

Sem vložte zadání Vaší práce.



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



Bakalářská práce

## **Softwarová aplikace pro řízení projektů s využitím gamifikačních prvků**

*Jan Slifka*

Vedoucí práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

10. května 2015



---

## Poděkování

Děkuji doc. RNDr. Ing. Marcelu Jiřinovi, Ph.D. za cenné rady a vedení mé práce. Rovněž bych chtěl poděkovat všem uživatelům, kteří se účastnili testování aplikace.



---

## 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 10. května 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Jan Slifka. 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**

Slifka, Jan. *Softwarová aplikace pro řízení projektů s využitím gamifikačních prvků*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.



---

## Abstrakt

Práce se zaměřuje na prozkoumání gamifikace a gamifikačních prvků a mechanismů. Úkolem je analyzovat současné využití gamifikace v praxi jako podpůrného nástroje pro motivaci lidí při plnění zadaných seriózních úkolů. Cílem práce je na základě analýzy navrhnout vlastní aplikaci zaměřenou na řízení projektů, zejména na zadávání úkolů, kontrolu pracovníků a jejich motivaci k práci.

**Klíčová slova** gamifikace, gamifikační prvky a mechanismy, řízení projektů, motivační aplikace

---

## Abstract

The thesis explores gamification and gamification elements and mechanisms. It analyzes current use of gamification as a supportive tool for motivation, especially when working on serious tasks. The goal is to design and implement a new application that focuses on project management, task management, workers control and their motivation to work.

**Keywords** gamification, gamification elements and mechanisms, project management, motivational application



---

# Obsah

Úvod	1
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Gamifikace a řízení projektů</b>	<b>5</b>
2.1 Gamifikace . . . . .	5
2.2 Projektové řízení . . . . .	8
2.3 Současná řešení . . . . .	10
<b>3 Koncept</b>	<b>13</b>
3.1 Základní myšlenka . . . . .	13
3.2 Uživatel/Hráč . . . . .	13
3.3 Postava . . . . .	14
3.4 Projekt . . . . .	15
3.5 Drak . . . . .	16
3.6 Práva . . . . .	18
3.7 Rozšíření do budoucna . . . . .	18
<b>4 Analýza a implementace</b>	<b>23</b>
4.1 Analýza . . . . .	23
4.2 Architektura . . . . .	31
4.3 Vybrané technologie . . . . .	31
4.4 Nasazení . . . . .	32
<b>5 Výsledky</b>	<b>35</b>
5.1 Testování . . . . .	35
5.2 Výhled do budoucna . . . . .	40
<b>Závěr</b>	<b>43</b>

<b>A Seznam použitých zkratek</b>	<b>45</b>
<b>B Obsah přiloženého CD</b>	<b>47</b>
<b>Literatura</b>	<b>49</b>

---

## Seznam obrázků

4.1	Případy užití – manažer a vlastník projektu . . . . .	26
4.2	Případy užití – běžný a pokročilý uživatel . . . . .	27
4.3	Diagram stavů úkolu . . . . .	28
4.4	Doménový model . . . . .	30
4.5	Diagram nasazení . . . . .	33
5.1	Jednoduchý registrační formulář . . . . .	36
5.2	Nastavení uživatelů na projektu . . . . .	36
5.3	Formulář pro vytvoření úkolu . . . . .	37
5.4	Vytváření nového sprintu . . . . .	38
5.5	Questlog aktivního sprintu . . . . .	38
5.6	Nastavení uživatelského profilu . . . . .	39
5.7	Změna hesla . . . . .	39



---

## Seznam tabulek

- 3.1 Tabulka celkového počtu zkušeností nutných ke získání další úrovně 14
- 3.2 Tabulka práv jednotlivých rolí na projektu . . . . . 18





---

# Úvod

Ve své práci se zaměřím na koncept gamifikace a jeho využití v oblasti projektového řízení. Analyzuji současná existující řešení, která gamifikaci a řízení projektů propojují.

Na základě analýzy gamifikace a projektového řízení navrhnu vlastní aplikaci, která bude umožňovat spravovat projekty a využívat při tom gamifikaci jako motivační faktor.

Poté vyberu vhodné softwarové nástroje a aplikaci pomocí nich implementuji.

Aplikaci otestuji na reálných uživatelích a případech užití a vyhodnotím úspěšnost konceptu a implementace. Bude-li to nutné, na základě výsledků testování koncept a implementaci upravím.



## Cíl práce

Gamifikace je v současné době moderním trendem. Má v první řadě motivovat lidi k nějaké činnosti. V průběhu řízení projektů je potřeba pracovníky motivovat, aby podávali co nejlepší výkony. Práce má za úkol prozkoumat trend gamifikace a zjistit jaké gamifikační prvky a mechanismy je možné použít. Zaměřuje se na to, jak je využít v kontextu řízení projektů.

Cílem je tedy navrhnout, pomocí vhodných nástrojů implementovat a otestovat aplikaci, která bude sloužit ke správě a řízení projektů a bude využívat gamifikace jako motivačního prvku.



---

# Gamifikace a řízení projektů

Práce se zaměřuje na využití gamifikace pro řízení projektů. Proto nejprve charakterizují pojem gamifikace a vysvětlím některé herní prvky a mechanismy. Dále se zaměřím na projektové řízení, zejména v oblasti softwarových projektů. Na závěr kapitoly popíšu existující řešení, která propojují gamifikaci s projektovým řízením.

## 2.1 Gamifikace

Přirozenou součástí lidské povahy je úsilí o úspěch. Ve chvíli, kdy lidé úspěchu dosáhnou, jejich úsilí nekončí, naopak chtějí další, větší úspěch. Této základní motivace využívá většina her. Před hráče jsou stavěny čím dál obtížnější výzvy a překážky a za jejich překonávání je hráč náležitě odměněn.

Gamifikace spočívá v rozšiřování tohoto konceptu do neherního prostředí (vzdělávání, businessu, marketingu aj.). Vytvoření krátkodobých, jasně vymezených úkolů, za jejichž plnění následuje nějaký způsob odměny, pomáhá motivovat k různým aktivitám. [1]

Gamifikace je tedy využívání herních prvků a mechanismů v neherním prostředí. Gamifikaci lze využívat v mnoha prostředích, kde je potřeba motivovat uživatele k nějaké činnosti. Jedna z oblastí je řízení projektů, kde je potřeba, aby pracovníci byli motivováni plnit zadané úkoly.

### 2.1.1 Herní prvky

Když hráči hrají nějaké hry, je důležité, aby chápali, o co ve hře jde a co mají dělat. Hra jim je nějakým způsobem prezentována jednak svou grafickou podobou, jednak pomocí herních prvků.

**Úrovně** Úrovně jsou běžnou součástí většiny her. Jedná se o systém, podle kterého jsou hráči odměňováni za získávání určitých bodů a získávají nové

úrovně. Může se jednat o nové oblasti, do kterých se hráč dostává, nebo o zlepšování postavy, za kterou hraje. Se získáváním úrovní často souvisí získávání nových schopností.

Použití úrovní v neherním prostředí může být různé. Například ve vzdělávání lze rozdělit řešení složitého problému na menší části a odměňovat studenty postupně za jejich řešení. Mohou se také posouvat řešením jednoduchých úkolů ke složitějším a tím odemykat nové úrovně. [2]

Často se ani nemusí jednat o zvyšování obtížnosti, jako spíš o různé celky z hlediska obsahu – různé oblasti vzdělávání, prozkoumávání částí webové aplikace apod. [1]

**Body** Body jsou jednoduchý a účinný herní prvek. Hráči ve hře sbírají body a mají nějaké skóre. Body je ale potřeba zkombinovat s dalšími herními prvky, jinak tolik motivující nejsou. Hráči mohou za body získávat odměny. Lze určit žebříček, kdy se při stanovených počtech bodů odemkne určitá odměna, nebo lze používat body jako herní peníze, za které si hráči nakupují odměny dle jejich výběru.

V případě gamifikace lze body rozdávat za různé aktivity, které hráč v gamifikovaném prostředí provádí. Může to být napsání článku na diskusním fóru nebo splnění úkolu v systému pro řízení úkolů. Důležité je vhodně nastavit, za co budou hráči body dostávat a stanovit hodnotu různých aktivit.

**Achievements** Achievements představují dosažení určitého úspěchu. Hráči za ně získávají odměny, velmi často v podobě různých odznaků nebo ikon. Achievements nepředstavují pravidelný proces zlepšování, jako je získávání nových úrovní, jedná se o splnění určitých jasně vymezených podmínek. [2]

Podmínky získání odznaku mohou být jednoduché i složité, získatelné samostatně nebo ve skupině hráčů, svázané s určitým ročním obdobím (např. odznak lze získat pouze na Vánoce) apod. V závislosti na tom, jak obtížné je odznak získat, může následovat i další, speciální odměna – hráč dostane titul, speciální předmět či schopnost pro svou postavu.

Dobré je, když hráč kromě odznaků, které získal, může vidět i ty, které zatím nemá (často bývají zobrazené šedě) spolu s podmínkami, které je potřeba splnit k odemknutí odznaku. Zobrazení nezískaných odznaků zvyšuje motivaci hráčů k tomu, aby se je snažili získat. [1]

**Ukazatele postupu** Pro hráče je důležité, aby jasně věděl, kolik ve hře něčeho získal (bodů, zkušeností apod.) a kolik mu chybí k dosažení většího úspěchu (další úroveň, achievement apod.). Pokud hráč vidí, kolik ještě potřebuje získat a ví jak, má větší motivaci cíle dosáhnout. Ukazatel postupu tedy jasně ukazuje, jak daleko hráč postoupil k cíli.

Ukazatel postupu využívá například LinkedIn k motivaci uživatelů, aby měli kompletní profil.

**Odměny** V každé hře hráč za hraní dostává v nějaké podobě odměnu. V gamifikovaném prostředí může být odměna za hraní něco v rámci hry samotné – nová věc, unikátní vzhled, odznak apod. Může to ale být i odměna v reálném světě, např. sleva na produkt.

Systém odměn musí být dobře vysvětlený a pro hráče pochopitelný. Pokud by o něm nevěděl nebo ho nechápal, nebude mít motivaci k tomu, aby se odměnu snažil získat.

**Kolekce** Hráče obvykle baví sbírat kolekce a chtějí je mít kompletní. Může se jednat o sběratelské karty z jedné edice, sada předmětů v RPG apod.

Klíčové k tomu, aby hráč začal kolekci chtít, je, aby snadno získal její první část. Často ji získají téměř hned potom, co začnou hrát. Získat další části je pak obvykle těžší. Avšak hráči, kteří už část kolekce mají, jsou mnohem více motivováni k dosbírání zbytku. Je také potřeba, aby hráči věděli, které části jim chybí a jak je mohou získat.

**Žebříčky** Žebříčky jsou jednoduchý, avšak účinný způsob gamifikace různých aktivit. Podporují soutěživost mezi hráči, aniž by spolu soupeřili přímo. Hráči jsou seřazeni v žebříčku podle různých bodů nebo skóre. Důležité je, zobrazovat žebříčky tak, aby hráči uvedení na vrcholu na tom nebyli výrazně lépe a nezpůsobovalo to naopak demotivaci, protože hráč ztratí naději, že by se na vrchol někdy dostal. [1]

Může být vhodné, zobrazovat pouze několik hráčů, kteří jsou přímo před hráčem a které má šanci předběhnout. Žebříček lze také rozdělit do několika částí, kde hráč vidí ostatní hráče, kteří jsou ve stejné části žebříčku jako on a všichni jsou na tom podobně. Na tomhle principu funguje žebříček ve hře League of Legends, kde jsou hráči rozděleni do lig a divizí. [9]

**Avatar** Avatar nebo jiná reprezentace hráče mu pomáhá, aby si uvědomil, že je součástí hry a jakou v ní hraje roli. Reprezentuje ho v různých žebříčcích, objevuje se u jeho příspěvků na fóru a podobně. Je to jeho součástí v herním prostředí.

### 2.1.2 Herní mechanismy

Herní mechanismy jsou principy, které využívají herní prvky k tomu, aby motivovaly hráče ke hraní hry nebo v případě gamifikovaného prostředí k používání aplikace.

**Soutěživost** Soutěživost mezi hráči patří mezi často používané herní mechanismy. Může být přímá, kdy hráči hrají přímo proti sobě, nebo nepřímá, kdy se hráči porovnávají podle herních bodů nebo umístění v žebříčku.

V oblasti gamifikace bývá častější použití právě nepřímého soutěžení. Hráči získávají body nebo jiné ohodnocení za různé aktivity a na základě toho jsou umístěni v žebříčku.

Lze toho využívat na internetových stránkách, které chtějí motivovat uživatele k psaní obsahu. Mohou být umístěni do žebříčku podle toho, jak moc přispívají, a vidět, jak přispívají ostatní a kolik jim chybí bodů, aby je dohnali.

Herní prvky vhodné pro soutěživost mezi hráči: žebříčky, achievementy.

**Týmová spolupráce** Další populární herní prvek je týmová spolupráce. Ve hře jsou stanoveny cíle, na jejichž splnění je potřeba více spoluhráčů. To může pomoci motivovat hráče, protože nikdo nebude chtít být ten, kdo to týmu skazí.

Herní prvky vhodné pro týmovou spolupráci: achievementy, odměny.

**Postup** V každé hře se objevuje určitý postup. Hráč potřebuje vidět, jak postupuje dál a otevírají se mu nové úrovně, vylepšuje svou postavu, má nové herní možnosti a podobně.

Důležité je, aby hráč rozuměl systému, ve kterém postupuje dál. Obvykle je získávání prvních úrovní jednoduché a hráč se při tom učí systému porozumět. Další úrovně už bývají složitější. Hráč musí chápat, jak postup funguje a vědět, co má udělat, aby se dostal dál.

Herní prvky vhodné pro postup: ukazatele postupu, úrovně.

**Vlastnictví** Pro lidi je běžné, že chtějí shromažďovat bohatství, mít stále víc. Stejně tak ve hře. Hráči chtějí získávat zkušenosti a věci pro svou postavu, budovat vesnici nebo vylepšovat cokoli jiného. Pokud to je v rámci hry jejich vlastnictví, chtějí ho rozšiřovat a bránit před ostatními hráči.

Herní prvky vhodné pro vlastnictví: odměny, kolekce, avatar.

## 2.2 Projektové řízení

Projektové řízení je jednorázový proces, který má za úkol rozplánování a realizaci projektu v požadovém čase a s naplánovanými náklady. Nedílnou součástí tohoto procesu je plánování a koordinace projektového týmu a jednotlivých činností nutných k dosažení cíle projektu. [12]

### 2.2.1 Softwarový proces

V oblasti vývoje software s projektovým řízením úzce souvisí softwarový proces, neboli množina aktivit potřebných k vývoji software. [13]

Softwarový proces lze rozdělit do dvou skupin – plánovaný softwarový proces a agilní softwarový proces.



**Plánovaný softwarový proces** Plánování všech aktivit probíhá na dlouhou dobu dopředu. Vše je naplánováno pevně a postup vývoje lze snadno kontrolovat oproti plánu. Měřítkem postupu je dodržování plánu. [13]

Má jasně oddělené fáze vývoje – sběr požadavků, analýza, návrh, implementace, testování, nasazení. [13] Ty následují jedna za druhou, začne se sběrem požadavků a skončí nasazením. Předem je jasný plán, tím pádem lze snadno predikovat, kdy bude co hotové. Je však nutné mít od začátku přesně definované požadavky. Tento přístup je však velmi nepružný v případě větších změn v průběhu vývoje, ty pak vyžadují velkou režii.

**Agilní softwarový proces** Plánování probíhá po menších částech, sprintech. Všechny fáze vývoje probíhají v každém sprintu. Je velmi flexibilní a snadno se přizpůsobuje změnám zadání v průběhu vývoje. Měřítkem postupu je fungující software. [13]

Cílem je mít na konci každého sprintu fungující produkt, který je možné předvést zákazníkovi. Úzká spolupráce se zákazníkem je v agilním vývoji klíčová. Je nutné, aby se právě zákazník podílel na plánování toho, co se v následujícím sprintu bude vyvíjet. Problém může být, že zákazník není schopen spolupracovat. [10]

V agilním vývoji se upřednostňují jednotlivci a interakce před procesy a nástroji. Lidé z byznysu i vývoje by měli spolupracovat po celou dobu každý den. Osobní komunikace je považována za nejefektivnější způsob sdělování informací v týmu. [10]

Vzhledem k tomu, že není dlouhodobý plán, je těžké na začátku odhadnout, kdy bude projekt hotový jako celek. Víme pouze to, co bude hotové na konci příštího sprintu.

### 2.2.2 Nástroje a prvky agilního vývoje

V práci se zaměřuji především na gamifikaci agilního vývoje. Proto je potřeba charakterizovat nástroje a prvky, které s agilním vývojem souvisí a které chci použít.

**User story** User story je popis nové funkcionality z pohledu koncového uživatele. Bývá ve tvaru:

*Jako <role>, chci <cíl>, abych <užitek>*

User story se pak rozpadá na části související s implementací – vytvoření formuláře, vytvoření tabulky v databázi apod.

**Backlog** Backlog je seznam úkolů, které je potřeba udělat. Úkoly jsou obvykle psané jako user story. V backlogu jsou seřazené podle priority tak, aby nejvýš byly ty s nejvyšší prioritou. Priorita se určuje na základě konzultace se zákazníkem podle toho, co on považuje za nejdůležitější.

**Sprint** Sprint vzniká na tzv. sprint planning meeting, kde se určí cíl sprintu a backlog sprintu. Z backlogu se vyberou úkoly s nejvyšší prioritou, které se realizují během sprintu. Množství úkolů by měl určovat primárně tým.

Sprint trvá obvykle 14 dní až měsíc. V průběhu si členové týmu vybírají úkoly z backlogu sprintu a pracují na nich.

Na konci sprintu tým prezentuje práci, kterou v průběhu sprintu zvládl. Také hodnotí jak funguje týmová spolupráce a vymýšlejí, jak ji zlepšit.

**Agile board** Agile board je tabule, na které lze sledovat průběh aktuálního sprintu. Skládá se z několika sloupců, jejichž přesná definice záleží na procesu týmu. Obvyklé jsou sloupce - „to do“, „in progress“, „to verify“, „done“. Používá se ke sledování práce v průběhu sprintu.

Členové týmu si vybírají úkoly ze sloupce „to do“, které chtějí dělat. Přesunou je do sloupce „in progress“. Tam je vidět, kdo aktuálně na čem pracuje. Po dokončení úkolu ho člen týmu přesune do části „to verify“ a ve chvíli, kdy je ověřeno jeho správně dokončení, je přesunut do sloupce „done“.

Agile board může obsahovat na začátku i backlog, ze kterého se úkoly vybírají do sprintu.

**Story point** Story point je abstraktní jednotka, která se používá k měření úsilí potřebného k dokončení user story. Na začátku projektu se určí nějaký základní user story, který si všichni dokážou představit a rozhodnou se, kolik story pointů představuje úsilí k jeho dokončení. Od této hodnoty se pak odvíjí story pointy ostatních tak, že se porovnává složitost příběhu se základním user story.

Nejčastěji se používají fibonnaciho čísla nebo mocniny dvou. Je to proto, že obtížné úkoly nelze určovat tak přesně. V podstatě nehraje roli jestli hodnota bude 64 nebo 65.

Důležité je, že story pointy nelze v žádném případě převádět na hodiny. Stejně tak nelze porovnávat obtížnost úkolů mezi různými týmy na základě story pointů, protože každý tým má jinak nastavenou hodnotu bodů.

## 2.3 Současná řešení

Existuje velká řada aplikací pro řízení a plánování projektů. Gamifikace je sice novým trendem, ale existují pluginy, které tyto aplikace rozšiřují o gamifikaci. Většinou se jedná o doplnění některých gamifikačních prvků, než o to, že by byl celý projekt pojat jako hra.

### 2.3.1 JIRA

JIRA[3] je rozsáhlá aplikace pro řízení a plánování projektů a sledování pracovních výkonů při plnění úkolů a nabízí k tomu rozsáhlou škálu nástrojů. Lze

nastavovat procesy pracovních postupů tak, jak tým potřebuje. Úkoly jsou plánované do sprintů a každý sprint je zobrazen v přehledné tabulce v závislosti na nastavených procesech. Podporuje týmovou spolupráci, je možné sledovat úkoly a jejich průběh, psát k nim komentáře apod. Existují pluginy, které aplikaci rozšiřují o gamifikaci.

**JIRA Hero**[4] Plugin rozšiřuje aplikaci o achievementy, uživatelé získávají body a nové úrovně, lze zobrazovat různé žebříčky hráčů. Pomocí achievementů také pomáhá lépe poznávat samotnou aplikaci (existují například achievementy za používání určitých klávesových zkratk).

**Jiraffe**[5] Jiraffe přidává projektu tematiku, proměňuje ho v pirátský příběh. Nabízí značnou volnost v nastavení gamifikace. Na každém projektu lze vytvořit určité uživatelské schopnosti (tester, developer, admin apod.) a k nim přiřazovat události, které se v systému dějí (vytvoření nové verze, splnění úkolu), a ohodnocovat je body. Uživatelé pak zlepšují svoje schopnosti, když s aplikací pracují. V různých tabulkách je pak vidět, kdo má kolik bodů, kdo se nejvíc zlepšil apod. Dále umožňuje vytvářet odznaky, které pomáhají motivovat uživatele. Nechybí ani statistiky s různými grafy.

### 2.3.2 Redmine

Redmine[6] je aplikace pro správu projektů. Nabízí nástroje pro vytváření a sledování úkolů a plánování projektů. Existuje gamifikační plugin[7], který má za úkol motivovat tým pomocí úrovní, odznaků a žebříčků.

### 2.3.3 RedCritic Tracker

RedCritic Tracker[8] je aplikace pro správu projektů zaměřená na agilní vývoj. Gamifikační prvky má integrované přímo v sobě bez nutnosti instalace doplňků. Uživatelé dostávají odznaky za plnění různých úkolů. Uživatelé dále získávají body na odměny, které může společnost nastavit. Odměny mohou být reálné, například oběd zdarma, záleží, jak si to společnost nastaví.



---

# Koncept

Na základě předchozí kapitoly jsem vytvořil koncept, který propojuje gamifikaci a projektové řízení. Popisuje aplikaci, která umožňuje plánovat a řídit projekty a podporovat motivaci pracovníků pomocí gamifikačních prvků.

Koncept je stavěn tak, že základní aplikace počítá s rozšířením do budoucna, které přinese další herní možnosti, ale je už nad rámec rozsahu této práce. Tato rozšíření nastíním v závěru konceptu.

## 3.1 Základní myšlenka

Účelem aplikace je primárně gamifikace agilního vývoje software. Lze ji však využít i jindy, kdy se projekt plánuje po menších částech.

Každý sprint je symbolizován drakem. Úkolem týmu je zabít draka. Životy draka jsou součet story poitnů všech úkolů sprintu. Splněním úkolu ztrácí drak životy. Každý hráč má svou postavu, která se postupně zlepšuje tím, jak získává zkušenosti za zabíjení draků.

Pokud se stihne vše, co bylo ve sprintu naplánované, tzn. budou všechny úkoly dokončené včas, drak byl úspěšně poražen a hráči budou odměněni. Pokud se sprint nestihne, drak způsobí hráčům škody a nebude z něj žádná odměna.

Pracovní název aplikace je **Projects and Dragons**.

## 3.2 Uživatel/Hráč

Uživatel se zaregistruje do služby Projects and Dragons a tím mu vzniká uživatelský účet. Může zakládat projekty nebo být přidán do jiných. Uživatel má více postav, na každém projektu právě jednu.

Registrace je velmi jednoduchý proces – uživatel vyplní jméno a příjmení, uživatelské jméno, heslo, email. Pak už se může přihlásit a začít aplikaci používat.

Po přihlášení vidí hráč přehled všech projektů, na kterých pracuje, seřazený podle poslední aktivity na nich.

## 3.3 Postava

Pro každý projekt si hráč vytvoří novou postavu. Když je pozván na projekt, tak si musí vytvořit novou postavu, než ho může začít používat.

Plněním úkolů a zabíjením draků postava získává zkušenosti. Po získání určitého počtu zkušeností postava postupuje na další úroveň. Na získání další úrovně je potřeba mít celkový počet zkušeností daný tabulkou 3.1. Minimální úroveň postavy je 1 (na té postava začíná), maximální 15.

Tabulka 3.1: Tabulka celkového počtu zkušeností nutných ke získání další úrovně

Úroveň	Zkušenosti
2	6
3	13
4	24
5	41
6	66
7	101
8	147
9	206
10	278
11	364
12	465
13	581
14	712
15	858

### 3.3.1 Role

Každá postava má roli, kterou si hráč zvolí při vytváření postavy. Má na výběr ze tří rolí:

- Rytíř (Knight)
- Lučištník (Archer)
- Kněz (Priest)

V základní aplikaci jsou role především pro větší rozmanitost hry, větší využití budou mít v rozšíření.

## 3.4 Projekt

Práce na projektu je pro hráče smysl hry. Projekt je rozdělen do sprintů, ve kterých hráči plní úkoly.

Jakýkoliv registrovaný uživatel může založit vlastní projekt, do kterého zve další hráče. Potom, co se hráč přidá do projektu, založí si postavu.

Na hlavní stránce projektu hráč vidí svoji postavu a postavy spoluhráčů, jejich životy, úroveň, zkušenosti a kolik mají odpracováno story pointů v aktuálním sprintu. Zobrazen je také drak, jeho jméno a zbývající životy.

Důležitou součástí přehledu projektu je burndown graf projektu, zobrazující reálný vývoj sprintu a ideální průběh po dnech.

### 3.4.1 Úkol

Každý sprint se skládá z úkolů, které jsou ohodnoceny story pointy, které přiděluje ten, kdo úkoly vytváří. Po splnění úkolu získává hráčova postava zkušenosti, jejichž množství je závislé na tom, kolika story pointy byl úkol ohodnocen.

Hráč si vybírá úkoly z nabídky v tzv. questlogu (upřesněno v sekci 3.4.4). Úkoly mohou mít nastavenou minimální úroveň, jakou musí mít hráčova postava, aby si ho mohl vybrat. Zajímavější úkoly tak bude možné omezit pouze pro hráče, kteří už mají na projektu něco odpracováno.

Na každého hráče na projektu je přiděleno v každém sprintu 80 zkušeností. Odměna za dokončení úkolu se vypočítá podle vzorce:

$$exp = 80 \cdot playerCount \cdot \frac{taskStoryPoints}{sprintStoryPoints}$$

kde *playerCount* je počet hráčů na projektu, *taskStoryPoints* je počet story pointů úkolu a *sprintStoryPoints* je celkový součet story pointů všech úkolů ve sprintu. Výsledek *exp* je počet zkušeností, které hráčova postava dostane za splnění úkolu.

### 3.4.2 Backlog

Backlog je seřazený seznam všech úkolů, které je potřeba na projektu udělat. Když je vytvořen nový úkol, je zařazen do backlogu. Úkoly je možné v seznamu libovolně přesouvat do požadovaného pořadí.

### 3.4.3 Sprint

Vlastník projektu nebo hráč s dostatečným oprávněním může plánovat sprinty. Z backlogu vybere úkoly a přiřadí je do sprintu. Sprint má nastavené časové období, kdy probíhá.

### 3. KONCEPT

---

Nespouští se však automaticky, ale je potřeba, aby ho vlastník projektu nebo hráč s dostatečným oprávněním ručně odstartoval. Po spuštění sprintu je náhodně vygenerován drak.

#### 3.4.4 Questlog

Každý sprint má svůj questlog, kde jsou zobrazeny všechny úkoly, které je potřeba ve sprintu udělat. Questlog je agile board, který má čtyři sloupce:

**To Do** Úkoly, které je potřeba udělat

**In Progress** Úkoly, na kterých se aktuálně pracuje

**To Verify** Dokončené úkoly, které musí uživatel s dostatečným oprávněním schválit

**Done** Úkoly schválené jako hotové

Hráči si vybírají nepřřiřazené úkoly ze sloupce „To Do“ podle toho, na čem chtějí pracovat. Běžný hráč může úkoly přesouvat pouze mezi prvními třemi sloupci, do posledního sloupce je může přesunout pouze hráč s dostatečným oprávněním. Odměnu za splnění úkolu hráč dostává až potom, co je jeho úkol přesunut do sloupce „Done“.

Každý úkol v questlogu má:

- Název úkolu, který stručně označuje, o co se jedná
- Detailní popis, který v questlogu není vidět, zobrazí se až po rozkliknutí detailu
- Jméno uživatele, který na úkolu pracuje
- Číslo, které ukazuje story pointy daného úkolu
- Minimální úroveň pro získání úkolu, pokud je vyšší než má hráčova postava, zobrazí se červeně

### 3.5 Drak

Drak je základním nepřítelem, kterého je nutné v každém sprintu porazit. Pro každý sprint je náhodně vygenerován drak, který má náhodné jméno a je určitého typu. Drak má životy, které jsou součet story pointů všech tasků ve sprintu. Drak ztrácí životy při dokončování úkolů.



### 3.5.1 Typy draků

Ve hře je několik typů draků. V základní aplikaci je to pouze pro variabilitu. Do budoucna to bude přinášet nové herní možnosti.

- ohnivý (fire)
- vodní (water)
- bouřkový (storm)
- přírodní (nature)
- kamenný (stone)

### 3.5.2 Odměna za zabití draka

Pokud je drak před koncem sprintu poražen, jsou hráči na konci sprintu odměněni zkušenostmi navíc. Ty se rozdělují podle story pointů, které hráč v daném sprintu odpracoval. Nejvíce pracující hráč dostane nejvíce zkušeností.

Hráči získají zkušenosti procentuálně podle úrovně, na které právě jsou, podle vzorce:

$$pctg = 0.5 + \frac{characterStoryPoints}{sprintStoryPoints}$$

kde *characterStoryPoints* je počet story pointů, které hráč v daném sprintu odpracoval a *sprintStoryPoints* je celkový součet story pointů daného sprintu. Výsledek *pctg* vyjadřuje, kolik procent aktuální úrovně získají (mohou získat i více než jednu úroveň, pokud hodně pracují).

### 3.5.3 Trest za nestihnutí sprintu

Pokud hráči nestihnou splnit všechny úkoly ve sprintu, ztrácí zkušenosti postavy, podle toho, kolik úkolů nebylo dokončeno.

Hráči ztrácí zkušenosti procentuálně podle úrovně, na které právě jsou, podle vzorce:

$$pctg = 1 - \frac{sprintStoryPointsDone}{sprintStoryPoints}$$

kde *sprintStoryPointsDone* je součet story pointů úkolů, které byly ve sprintu dokončeny a *sprintStoryPoints* je součet story pointů všech úkolů sprintu. Výsledek *pctg* vyjadřuje, kolik procent aktuální úrovně postava ztratí.

## 3.6 Práva

Každý uživatel má na každém projektu přiřazenou roli (jiná role, než herní role postavy), ke které se vztahují určitá práva. Podle toho může na projektu provádět různé akce. Blíže vysvětleno v tabulce 3.2.

Tabulka 3.2: Tabulka práv jednotlivých rolí na projektu

Akce	Vlastník	Manager	Pokročilý	Základní
Výběr úkol	x	x	x	x
Vytvoření podúkolů	x	x	x	x
Vytvoření úkolů	x	x	x	
Schvalování úkolů	x	x		
Plánování sprintů	x	x		
Spravování projektu	x			

## 3.7 Rozšíření do budoucna

Koncept počítá s rozšířením do budoucna, aby vznikla ještě komplexnější a zajímavější aplikace.

### 3.7.1 Kouzla a schopnosti

Každá role má specifická kouzla a schopnosti, které určitým způsobem ovlivňují hru. Každé kouzlo má danou dobu, než se může znovu použít (cooldown). Schopnosti mohou například chránit ostatní členy, oslabovat draka nebo léčit zranění ostatním.

### 3.7.2 Role

Role postavy ovlivňuje její herní možnosti. Každá role přináší jinou sadu kouzel a schopností, je tedy vhodné mít v projektu různé role, ale není to podmínkou. Role budou zaměřeny na různé typy kouzel:

**Rytíř (Knight)** Je zaměřen na obranu ostatních.

**Lučištník (Archer)** Je zaměřen na oslabování draka.

**Kněz (Priest)** Je zaměřen na léčení ostatních.

### 3.7.3 Vlastnosti

Každá postava má určité vlastnosti, které ovlivňují různé herní prvky.

### Defensivní vlastnosti

Redukují ztráty, které plynou ze hry, a pomáhají se s nimi vyrovnat.

**Armor** Snižuje zranění způsobené drakem

**Health** Zvyšuje počet životů.

**Health regeneration** Obnovuje ztracené životy.

### Ofensivní vlastnosti

Pomáhají hráčům získat více zkušeností, více odměn, atd.

**Power** Zvyšuje poškození, které postava způsobí navíc kromě story pointů, tím toho zabije víc a získá víc zkušeností a odměn.

**Cooldown reduction** Zkracuje cooldown kouzel, takže je možné je používat častěji.

**Magická síla** Pomáhá proti drakům, kteří jsou proti této síle slabí. Jednak snižuje poškození od kouzel draka, kteří jsou proti dané magické síle slabí, a zároveň zvyšuje poškození, které mu způsobují.

#### 3.7.4 Inventář a předměty

Každý hráč má inventář, ve kterém má věci, které právě nemá na sobě, a peníze. Velikost inventáře není nijak omezena.

**Předměty** Předměty mají několik úrovní a tomu odpovídají vlastnosti, které postavě přidají – předmět úrovně 1 přidá méně než předmět úrovně 3. Předměty jsou také různého typu, podle toho, na jakou část těla postavy se oblékají:

- Hlava – přilba, klobouk, kápě
- Ramena – nárameníky
- Hrud' – brnění, vesta, hábit
- Ruce – rukavice
- Nohy – kalhoty
- Boty
- Zbraň – zbraně jsou tří typů a každý je určen pouze pro jednu roli (při rozdělování odměny je toto zohledněno, takže se pokud to bude možné nestane, že by zbraň dostal někdo, kdo ji nemůže nosit).

### 3. KONCEPT

---

- Hůl – kněz
- Luk – lučištník
- Meč – rytíř

**Vzhled předmětů** Barva předmětu je daná podle toho, které magické síly přidává nejvíc. Například pokud předmět přidává nejvíc ohnivé magické síly, bude červený.

Předmět (kromě zbraní) vypadá různě podle toho, na které roli je oblečen. Například pokud je předmět na hrud, na rytířovi vypadá jako brnění, na knězi jako hábit.

#### 3.7.5 Odměny za úkoly

Odměnou za úkol budou kromě zkušeností také herní peníze. Jejich počet bude záviset na story pointech úkolu. U složitějších úkolů je šance, že bude odměna i nějaký náhodný předmět. Tato šance se zvyšuje se složitostí úkolu.

#### 3.7.6 Podúkol

Hráč, který má přiřazený úkol, z něj může vygenerovat podúkol. Oddělí část story pointů z úkolu (nebo nechá na 0). Dále může jako odměnu přidat vlastní herní peníze nebo věc ze svého inventáře. Tato odměna je pak přiřazena k úkolu a hráči jsou peníze hned strženy a odebrána věc z inventáře.

Tento úkol může buď rovnou nabídnout konkrétnímu hráči nebo ho nechá nepřijížený v questlogu a nějaký jiný hráč si ho může vzít sám.

Podúkol schvaluje přímo jeho zakladatel (tj. ten kdo podúkol vytvořil), nikoliv vlastník nebo manažer projektu. Potom, co je podúkol dokončen, drak žádné životy neztrácí. Ty ztratí až potom, co je schválen celý úkol, ze kterého byly podúkoly vygenerované. Zkušenosti ale hráč dostává ihned potom, co je podúkol schválen zakladatelem úkolu.

Za hlavní úkol pak získává hráč méně zkušeností podle toho, kolik story pointů oddělil na podúkoly. Je možné vygenerovat i podúkol z podúkolu.

#### 3.7.7 Drak

Drak v průběhu sprintu generuje zuřivost podle toho, jak moc nad ideálním průběhem v burndown grafu tým je. Hodnota je v intervalu od 0 do 100. Čím je na tom tým hůř, tím víc drak zuří.

V závislosti na typu má drak několik speciálních schopností, které způsobují hráčům značná zranění. Schopnosti ale vyžadují, aby měl určitou hladinu zuřivosti, než je může používat. Např. dokud nemá hladinu zuřivosti alespoň 60 nemůže schopnost použít.

Silnější schopnosti vyžadují více zuřivosti. Takže když tým hodně nestíhá, drak hodně zuří a používá hodně silné schopnosti proti týmu. Cílem by tedy

mělo být pracovat podle plánu a vyhnout se tak drakovým schopnostem. Drakovy schopnosti mají také cooldown, takže je drak nemůže používat pořád, i když má dostatek zuřivosti.

Kromě speciálních efektů drak průběžně útočí na hráče. Síla draka je vygenerovaná podle síly hráčů aktuálně na projektu.

Ve hře existuje 5 druhů draků. Každý druh používá specifický typ magie a je slabý proti jinému druhu magie. Předměty, které jsou jako odměna z draka, jsou pak stejného typu magie, jako byl drak, a lze je využít později, proti jinému drakovi, který je proti tomuto typu magie slabý.

### 3.7.8 Odměny za zabití draka

Potom, co je drak zabit, dostanou hráči odměnu tvořenou určitým počtem předmětů. Minimální počet je dán počtem lidí v týmu, každý je tedy odměněn. Předměty přidávají magické vlastnosti vždy pouze toho typu, jakého byl drak. Přidávají ale i další ofensivní a defenzivní vlastnosti.

Úroveň předmětů je daná částečně náhodně, předmět nejlepší úrovně má nejmenší šanci, že bude v odměně zahrnut, každý další nižší úrovně má větší šanci. Celkový počet je pak doplněn předměty nejhorší úrovně, aby jich byl daný minimální počet. Typ předmětu (rukavice, boty, . . .) je dán zcela náhodně.

Rozdělení probíhá podle skóre hráčů, kterého dosáhli v aktuálním sprintu. Skóre bude primárně podle story pointů úkolů, které hráč ve sprintu vyřešil, ale budou zahrnuté i další faktory jako způsobené poškození navíc, léčení spoluhráčů apod. Hráč s nejvyšším skóre dostane nejlepší věc. Pokud nějaké věci zůstávají po odměnění každého hráče, provádějí se další kola, opět podle skóre hráčů, dokud není vše rozděleno.

### 3.7.9 Avatar

Každá postava má svůj avatar. Avatar si vytvoří předtím, než se přidá do hry. Má některé nastavitelné prvky (barva a tvar vlasů, barva očí apod.). Zbytek vzhledu závisí na předmětech, které má postava právě na sobě.

**Achievements** Za různé speciální úspěchy jsou hráči odměňováni různými odznaky nebo speciálními herními odměnami (titul před jménem, speciální věc pro postavu apod.).

Achievements jsou různé složitosti, některé velmi lehké, jiné těžké. Čím těžší, tím lepší odměna. Mohly by to být např. první postava na úrovni 15, zabití prvního draka, zabití deseti draků, postavy všech rolí na úrovni 15 apod.



---

# Analýza a implementace

Nejprve na základě konceptu vytvořím analýzu aplikace jako podklad pro implementaci. Poté zvolím vhodnou architekturu a vyberu vhodné technologie. Na závěr kapitoly popíšu nasazení aplikace, a jak celá funguje.

## 4.1 Analýza

Zatímco v konceptu jsem se zaměřoval na gamifikační možnosti aplikace a herní principy, zde se zaměřím na analýzu aplikace z pohledu vývoje software. Tato analýza slouží jako podklad pro implementaci.

### 4.1.1 Uživatelské role

Uživatelské role se vztahují vždy ke konkrétnímu projektu. Uživatel může mít na různých projektech různé role.

**Základní uživatel** Základní uživatel může provádět pouze základní akce – vybírat si úkoly, na kterých bude pracovat.

**Pokročilý uživatel** Pokročilý uživatel může kromě výběru úkolu, který bude dělat, také zakládat nové úkoly a zařazovat je do backlogu.

**Manager** Manager projektu má práva provádět na projektu vše, kromě spravování projektu samotného tj. přidávání nových uživatelů a úprava jejich rolí.

**Vlastník projektu** Vlastníkem projektu je ten, kdo projekt založil nebo dostal od vlastníka taková práva. Role umožňuje provádět na projektu veškeré akce.

### 4.1.2 Případy užití

**Zaregistrovat se do systému** Umožňuje uživatelům, aby se zaregistrovali do systému. Při registraci je potřeba vyplnit uživatelské jméno, email, jméno a příjmení, heslo. Po dokončení registrace se uživatel může přihlásit.

**Přihlásit se** Umožňuje registrovaným uživatelům, aby se pomocí svého emailu a hesla přihlásili do systému.

**Upravit uživatelské údaje** Umožňuje uživateli, aby si upravil svoje uživatelské údaje, které vyplnil při registraci – uživatelské jméno, jméno, příjmení a email.

**Změnit si heslo** Umožňuje uživateli, aby si změnil své heslo. Pro změnu je potřeba vyplnit staré heslo a zadat dvakrát nové heslo.

**Založit projekt** Umožňuje libovolnému uživateli založit projekt, na němž je pak v roli vlastník.

**Přidat uživatele na projekt** Umožňuje vlastníkovvi projektu vyhledat další uživatele aplikace, přidat je do projektu a zvolit jejich roli.

**Vytvořit úkol** Umožňuje uživateli s rolí pokročilý uživatel a vyšší založit nový úkol. Při vytváření úkolu musí vyplnit povinná pole – název úkolu, počet story pointů, prioritu a minimální úroveň, kterou musí uživatel mít, aby si mohl úkol vybrat. Dále může vyplnit popis úkolu.

**Editovat úkol** Umožňuje uživateli s rolí pokročilý uživatel a vyšší editovat úkol. Vyplňují se stejná data, jako při vytváření úkolu.

**Smazat úkol** Umožňuje uživateli s rolí manager a vyšší smazat úkoly, které nejsou přiřazené do žádného sprintu.

**Měnit pořadí úkolů v backlogu** Umožňuje uživateli s rolí pokročilý uživatel a vyšší přesouvat úkoly v backlogu a měnit tak jejich pořadí.

**Vytvořit sprint** Umožňuje uživateli s rolí manager a vyšší vytvořit nový sprint. Uživatel musí vyplnit název sprintu, datum startu a konce. Dále musí vybrat z backlogu jeden nebo více úkolů, které se mají v průběhu sprintu vypracovat.

**Editovat sprint** Umožňuje uživateli s rolí manager a vyšší editovat sprint. Vyplňují se stejná data jako při vytváření sprintu.



**Smazat sprint** Umožňuje uživateli s rolí manager a vyšší smazat sprint. Po smazání sprintu jsou úkoly, které v něm byly přiřazené automaticky vráceny do backlogu.

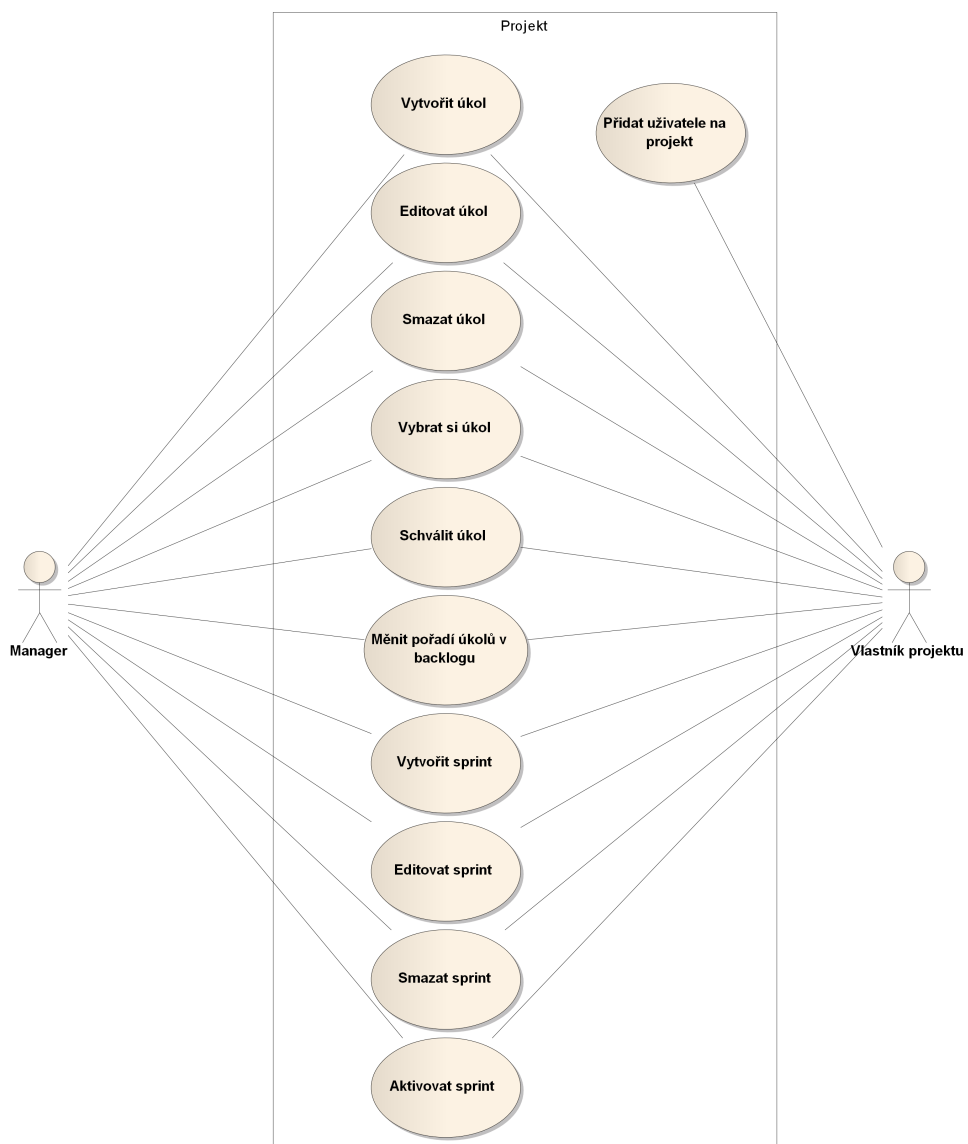
**Aktivovat sprint** Umožňuje uživateli s rolí manager a vyšší aktivovat vybraný sprint. Po aktivování se úkoly tohoto sprintu objeví v questlogu a na dashboardu projektu je vidět průběh sprintu a vygenerovaný drak pro aktivovaný sprint.

**Vybrat si úkol** Umožňuje libovolnému uživateli na projektu vybrat si v questlogu úkol, který zatím není nikomu přiřazen.

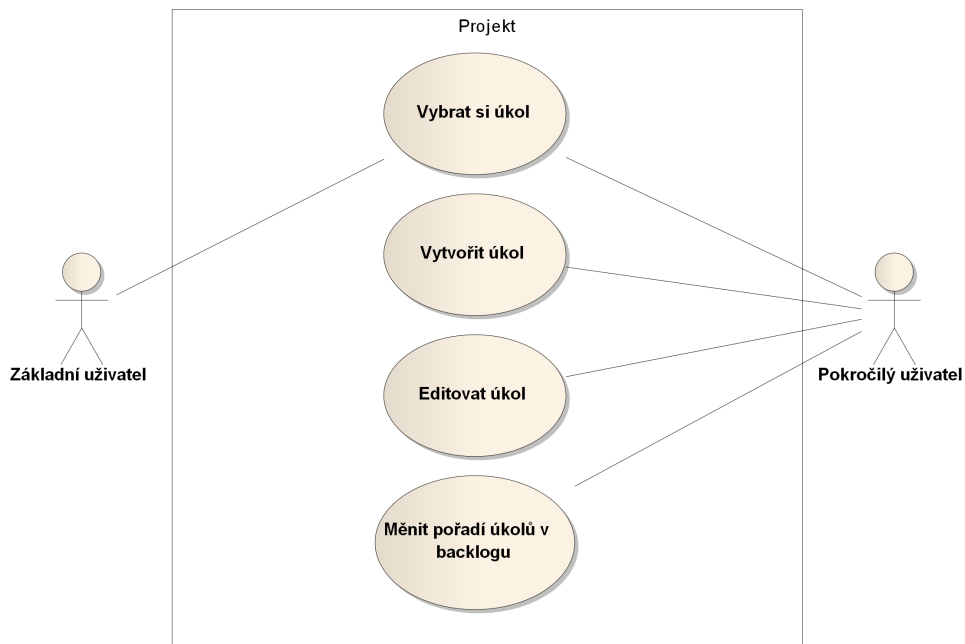
**Schválit úkol** Umožňuje uživateli s rolí manager a vyšší označit úkol jako hotový. Potom, co je úkol poprvé schválený, dostane uživatel, který úkol vypracoval, odměnu.

#### 4. ANALÝZA A IMPLEMENTACE

---



Obrázek 4.1: Případy užití – manažer a vlastník projektu



Obrázek 4.2: Případy užití – běžný a pokročilý uživatel

### 4.1.3 Doménový model

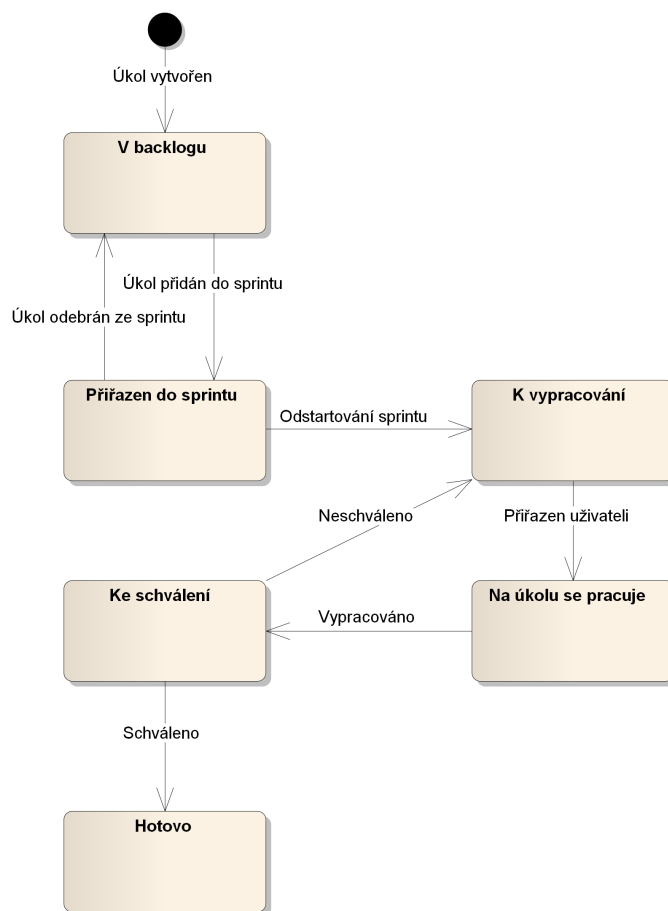
**Uživatel** Uživatel má jméno, příjmení, uživatelské jméno, email a heslo. Pomocí emailu a hesla se přihlašuje do systému. Uživatelské jméno, jméno a příjmení se zobrazují ostatním uživatelům. Email zůstává vždy skrytý.

**Projekt** Na každém projektu pracuje několik uživatelů, kteří jsou přizváni vlastníkem projektu. Na projektu má každý uživatel svoji roli a podle ní mu přísluší určitá práva, co může na projektu dělat.

**Backlog** Backlog je seznam všech úkolů, které nejsou přiřazeny do sprintu. Ke každému projektu se váže právě jeden backlog. Úkoly v backlogu mají své pořadí, které mohou měnit uživatelé s dostatečným oprávněním.

**Sprint** Sprint obsahuje část úkolů vybraných z backlogu. Sprint má název a časové období od kdy do kdy probíhá. Musí být ručně odstartován, aby začal. V tu chvíli se ke sprintu vygeneruje drak.

**Úkol** Úkol má název, popis, story pointy, prioritu a minimální úroveň, jakou hráč musí mít, aby si ho mohl vzít.



Obrázek 4.3: Diagram stavů úkolu

Úkoly jsou po vytvoření zařazeny do backlogu, dokud nejsou přiděleny do nějakého sprintu. Jakmile je sprint odstartován, objeví se v questlogu ve sloupci „To Do“. Potom, co si ho hráč vybere, přesune ho do sloupce „In Progress“, a když ho dokončí, tak do sloupce „To Verify“. Pokud je schválen uživatelem s dostatečným oprávněním, je přesunut do sloupce „Done“ a v tu chvíli dostává hráč odměnu za splnění úkolu. Pokud schválen není, vrací se do „To Do“.

**Postava** Na projektu má každý uživatel svoji postavu. Postava je určité role, získává zkušenosti a podle toho nové úrovně. Má také životy, které může ztrácet za trest, když se nestihne sprint.

Postavu si uživatel vytvoří, když poprvé přijde na projekt. Pak už ji nelze měnit.

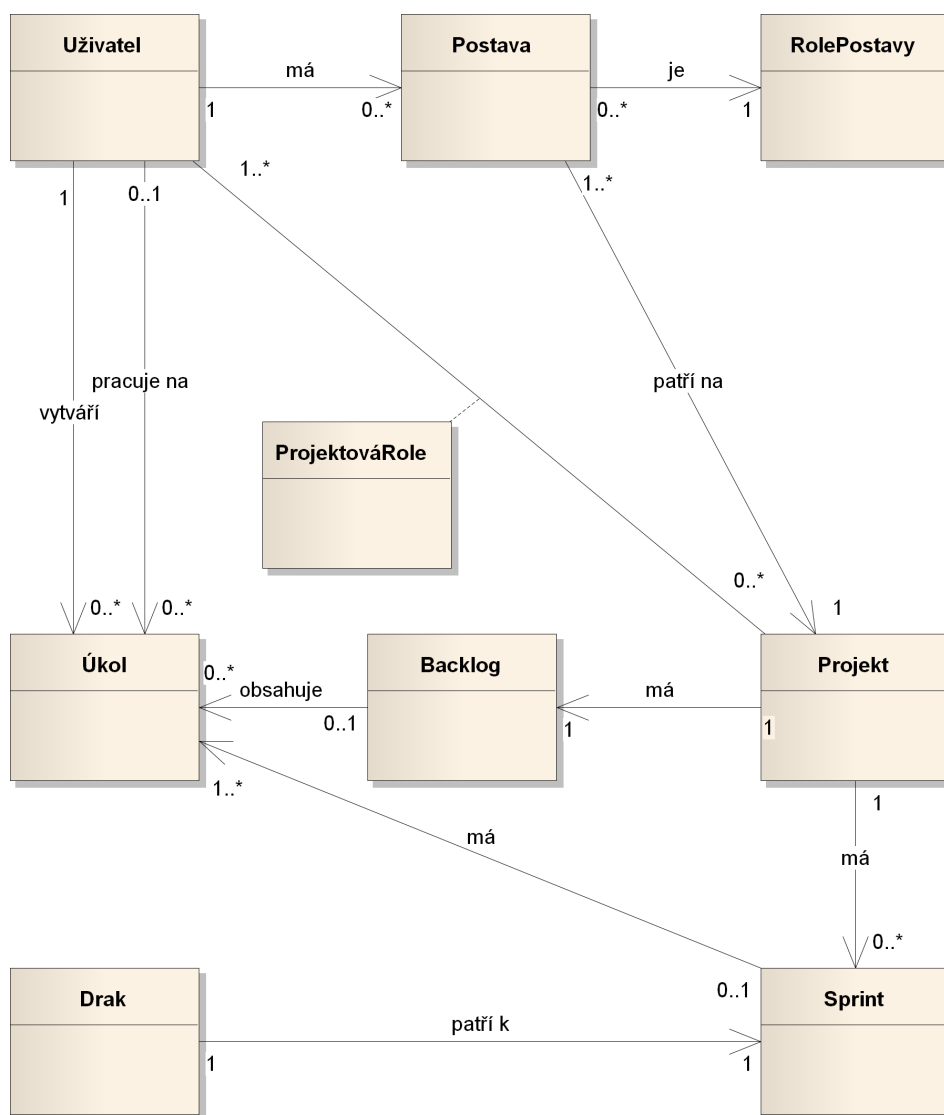
**Drak** Drak je vygenerován když je odstartován sprint a je s ním pevně svázán. Má vygenerované jméno a typ. Počet jeho životů závisí na součtu story pointů všech úkolů ve sprintu. Drak ztrácí životy tím, že jsou úkoly schvalované.

**Role postavy** Každá postava je určité role, kterou si hráč vybere při prvním otevření projektu. k roli se vážou schopnosti a specializační strom.

**Role uživatele na projektu** Na každém projektu má uživatel roli, která určuje jeho práva.

#### 4. ANALÝZA A IMPLEMENTACE

---



Obrázek 4.4: Doménový model

## 4.2 Architektura

Aplikace má architekturu sestávající ze tří vrstev – prezentační, aplikační a datovou.

**Prezentační vrstvu** tvoří webový klient. Klient je vytvořen jako tzv. Single Page Application (SPA), kde je veškerá funkcionality v jedné webové stránce. Klient komunikuje s aplikačním serverem pomocí REST API [27].

**Aplikační vrstva** je tvořena PHP aplikací, která běží na webovém serveru. Aplikace poskytuje REST API, se kterým komunikuje webový klient.

**Datovou vrstvu** představuje relační databáze, která může být buď na stejném serveru jako samotná aplikace nebo na jiném. Aplikace s databází komunikuje pomocí objektově relačního mapování (ORM).

## 4.3 Vybrané technologie

Pro každou vrstvu bylo potřeba vybrat technologie, které odpovídají požadavkům architektury.

### 4.3.1 Prezentační vrstva

**AngularJS** [14] AngularJS patří mezi nejrozšířenější JavaScriptové frameworky pro tvorbu webových aplikací. Za Angularem stojí Google a silná komunita. Existuje velká řada doplňků a implementací jiných knihoven uzpůsobených pro Angular.

**Bootstrap** [15] Bootstrap je velmi oblíbený HTML, CSS a JavaScriptový framework. Usnadňuje a urychluje tvorbu webového front-endu, zejména kódování. Obsahuje množství připravených komponent – různé bloky, dropdown menu, tlačítka a spoustu dalších. Další obrovskou výhodou je, že existuje knihovna, která propojuje AngularJS a Bootstrapové komponenty.

**Jade** [16] Jade je šablonovací systém pro psaní HTML šablon. Velmi zjednodušuje strukturu (není potřeba psát závorky a ukončovací tagy) a tím urychluje psaní HTML. Velmi dobře funguje s Angularem.

**Sass** [17] Sass je rozšířením CSS. Zjednodušuje a zpřehledňuje psaní CSS, není potřeba psát znaky, které se stále opakují a ubírají na přehlednosti – středníky, závorky. Styly je možné do sebe zanořovat a vytvářet tzv. mixiny – skupiny stylů, které lze používat u jiných. Lze tak vytvářet makra pro styly, které jsou specifické pro prohlížeč (border-radius, opacity apod.), takže pak není nutné vše rozepisovat.

**Bower [20]** Bower je balíčkovací systém pro webové komponenty. Pomocí jednoduché konfigurace lze instalovat všechny potřebné knihovny jako například AngularJS nebo Bootstrap.

**Grunt [21]** Grunt umožňuje automatizovat kroky, které je potřeba při vývoji často dělat. Umí například kompilovat Jade do HTML nebo Sass do CSS, spojovat a minifikovat CSS a JS soubory a mnoho dalšího.

### 4.3.2 Aplikační vrstva

**Symfony 2 [18]** Symfony 2 je rozšířený soubor PHP komponent a PHP framework, který využívá velká řada projektů (například redakční systém Drupal). Psaní v Symfony vede k udržování dobré struktury projektu a snadnému udržování a rozšiřitelnosti do budoucna. Navíc existuje velká řada hotových komponent, které je možné použít.

**Doctrine 2 [19]** Doctrine 2 je objektové relační mapování (ORM) a abstraktní databázová vrstva (DBAL) pro PHP. Doctrine projekt je inspirovaný Hibernate ORM [22] z Javy. Umožňuje snadnou práci s databází, podporuje několik databázových systémů (mezi nimi MySQL [23] nebo PostgreSQL [24]). Symfony Standard Edition používá Doctrine 2.

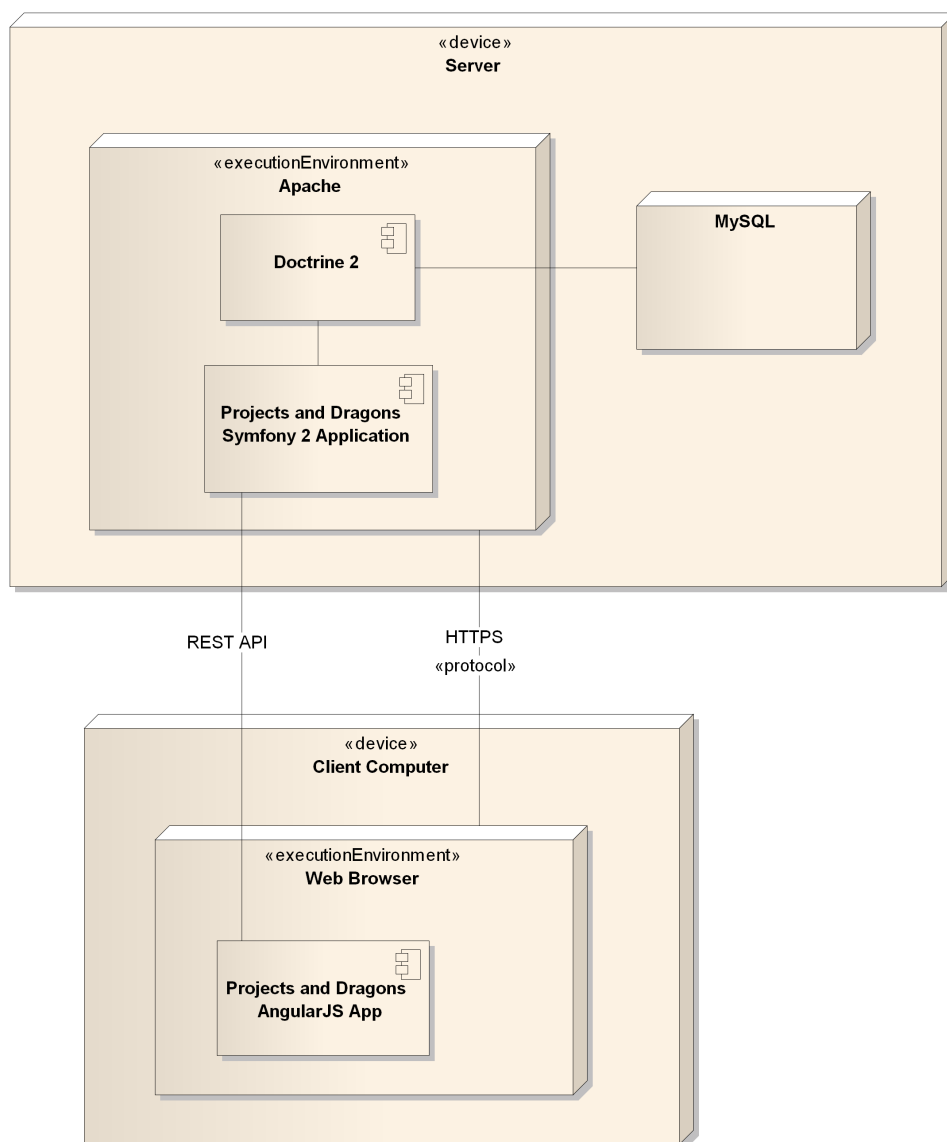
### 4.3.3 Datová vrstva

Doctrine 2 slouží jako prostředek pro spojení mezi aplikací a databází. Jako databázi jsem zvolil **MySQL** [23] vzhledem k její rozšířenosti a snadné použitelnosti. Do budoucna není problém použít MariaDB [25], která je kompatibilní s MySQL a lze ji snadno vyměnit. [26]

## 4.4 Nasazení

Webový klient se stáhne do webového prohlížeče přes HTTP protokol. Následně komunikuje s aplikací na serveru přes REST API. Na serveru je Apache HTTP Server a v něm samotná Symfony 2 aplikace. Pomocí Doctrine 2 komunikuje aplikace s MySQL databází.





Obrázek 4.5: Diagram nasazení



---

# Výsledky

Pomocí několika scénářů, které pokrývají použití aplikace, jsem provedl uživatelské testování, ze kterého vplynuly návrhy na změny a zlepšení do budoucna.

## 5.1 Testování

Pro ověření funkčnosti aplikace jsem na několika lidech, kteří se pohybují v oblasti vývoje software a mají různý stupeň zkušeností s nástroji pro řízení projektů, otestoval aplikaci.

Použil jsem k tomu několik scénářů, které pokrývají použití aplikace a na základě jejich zpětné vazby jsem provedl vyhodnocení.

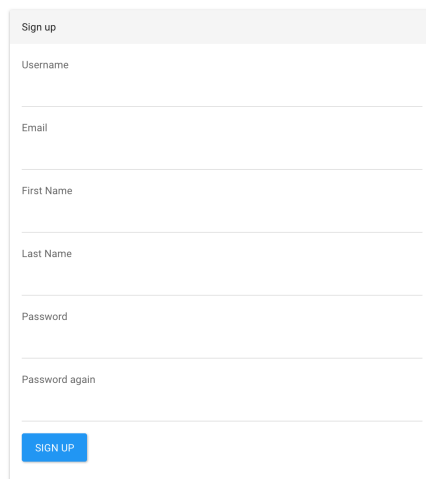
1. Zaregistrujte se do služby a přihlaste se.
2. Vytvořte nový projekt, pozvěte na něj uživatele Jan Slifka v roli manažer.
3. Vytvořte 3 nové úkoly, následně vytvořte sprint, do kterého úkoly přiřadíte, a sprint odstartujte.
4. Vyberte si úkol, na kterém chcete pracovat. Po dokončení ho označte jako hotový.
5. Změňte si své uživatelské údaje a heslo.

### 5.1.1 Registrace a přihlášení

Cílem prvního scénáře je ověřit, jestli je registrace pro uživatele dostatečně jednoduchá a snadno pochopitelná.

Všem uživatelům se povedlo bez problémů zaregistrovat a nikdo neměl problém se následně do aplikace přihlásit.

### Projects and Dragons Gamify your project development



The registration form is titled "Sign up" and contains the following fields:

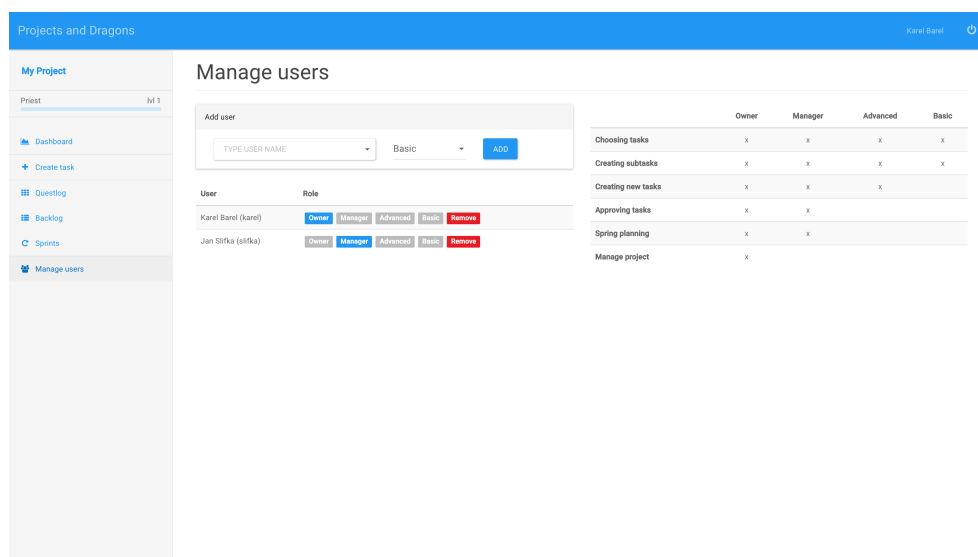
- Username
- Email
- First Name
- Last Name
- Password
- Password again

A blue "SIGN UP" button is located at the bottom of the form.

Obrázek 5.1: Jednoduchý registrační formulář

### 5.1.2 Vytvoření nového projektu

Cílem tohoto scénáře je otestovat jestli je uživatel schopen vytvořit nový projekt. Dále se testuje, jestli dokáže na projekt pozvat další uživatele a nastavit jim požadovanou roli.



The screenshot shows the "Manage users" interface. On the left is a sidebar with navigation options: My Project, Dashboard, Create task, Questlog, Backlog, Sprints, and Manage users. The main content area is titled "Manage users" and includes an "Add user" form with a dropdown for "TYPE USER NAME", a "Basic" role selector, and an "ADD" button. Below the form is a table of existing users:

User	Role
Karel Barel (karel)	Owner   Manager   Advanced   Basic   Remove
Jan Šifka (šifka)	Owner   Manager   Advanced   Basic   Remove

On the right side, there is a permissions matrix table:

	Owner	Manager	Advanced	Basic
Choosing tasks	x	x	x	x
Creating subtasks	x	x	x	x
Creating new tasks	x	x	x	
Approving tasks	x	x		
Spring planning	x	x		
Manage project	x			

Obrázek 5.2: Nastavení uživatelů na projektu

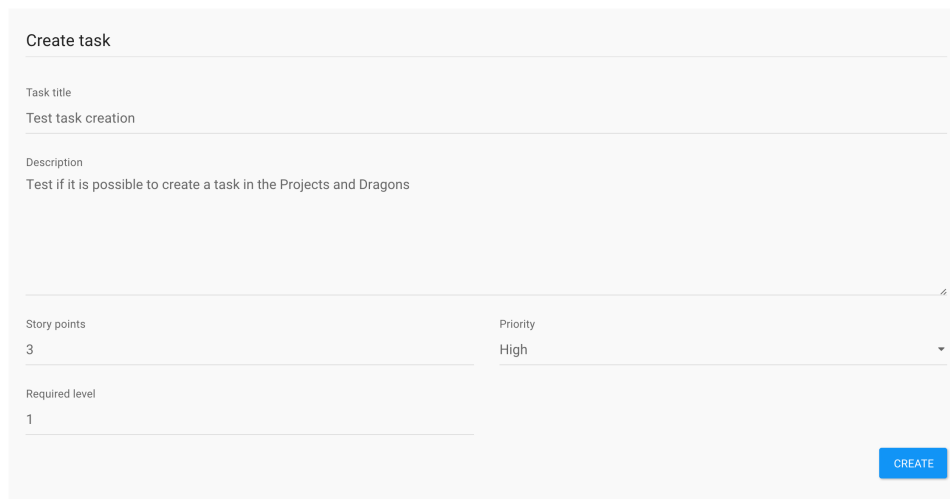
Všichni uživatelé bez problémů vytvořili nový projekt a byli schopni pozvat další uživatele a nastavit jim roli podle zadání.

### 5.1.3 Vytvoření úkolů a sprintu

Cílem je vyzkoušet, jestli uživatel dokáže vytvořit nové úkoly. Dále se ověřuje, že je schopen vytvořit z těchto úkolů nový sprint a odstartovat ho.

Vytváření nového úkolu všichni uživatelé zvládli. Zde však nastala menší komplikace. Po vytvoření úkolu, zůstává uživatel na detailu nově vytvořeného úkolu, který je zároveň formulářem pro editaci úkolu. To pro někoho bylo matoucí a očekával by spíš, že bude přesměrován na detail úkolu. Ostatní uživatelé však s tímto problémem neměli.

Vytvoření nového sprintu proběhlo bez problémů. Objevil se však menší bug při aktivaci sprintu, kdy se sprint někdy hned nezobrazí v přehledu projektu a je nutné stránku znovu načíst.



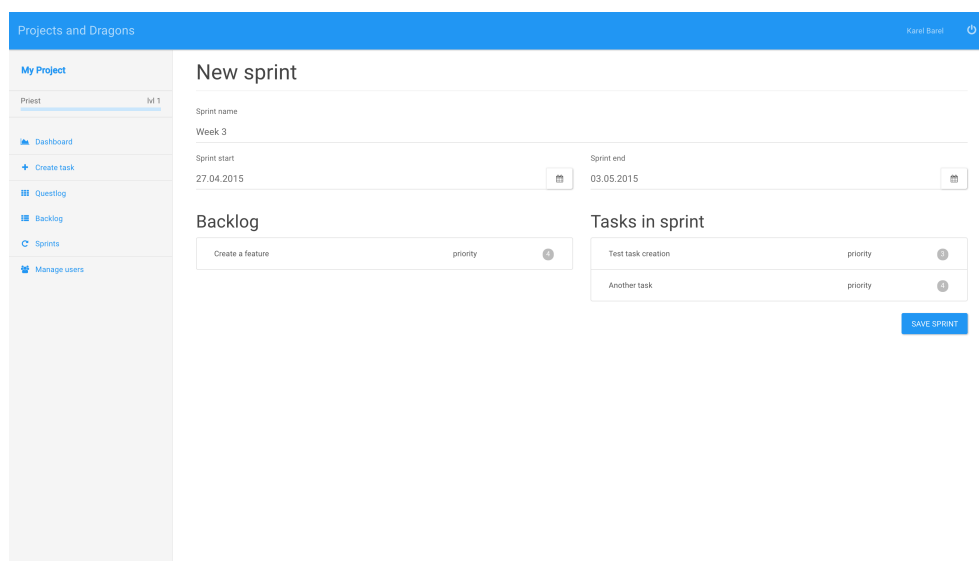
The image shows a 'Create task' form with the following fields and values:

- Task title:** Test task creation
- Description:** Test if it is possible to create a task in the Projects and Dragons
- Story points:** 3
- Priority:** High
- Required level:** 1

A blue 'CREATE' button is located at the bottom right of the form.

Obrázek 5.3: Formulář pro vytvoření úkolu

## 5. VÝSLEDKY

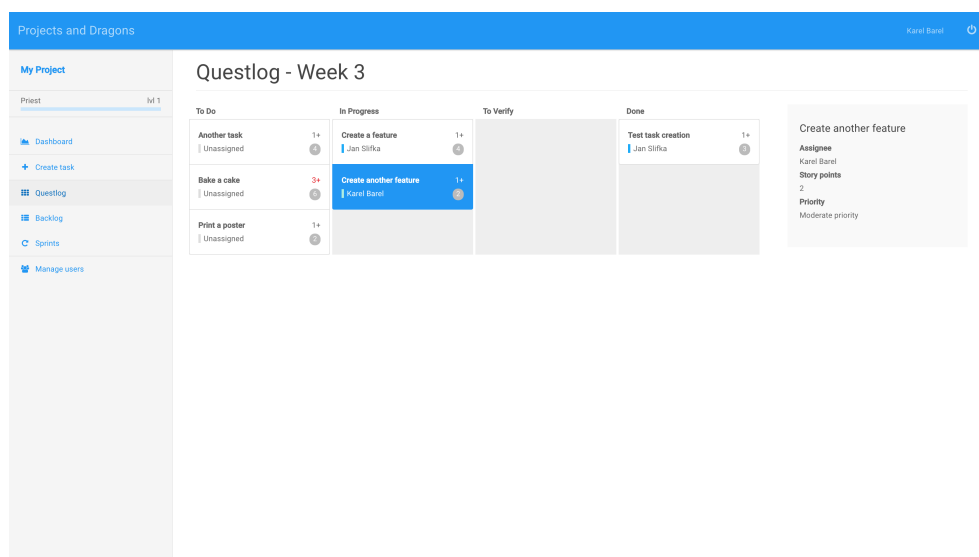


Obrázek 5.4: Vytváření nového sprintu

### 5.1.4 Výběr úkolu a práce na něm

Cílem toho scénáře je otestovat, jestli uživatel snadno chápe používání aplikace v průběhu sprintu, výběr úkolů a jejich přesouvání mezi různými stavy.

Uživatelé pochopili, jak mají questlog používat, dokázali si vybírat úkoly a měnit jejich stav. S dokončováním úkolů sledovali, jak se zlepšují postavy

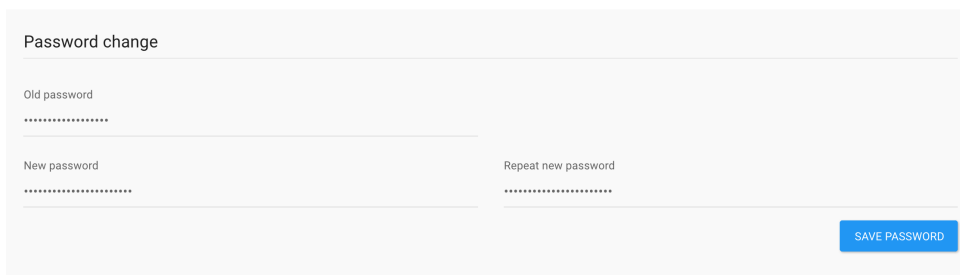


Obrázek 5.5: Questlog aktivního sprintu



The screenshot shows a 'Profile' settings form. It has a title 'Profile' at the top. Below it, there are four input fields arranged in two columns. The first column contains 'First name' with the value 'Jan' and 'Username' with the value 'slifkjan'. The second column contains 'Last name' with the value 'Slifka' and 'Email' with the value 'slifkjan@fit.cvut.cz'. At the bottom right of the form is a blue button labeled 'SAVE CHANGES'.

Obrázek 5.6: Nastavení uživatelského profilu



The screenshot shows a 'Password change' form. It has a title 'Password change' at the top. Below it, there are three input fields. The first is 'Old password' with a masked value '.....'. The second is 'New password' with a masked value '.....'. The third is 'Repeat new password' with a masked value '.....'. At the bottom right of the form is a blue button labeled 'SAVE PASSWORD'.

Obrázek 5.7: Změna hesla

uživatelů, kteří úkoly splnili.

### 5.1.5 Uživatelský účet

Cílem posledního scénáře je otestovat, jestli uživatel dokáže upravovat svůj účet a měnit si své heslo.

Všichni uživatelé bez problémů zvládli svůj profil a heslo upravit.

### 5.1.6 Výsledek testování

Testování ukázalo, že aplikace je pro svůj účel použitelná. Nenarazilo se na zásadní problémy, které by zabraňovaly používání aplikace. Všichni testovací uživatelé zvládli projít testovací scénáře.

Chyby, které se našly, se týkaly především různých drobností v UI – špatné zarovnání některých prvků apod. Jediný trochu větší problém byl při aktivaci sprintu, kdy se aktivovaný sprint hned nezobrazí v přehledu projektu. To ovšem nezabraňuje používání aplikace, po znovu načtení stránky už je vše správně a aktivace sprintu není akce, která by se prováděla často.

Dále testování přineslo podněty na zlepšení nebo další funkcionalitu, která by byla v aplikaci užitečná. Tyto návrhy budou diskutovány v další kapitole.

### 5.2 Výhled do budoucna

Základní aplikace nabízí hodně možností, jak může být v budoucnu rozšiřována a vylepšována, ať už z hlediska gamifikačního nebo z hlediska řízení projektů.

#### 5.2.1 Návrhy na změny a vylepšení aplikace

Aplikace počítá s rozšířením herních prvků. Mnoho z nich je popsáno v rozšíření konceptu. Na základě testování vznikly i další podněty na rozšíření a novou funkcionalitu.

**Více rolí** Až budou implementovaná vylepšení související s rolemi, které si hráči vybírají při vstupu na projekt, dalším herním rozšířením může být přidání nových rolí. Ty budou mít zase jiné schopnosti a jiné vzhledy věcí, takže to bude zajímavější pro hráče, kteří pracují na více projektech.

**Nahrávání souborů k úkolům** K úkolům by měla být možnost nahrávat soubory. Často je zapotřebí přidat screenshot nebo nějaký dokument, který s úkolem souvisí.

**Emailové notifikace** Nastavitelné emailové notifikace, které se budou odesílat při různých významných událostech – splnění úkolu, odstartování sprintu apod.

**Přiřazení úkolu konkrétnímu uživateli** Při vytváření úkolu by měla být možnost přiřadit úkol přímo konkrétnímu uživateli. V současné verzi si uživatelé vybírají úkoly sami, ale někdy může být vhodné úkol už při vytváření přidělit.

**Komentáře k úkolům** k úkolům by měly jít přidávat komentáře, aby bylo možné v případě potřeby položit dotaz a upřesnit zadání přímo v systému.

**Přehled všech úkolů** V aplikaci je backlog, kde jsou zobrazeny všechny úkoly, které nejsou přiřazené do žádného sprintu. Ostatní úkoly se zobrazují pouze u sprintů, ke kterým jsou přiřazené. Pro větší přehlednost by bylo užitečné vytvořit přehled všech úkolů, který by byl podobný jako backlog, ale zobrazoval by všechny úkoly, včetně jejich stavu a přiřazení na sprint.

#### 5.2.2 Mobilní aplikace

Vzhledem k tomu, jak je aplikace řešená – oddělení aplikačního serveru a webového klienta, nebude problém do budoucna vytvořit mobilní aplikaci, která bude používat stejný aplikační server a stejné REST API jako webový klient.



Mobilní aplikace by mohla implementovat část nebo celou funkcionalitu webového klienta. Mohla by mít i nějaké funkce navíc, například notifikace když někdo dodělá úkol nebo když skončí sprint. Vyžadovalo by to sice vytvoření nové funkcionality na straně serveru, ale nemělo by to být složité.



---

## Závěr

Analyzoval jsem pojem gamifikace a zaměřil jsem se na jeho využití v projektovém řízení.

Na základě analýzy jsem navrhl vlastní aplikaci pro řízení projektů, která využívá gamifikační prvky jako motivační faktor.

Vybral jsem vhodné softwarové nástroje a pomocí nich jsem aplikaci implementoval.

Aplikaci jsem otestoval na reálných úlohách, vyhodnotil jsem její úspěšnost a na základě testování jsem ověřil, že aplikace je použitelná, funguje správně a nejsou potřeba žádné zásadní úpravy konceptu a implementace.

Udělal jsem vše, co bylo v zadání.



## Seznam použitých zkratk

**API** Application Programming Interface

**CSS** Cascading Style Sheet

**DBAL** Database Abstraction Layer

**HTML** HyperText Markup Language

**JS** JavaScript

**MVC** Model-view-controller

**ORM** Object-relational mapping

**PHP** PHP: Hypertext Preprocessor

**REST** Representational state transfer

**RPG** Role Playing Game

**SPA** Single Page Application



## Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kódy implementace
├─ thesis.....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
text.....	text práce
├─ assignment.pdf.....	zadání práce ve formátu PDF
├─ thesis.pdf.....	text práce ve formátu PDF





---

## Literatura

- [1] CHAPMAN, Patrick. *Gamification and Game Mechanics Made Simple*. Nordic Book Group, 2012. ASIN: B00AD99UQI.
- [2] SIERING, Greg. *Gamification: Using Game-like Elements to Motivate and Engage Students*. [online]. Březen 2012 [přístup 11. 4. 2015]. Dostupné z: <http://citl.indiana.edu/news/newsStories/dir-mar2012.php>
- [3] ATlassian, INC. *JIRA* [software]. [přístup 14. 4. 2015]. Dostupné z: <https://www.atlassian.com/software/jira>
- [4] HOARAU, Julien. *JIRA Hero* [software]. [přístup 14. 4. 2015]. Dostupné z: <https://marketplace.atlassian.com/plugins/com.madgnome.jira.plugins.jirachievements>
- [5] BUGPOTION. *Jiraffe* [software]. [přístup 29. 4. 2015]. Dostupné z: <http://bugpotion.com>
- [6] LANG, Jean-Philippe. *Redmine* [software]. [přístup 29. 4. 2015]. Dostupné z: <http://www.redmine.org>
- [7] CAMAYO, Mauricio. *Redmine Gamification Plugin* [software]. [přístup 29. 4. 2015]. Dostupné z: [http://www.redmine.org/plugins/redmine\\_gamification\\_plugin](http://www.redmine.org/plugins/redmine_gamification_plugin)
- [8] RedCritic Corp. *RedCritic Tracker* [software]. [přístup 29. 4. 2015]. Dostupné z: <http://www.redcritictracker.com>
- [9] League system. *League of Legends Wiki* [online]. [přístup 12. 4. 2015]. Dostupné z: [http://leagueoflegends.wikia.com/wiki/League\\_system](http://leagueoflegends.wikia.com/wiki/League_system)
- [10] MLEJNEK, Jiří. *Softwarové inženýrství 1*. Agilní přístup. (přednáška) Praha, ČVUT FIT,

- [11] MLEJNEK, Jiří. *Softwarové inženýrství 1. Metodiky vývoje.* (přednáška) Praha, ČVUT FIT,
- [12] PŮLPITEL, Martin. *Projektové řízení.* (přednáška) Praha, ČVUT FIT, 22. 9. 2014
- [13] KRÁTKÝ, Tomáš. ZOUBEK, Bohumír. *Softwarové inženýrství 2. Softwarový proces.* (přednáška) Praha, ČVUT FIT
- [14] GOOGLE, INC. *AngularJS* [software]. [přístup 16. 4. 2015]. Dostupné z: <https://angularjs.org>
- [15] *Bootstrap* [software]. [přístup 16. 4. 2015]. Dostupné z: <http://getbootstrap.com>
- [16] LINDESAY, Forbes. *Jade* [software]. [přístup 16. 4. 2015]. Dostupné z: <http://jade-lang.com>
- [17] CATLIN, Hampton. WEIZENBAUM, Natalie. EPPSTEIN, Chris. *Sass* [software]. [přístup 16. 4. 2015]. Dostupné z: <http://sass-lang.com>
- [18] SensioLabs *Symfony* [software]. [přístup 16. 4. 2015]. Dostupné z: <http://symfony.com>
- [19] Doctrine Team. *Doctrine* [software]. [přístup 16. 4. 2015]. Dostupné z: <http://www.doctrine-project.org>
- [20] Bower Team. *Bower* [software]. [přístup 16. 4. 2015]. Dostupné z: <http://bower.io>
- [21] Bocoup. *Grunt* [software]. [přístup 16. 4. 2015]. Dostupné z: <http://gruntjs.com>
- [22] Red Hat. *Hibernate* [software]. [přístup 18. 4. 2015]. Dostupné z: <http://hibernate.org/orm/>
- [23] ORACLE. *MySQL* [software]. [přístup 23. 4. 2015]. Dostupné z: <http://www.mysql.com>
- [24] The PostgreSQL Global Development Group. *PostgreSQL* [software]. [přístup 23. 4. 2015]. Dostupné z: <http://www.postgresql.org>
- [25] MariaDB Corporation Ab. *MariaDB* [software]. [přístup 23. 4. 2015]. Dostupné z: <https://mariadb.com>
- [26] MariaDB Corporation Ab. MariaDB versus MySQL - Compatibility. *MariaDB* [online]. [přístup 23. 4. 2015]. Dostupné z: <https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-compatibility/>

- [27] RODRIGUEZ, Alex. RESTful Web services: The basics. *IBM developerWorks* [online]. 6. 11. 2008 [přístup 23. 4. 2015]. Dostupné z: <https://www.ibm.com/developerworks/webservices/library/ws-restful/>