření návrhu, byla další část vytváření modulu implementace. Zprv bylo na řadě modul správně zaregistrovat. Následně jsem postupně v metodě `initialize()` přidal práva pro zobrazení modulu uživatelům. Přidal vlastní cesty pro:

- zobrazení výpisů milníků, které náleží projektu
- zobrazení výpisu milníku na hlavní stránce v časové linii

- změnu stavu vyfakturování milníku

Cesty jsou přidávány zavoláním následující metody `addRoute()` třídy `Route`. Zde je uveden příklad pro přidání cesty pro zobrazení výpisů milníků, které náleží projektu:

```
$this->route->addRoute('/milestone/project/:project_id',
'MilestoneController','show','milestones');
```

První parametr je url, kde dvojtečka značí dynamický parametr, druhý parametr je název controlleru, třetí metoda controlleru a čtvrtý parametr je modul v kterém se controller nachází.

Poté bylo přidáno rozšíření současných šablon pro zobrazení milníku jako položky v menu a to jak v menu nastavení projektu, tak na dashboardu. Přidal jsem také vlastní css soubor.

V metodě `onStartup()` jsem zaregistroval českou jazykovou mutaci a v `getClasses()` jsem zaregistroval modely nového modulu a to `model Milestone` a `TaskHasMilestone`.

5.2.2 Zobrazení obsahu

O výpis obsahu se v aplikaci Kanboard starají controllery. V našem případě to je soubor `MilestoneController.php`. Jejich hlavní úkolem je, že při akci uživatele získá všechna vyžadovaná data a předá je šabloně. Šablona se následně postará o celkový vzhled stránky. Třída, která slouží jako controller se jmenuje `MilestoneController` a dědí z třídy `BaseController`. To zajistí správné napojení modulu do aplikace a zároveň je controlleru modulu poskytnuto mnoho užitečných základních funkcí. Pro přesné definování obsahu, který chceme dostat z určité metody controlleru, je potřeba do url přidat tyto tři GET parametry:

- controller – jméno třídy
- action – jméno metody, kterou chceme zavolat
- plugin – jméno našeho modulu

Abychom tyto parametry do url nemuseli sami zadávat ručně, použijeme třídu `url`. Tato třída obsahuje spoustu užitečných metod pro práci s odkazy.

Metody controlleru můžeme rozdělit do dvou částí a to:

- pro seznam milníků
- pro kompletní přehled milníků, projektů a úkolů

Sekce seznamu milníků je využívána při výpisu dat v detailu projektu a druhá část je pro vypisování dat na dashboardu.

Seznam milníků

Do této sekce se dostaneme po kliknutí na tlačítko *Milník* v nastavení projektu viz. *Obrázek 9*. Uživateli se zobrazí seznam milníků pro daný projekt a to invokací metody `show()`. K získání těchto milníků je využita metoda `getMilestoneByProjectId($project_id)`, následně na každý milník zavolána metoda `getTasksIdByMilestoneId($milestone_id)`, abychom dostali všechny úkoly, které jsou navázány na tento milník. Následně získaná data pošleme do šablony.

Další metody v této sekci jsou:

- `create()` – získání obsahu pro vytvoření nového milníku
- `save()` – získání dat a následné vytvoření nového milníku
- `edit()` – získání obsahu pro aktualizaci již existujícího milníku
- `update()` – získání dat pro aktualizaci již vytvořeného milníku
- `billed` – negace hodnoty „vlajky“ fakturace

Kompletní přehled

Do sekce dashboard se uživatel dostane z úvodní stránky po kliknutí v menu na tlačítko *Milník* viz. *Obrázek 8*. Zde je nejdůležitější metoda `dashboard()`. Tato metoda se stará o získání obsahu pro celkový přehled milníků, úkolů a projektů. Na začátku si načte GET atributy, pomocí metody implementované v aplikaci Kanboard, jimiž jsou:

- `date_from` – hodnota filtru, datum od jakého data chceme vypisovat časovou osu, pokud není nastavena, nahrazují dnešním datem
- `project_name` – hodnota filtru, slouží pro vyhledání určitého projektu nebo pro vyhledání projektu, v jehož jméně se vyskytuje hledaný výraz
- `page` – stránka na které se nacházíme v přehledu

Dále si tyto hodnoty upravíme z důvodu, že se v přehledu může uživatel pohybovat pomocí navigačních šipek a vygenerujeme si odkazy na další a předchozí stránku. Nyní nastal čas na získání projektů. Pokud je vyplněn atribut `project_name`, využijeme pro to metodu `getProjectForDashboard($project_name)`. Pokud tento atribut vyplněn není použijí metodu `getAll()` z modelu `ProjectModel`, který je již implementován v aplikaci. Poté ke každému projektu přiřadím úkoly a milníky pomocí metod `getTaskForDashboard($project_id,` `$date_from)` respektive

`getMilestoneForDashboard($project_id, $date_from)`. Následně jsou ještě pomocí metody `getTasksIdByMilestoneId($milestone_id)` získány úkoly, které jsou přiřazené k milníku. Tyto hodnoty jsou potom přiřazeny ke sloupci (datumu), ve kterém mají být zobrazeny. Z této obrazovky může být také vyvolána metoda `detail()`, která získává obsah pro zobrazení detailních informací o milníku.

5.2.3 Šablony

V šablonách je definován kompletní vzhled modulu. Všechny šablony se nachází v adresáři `Template`. Z důvodu přehlednosti jsem vytvořil ještě podadresáře. Jeden podadresář je pro část s kompletním přehledem, ten se jmenuje `dashboard`. Druhý podadresář je pro seznam projektů a jmenuje se `milestone`. Jako cesta k šabloně, která je vyžadována v některých funkcích, se uvádí název modulu, podadresář a název souboru šablony. Například:

```
'milestones:dashboard/milestone_detail'
```

Hlavní šablonou v podadresáři `milestone` je šablona `show.php`, která zobrazuje přehled všech milníků, které jsou přiřazené k projektu. V horní části kontejneru zobrazuje odkaz pro přidání nového milníku. Po kliknutí na něj se otevře „modal“, který zobrazuje šablona `create.php`, jedná se pouze o formulář. Ale zpět k hlavní šabloně. Největší část zabírá tabulka s milníky, kde hlavní obsah je zobrazován pomocí cyklu. Dalšími šablonami, kteří se nachází v tomto podadresáři, jsou `update.php`. Tato šablona je velmi podobná šabloně `create.php`. Zobrazuje formulář pro aktualizaci milníku. Poté tu je také `sidebar.php`, který rozšiřuje menu nastavení projektu o položku milníku viz. **Obrázek 9**. Hlavní šablonou druhého podadresáře je `show.php`. Tato šablona zobrazuje celkový přehled milníků, úkolů a projektů. Ve vrchní části kontejneru je zobrazen filtr. Jedná se o formulář, který je odesíláný *GET* metodou. Jsou tu také navigační šipky, které jsou vpravo nahoře. Jsou to pouze odkazy, které modifikují *GET* atribut *page*.

Hlavní část zabírá tabulka. V její hlavičce jsou zobrazovány datумы. Ty jsou zobrazovány pomocí cyklu `for()`. Hlavní data jsou zobrazována pomocí cyklu `foreach()`. Poprvé je zavolán na všechny projekty a vypíše název projektu a odkaz na něj do prvního sloupce. Následuje další cyklus `foreach()`, který iteruje přes data. Ta jsou milníky a úkoly. Abych zajistil, že úkoly se budou zobrazovat na levé straně buňky a milníky na pravé straně, je buňka jednoho dne reprezentována dvěma buňkami, které se v případě, že na tento den připadá pouze úkol, sloučí do jedné. Úkoly a milníky se zobrazují jako štítky. U milníků je zobrazována ještě ikona informace. Po kliknutí na ikonu se zobrazí „modal“ s detailními informacemi milníku. Tuto ikonu zobrazuji, aby uživatel nemusel stále překlikávat na přehled milníků a zpět.

V podadresáři se ještě nachází šablona `milestone_detail.php`, která zobrazuje detailní informace milníku a šablona `sidebar.php`, která opět rozšiřuje menu. Tentokrát je to menu dashboardu.

5.2.4 Modely

Modul ukládá data do dvou tabulek. Do tabulky `milestones` a do tabulky `task_has_milestone`. Třídy, které se starají o práci s těmito tabulkami jsou `Milestone` respektive `TaskHasMilestone`. V těchto třídách jsou metody, které se starají o práci s daty v těchto tabulkách. Práci s daty je myšleno ukládání nových záznamů a jejich následná úprava a mazání. Je tím myšleno také vyhledávání záznamů podle kritérií. Tyto třídy jsou umístěny dle doporučení v dokumentaci Kanboard ve složce `Model`.

Vytváření a úprava záznamů v tabulce

Pokud vytváříme nový milník, je potřeba tento nový záznam uložit do tabulek. To je docela jednoduché, protože všechny data která potřebujeme uložit do tabulky jsou odeslány formulářem a modelu tyto data předává controller. Je pro to použita metoda `save()` třídy `Milestone`. Je pouze nutné v této metodě zavolat metodu pro ukládání z třídy `TaskHasMilestone` a to z toho důvodu, že milníku může být přiřazeno více úkolů. Toto přiřazení je uloženo v jiné tabulce, než samotný milník.

Úprava záznamu je velmi podobná jako ukládání. Akorát při ukládání úpravy milníku nejprve uložím samotný milník a následně smažu přidělené úkoly z tabulky `task_has_milestone` a následně je znovu vložím. Je to z toho důvodu, že není nutné mít po celou dobu stejná `id` toho přiřazení.

Výběr záznamů

Daleko zajímavější je výběr záznamů pro celkový přehled milníků. Tady je totiž nutné vybírat pouze ty záznamy, které jsou v časovém úseku, který chci zobrazit, případně pokud je používán filtr pro vyhledání určitého projektu. Pro vyhledání milníků a úkolů, které odpovídají časovému úseku jsou implementovány funkce `getTaskDate($task, $date)` a `getMilestoneDate($milestone, $date)`, kde prvním parametrem je pole aktivních úkolů nebo pole milníků, z kterých následně vyberu záznamy odpovídající časovému úseku. Druhým parametrem je datum, od kterého vyhledávám záznamy. Z důvodu, že zobrazuji vždy sedm dní, tak mi stačí pouze počáteční datum a nemusím znát konečné datum. To si dopočítám.

Toto řešení není optimální, protože z tabulky si vybírám zbytečně mnoho dat a některé záznamy pak pomocí PHP na straně serveru zahodím. Měl jsem zde obavy z důvodu rychlosti, nicméně při

testování to dopad nemělo. Bylo by to poznat asi až v ostrém provozu a v nějaké větší společnosti, která má mnoho desítky projektů a mnohem více úkolů a milníků.

Další problém byl jaké úkoly vybrat, protože povinný údaj, který musí být vyplněn při vytváření úkolu je pouze název. Vyřešil jsem to tak, že vybírám pouze aktivní úkoly, které mají nastavené počáteční datum a vybírám je do data splnění úkolu.

Kapitola 6

Ovládání

Modul byl navržen, aby dodržel základní myšlenku celé aplikace Kanboard, a to aby modul nezahlocoval uživatele velkým množstvím informací. Aby ovládání bylo přehledné, a aby mu rozuměl každý uživatel.

6.1 Seznam milníků

Na obrazovce s výpisem milníků, které náleží určitému projektu vidí uživatel tabulku se seznamem milníků s jednoduchým popisem, jako je:

- datum ukončení milníku
- jméno milníku
- typ milníku
- rozpočet milníku
- utracená částka / strávených hodin
- zda byl milník vyfakturován
- úkoly, které jsou k milníku přiřazeny
- akce které mohou být zavolány na milník, jako je aktualizace a detailnější informace, jsou reprezentovány ikonkou

Pokud není projektu žádný milník, je o tom uživatel informován vypsáním upozorňujícího textu na obrazovce. Nad tabulkou seznamu milníků je odkaz, na kterém můžeme vytvořit nový milník pro daný projekt. Náhled seznamu milníků můžeme vidět na **Obrázek 10**

6.2 Přehled milníků a projektů

Nejhlavnější částí modulu je obrazovka s celkovým přehledem milníků, úkolů a projektů. Na této obrazovce můžeme vidět, ke kterému dni a projektu je přiřazen jaký milník, na jakých úkolech se může pracovat.

KB Projekt > Milník

Zobrazit jiný projekt

Prehled Milník

status:open

Nový milník

Plánovaný termín	Milník	Typ	Rozpočet	Utraceno	Fakturováno	Úkoly	Akce
27.07.2019	test2	Hodinový	0,00	0,00	Ne	task 2,	
27.07.2019	test3	Hodinový	25,00	5,00	Ne	Task 1,task 2,	
31.07.2019	test1	Hodinový	0,00	0,00	Ne	task 2,Task 1,	
31.07.2019	test4	Hodinový	0,00	0,00	Ne		
02.08.2019	test5	Hodinový	25,00	0,00	Ne		

Obrázek 10 Seznam milníků

Tato data jsou zobrazována tabulkou, kde každý řádek reprezentuje projekt. Tabulka má osm sloupců. V první sloupci je název projektu s odkazem na jeho detail. Dalších sedm sloupců jsou dny. Zobrazuje se vždy týden. Ve vrchní části obrazovky je filtr viz. **Obrázek 12**. Filtrovat můžeme projekty, podle jejich názvu, a počáteční den týdne. Pokud není nastaven filtr na počáteční datum, tak se zobrazí celý týden, kde počáteční datum je dnešní datum. Mezi týdny může uživatel přepínat pomocí navigačních šipek (**Obrázek 12**), které jsou umístěné v pravém horním rohu. Každá buňka dne je rozdělena na dvě části.

Projekt	31.07.2019	
Projekt	Úkol: #4 task3	Milník: #1 test1
	Úkol: #6 task5	Milník: #4 test4

Obrázek 11 Přiřazené milníky a úkoly k projektu

V první (levé) části se zobrazují úkoly. Úkoly se zobrazují jako štítky. Ve štítku se zobrazuje id úkolu a jméno. Při kliku na jméno bude uživatel přesměrován stránku s úkolem. Pokud je v daný den více projektů zobrazují se pod sebou, to můžeme vidět na **Obrázek 11**.

Datum od Jméno projektu

Projekt	31.07.2019	01.08.2019	02.08.2019	03.08.2019	04.08.2019	05.08.2019	06.08.2019

Obrázek 12 Filtr a navigace

V druhé (pravé) části jsou milníky. Milníky se zobrazují jako štítky, stejně jako úkoly. Ve štítku se zobrazují stejné hodnoty, jako ve štítku úkolu tj. id milníku a jméno. Poté tam je ještě ikonka informace. Po kliknutí na ní se zobrazí „modal“ s informacemi o milníku. V případě více milníku v jeden den se zobrazují také pod sebou, příklad můžeme vidět na **Obrázek 11**.

KB Milník Zobrazit jiný projekt

[+ Nový projekt](#)
[Nový soukromý projekt](#)
[Správa projektů](#)
[Přehled mých aktivit](#)

Přehled Datum od 31.07.2019 Jméno projektu Filtrovat

Projekt	31.07.2019	01.08.2019	02.08.2019	03.08.2019	04.08.2019	05.08.2019	06.08.2019
Projekt	Úkol: #4 task3 Milník: #1 test1 i	Úkol: #4 task3	Úkol: #4 task3 Milník: #5 test5 i	Úkol: #6 task5	Úkol: #6 task5	Úkol: #6 task5	Úkol: #6 task5
	Úkol: #5 task5	Milník: #4 test4 i Úkol: #6 task5	Úkol: #6 task5	Úkol: #5 task 4			
projekt 2	Úkol: #3 ukol projekt 2	Úkol: #3 ukol projekt 2	Úkol: #3 ukol projekt 2	Úkol: #3 ukol projekt 2	Úkol: #3 ukol projekt 2	Úkol: #3 ukol projekt 2	Úkol: #3 ukol projekt 2
projekt 7							
projekt3							
projekt4							
projekt5							
projekt6							

Obrázek 13 Přehled úkolů a milníků přiřazené k projektu

Kapitola 7

Testování

Testování je důležitou a v některých případech podceňovanou částí vývoje. Při každém přidání nové funkcionality se musí otestovat nejenom samotná nová funkcionality, ale i otestovat celek, jestli nedošlo ke změně původně funkčního celku. Je nutno tedy otestovat jak nový modul, tak i celou aplikaci Kanboard.

Toto testování probíhalo pokusem o nasimulování jeho používání v menší až střední společnosti. Byly vytvořeny testovací projekty. Dále jsem k nim přiřadil testovací milníky. Následně jsem vytvořil testovací milníky a přiřadil k nim úkoly. Vyzkoušena byla úprava milníků, projektů i úkolů. Testoval jsem také celkový přehled. Zde jsem upravoval datum splnění a začátek úkolů a sledoval, zda se tato změna správně propíše do celkového přehledu. Takto jsem upravoval i datum u milníků. Testováním také prošlo filtrování a změna výpisu dnů pomocí navigačních šipek.

Při testování jsem nezaznamenal problém s výkonem ani při větším počtu projektů, úkolů a milníků. Nicméně z důvodů simulace menší, či středně velké společnosti bylo pracováno s omezeným množstvím dat. Není tedy vyloučeno, že se problémy s výkonem nevyskytnou, protože větší společnosti pracují na mnohem více projektech a úkolech.

Kapitola 8

Možnost rozšíření

Tento modul by mohl být rozšířen o některé další funkce. V následujícím textu zkusím navrhnout některá z nich.

8.1 Hodinová mzda

Nyní musí správce zadávat utracený rozpočet ať už hodinový, tak i finanční, ručně. Toto rozšíření by přidalo možnost, přiřadit každému uživateli hodinovou mzdu. Pouze toto rozšíření by ale neodstranilo ruční zadávání útraty. Bylo by potřeba ještě zavést měření času pro každý úkol. Představoval by si to tak, že by v detailu úkolu přibyly stopky, které by uživatel, který na daném úkolu zrovna pracuje zapnul. Byla by ale i možnost ručního stráveného času jako dosud. Tento čas by pak šel přiřazovat k milníku. Buď by se přiřazoval strávený čas, nebo částka, která by se vypočítala z nastavené hodinové mzdy u uživatele.

Pro stopky by bylo nutné aby každý uživatel měl své, aby je mohlo mít spuštěno více uživatelů najednou. Případně by šlo ještě vytvořit časový rozpočet pro každého uživatele. Je tím myšleno že by se celkový odhadovaný čas, který bude potřeba na provedení úkolu, rozdělil mezi uživatele, kteří na úkolu pracují.

Kapitola 9

Závěr

Cílem práce bylo navrhnout a vytvořit modul pro sledování milníků, který rozšiřuje základní funkcionalitu aplikace Kanboard. Modul pracuje s databázovým systémem SQLite. Podařilo se také modul připravit na pro databázový systém MySQL a byl přeložen do českého jazyka, který podporuje i hlavní aplikace.

Tento cíl byl splněn. Seznámil jsem se s aplikací Kanboard.org a technologiemi SQLite a PHP. Byly nastudovány i další bodů zadání. Modul byl navržen a implementován. Byli implementována možnost vytváření milníků a jejich přiřazování k projektům a k úkolům. Byl vytvořen přehled, pro rychlé zjištění k jakému projektu milník patří a jaké úkoly jsou k němu přiřazeny. Snažil jsem se, aby modul dobře zapadnul do stávající aplikace a aby nezasahoval do stávající databáze, a to z důvodu možnosti případného odstranění modulu nebo možnému konfliktu s jinými moduly.

Největší problém asi byl se zjištěním, jak Kanboard funguje a jak mohu stávající aplikaci využít při implementaci modulu. Dokumentace totiž není tak rozsáhlá jako třeba dokumentace programovacího jazyka PHP. Proto bylo potřeba procházet kódem stávající aplikace Kanboard a zjišťovat, jak jsou třídy a jejich metody používány.

Modul bude určitě třeba upravit, protože testování bylo pouze manuální a simulovala se pouze menší společnost. Dále by byla potřeba zpětná vazba od vedoucích týmů, či projektových manažerů, kterým je tento modul převážně určen

Literatura

- [1] Krieg, G. N. (c2005). *Kanban-controlled manufacturing systems*. New York: Springer.

- [2] HUANG, C. H. U. N. -C. H. E., & KUSIAK, A. N. D. R. E. W. (1996). Overview of Kanban systems. In *INT. J. CO MPUTER INTEGRATED M ANUFACTURING* (vol. 9, no. 3, pp. 169-189). Taylor & Francis.

- [3] Hobbs, D. P. (c2004). *Lean manufacturing implementation: a complete execution manual for any size manufacturer*. Boca Raton, Fla.: APICS.

- [4] Cimorelli, S. (2017). *Kanban for the Supply Chain: Fundamental Practices for Manufacturing Management*(2nd Edition). CRC Press.

- [5] Barolli, L., & Xhafa, F.Hussain, O. (Ed.). *Innovative Mobile and Internet Services in Ubiquitous Computing: Proceedings of the 11th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. Switzerland.

Příloha A

Obsah CD

- readme.txt – soubor popisující instalaci modulu
- technicka-zprava.php – tato technická zpráva
- /src – složka se zdrojovými kódy modulu