Architektura projektu

(NRE) Nadstavba Redmine pro evidenci času

v předmětu KIV/ASWI

Kontext

Cíle projektu

- Vytvořit webovou nadstavbu k systému Redmine pro snadné zadávání času odpracovaného na různých činnostech.
- Uživatelská část bude umožňovat snadné zadání odpracovaného času s jednoduchým výběrem projektu a úkolu. Primární je časový úsek, pro který je vybrána činnost (analogie s diářem).
- Využití správy projektů, úkolů, uživatelských účtů a další dat ze systému Redmine přes jím poskytované REST API (“nevytváříme nic nového”).
- Administrační část bude umožňovat zadání šablon reportů, nejlépe formou uploadu XLSX 9souborů s domluvenými placeholdery.
- Generování reportů ve formě XLSX nebo CSV souborů.
- Maximální jednoduchost ovládání a efektivita práce uživatelů.

Technologie

Technologie poměrně vyplývaly sami ze zadání.

- PHP v naší aplikaci bude sloužit jako jako fasáda pro volání jednotlivých requestů z UI rozhraní. Rovněž bude sloužit jako vstupní bod aplikace.
- JavaScript v naší aplikaci tvořit uživatelské rozhraní a vyhodnocovat uživatelské akce (události).
- AJAX používáme pro asynchronní komunikaci s fasádou aplikace.
- REST API bude využito pro získání dat ze systému Redmine a jejich doplnění o evidenci času.

Knihovna a frameworky použité při implementaci uživatelského rozhraní:

- knihovna jQuery - zajišťuje snadnější práci s JavaScriptem
- framework bootstrap - zajišťuje grafické rozvržení formulářů (přihlašovací, export) (http://getbootstrap.com/)
- framework DHTMLX - zajišťuje hlavní komponentu aplikace (kalendáře) (http://dhtmlx.com/docs/products/dhtmlxScheduler/)

Knihovny použité při implementaci exportů:

- PHPWord - slouží pro export reportů do souboru s koncovkou *.docx (https://phpword.codeplex.com/)
- PHPExcel - slouží pro export reportů do souboru s koncovkou *.xlsx a *.xls (https://phpexcel.codeplex.com/)

Knihovny použité při implementaci fasády:

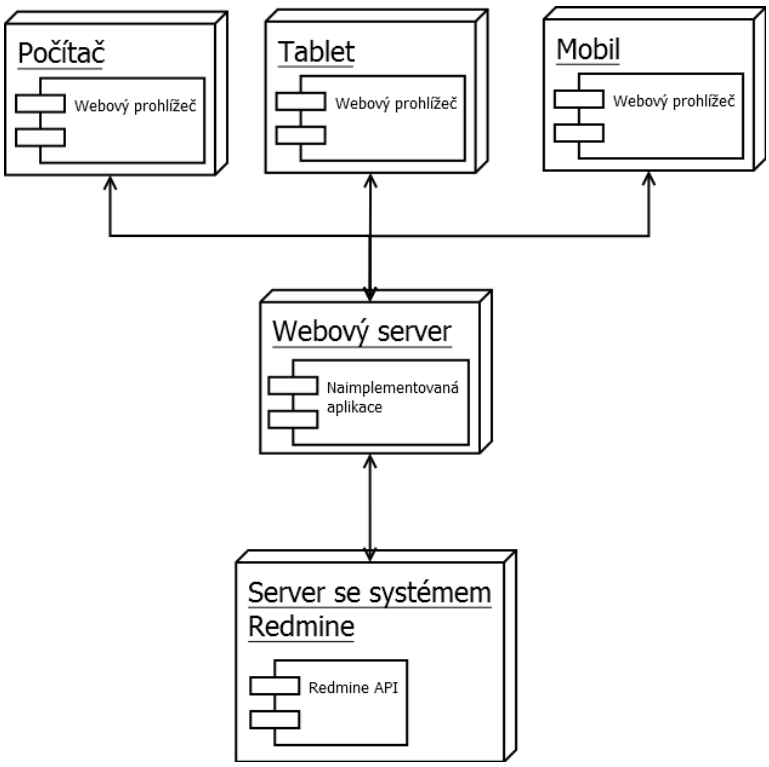
- https://github.com/kbsali/php-redmine-api wrapper pro volání Redmine REST API

Navržená architektura

Pro popis architektury bude použit systém 4+1 pohledů na softwarový produkt.

Fyzický pohled

Aplikace fyzicky poběží webovém serveru a bude vzdáleně komunikovat se systémem Redmine pomocí REST API.



Procesní pohled

Uživatelské rozhraní je podporováno pro všechny webové prohlížeče na všech operačních systémech. S fasádou komunikuje asynchronně pomočí technologie AJAX.
Fasáda aplikace musí běžet na webovém PHP serveru. Se systémem Redmine komunikuje synchronně pomocí REST API.
Aplikace je škálovatelná jak z pohledu administrativního tak funkčního. Vzhledem k tomu, že uživatelské rozhraní běží na straně klienta, tak aplikace je odolná vůči připojení velkému množství uživatelů.

Logický pohled

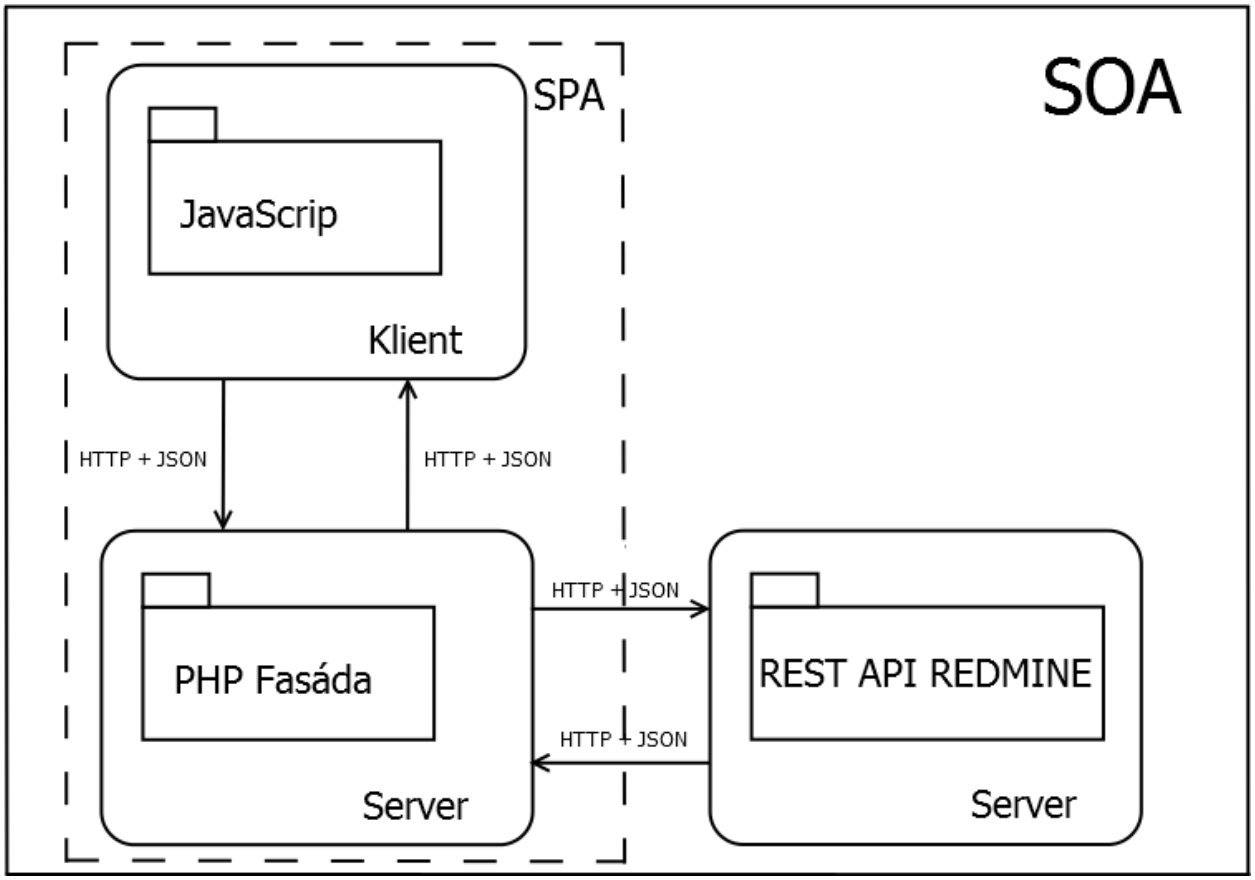
Pro vývoj aplikace jsme zvolili architekturu SPA (Single-page aplikace), která komunikuje přes PHP fasádu s Redmine REST API. Komunikace probíhá mezi sousedícími vrstvami, z čehož plyne, že klientská část (uživatelské rozhraní) nemůže přímo komunikovat se systémem Redmine, ale zašle požadavek na server (fasáda), ten ho zpravuje a pomocí REST API Redmine se dotáže pro příslušná data, či je upraví. Celý tento pohled se dá zařadit pod architekturu SOA (Service-oriented architecture).

Jedná se o obecný architektonický vzor založený na spolupráci navzájem nezavislých služeb. Služba je určitá část funkčnosti aplikace, která je zpřístupněna pomocí definovaného rozhraní. V našem případě se jedná o REST API.

REST (Representational State Transfer) je orientováno datově z toho plyne, že rozhraní webových REST služeb je použitelné pro jednotný a snadný přístup k tzv. zdrojům. Všechny zdroje jsou jednoznačně identifikovatelné pomocí URI a jsou reprezentovány (a přenášeny) pomoc různých datových formátů (v našem případě JSON).

REST definuje základní CRUD(Create-Read-Update-Delete) metody pro manipulaci s těmito zdroji (metody mění stav zdroje). Tyto metody jsou poté realizovány pomocí odpovídajících HTTP metod (GET, POST, PUT, DELETE).

Viz: http://www.redmine.org/projects/redmine/wiki/Rest_api



Vývojový pohled

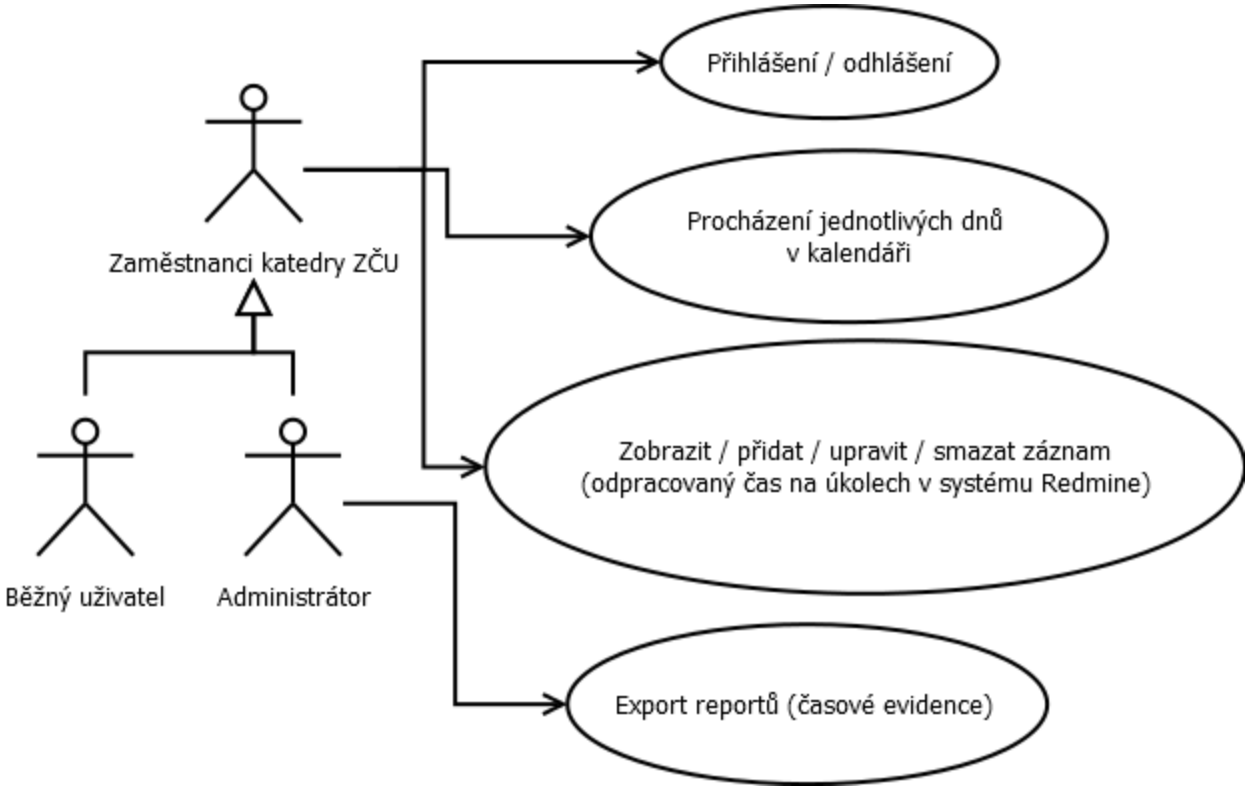
Vývojové prostředí si může zvolit každý developer podle svého uvážení. Nástrojovou sadu, Git a Redmine, určil mentor.

Komentování kódu

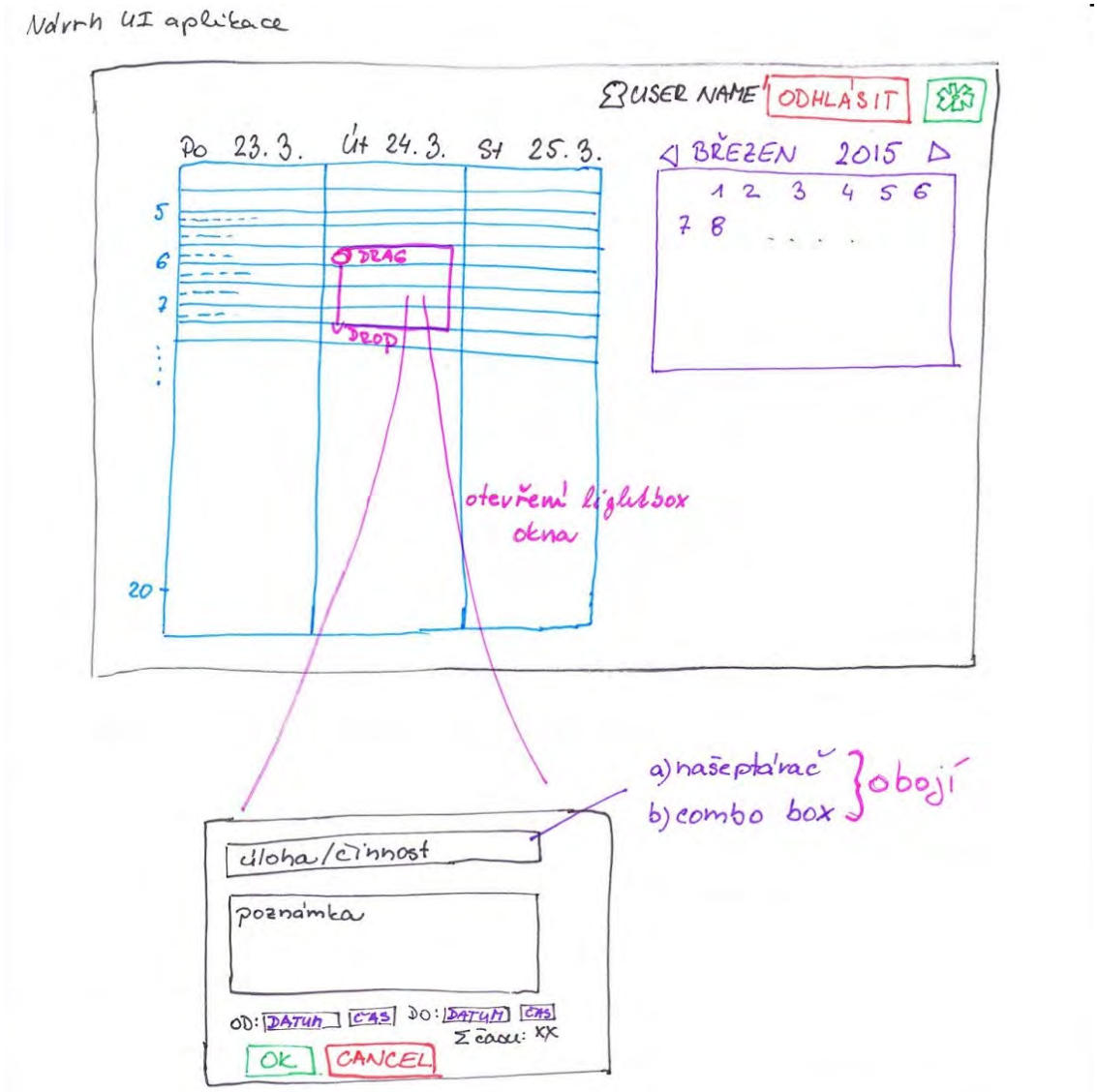
Komentáře v kódu jsou požadovány ze strany zákazníka. Kód se komentuje česky. Není nutné komentovat zřejmé konstrukce, např. HTML tagy. Naopak je nutné komentovat klíčové části kódu z důvodu udržitelnosti aplikace a možnosti snadného rozšíření.

Scénáře užití

Následující obrázek znázorňuje případ užití.



Návrh uživatelského rozhraní



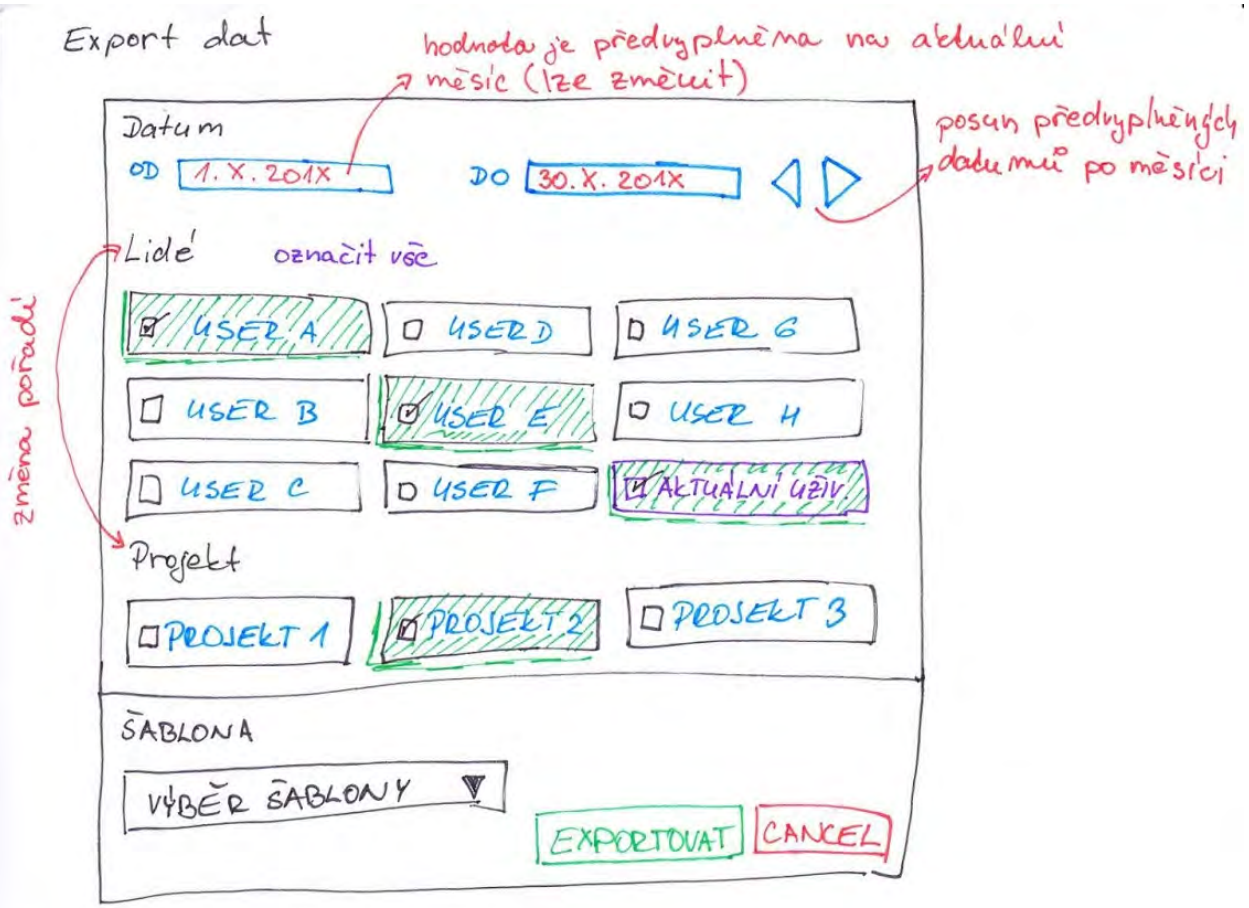
Use case: práce na 1 úkolu v různých časech (10-12, 14-15, 17-20)

⇒ zadávací formulář si bude pamatovat poslední zadanou hodnotu (úlohu i poznámku) – při kliknutí do poznámky úloha zůstane; při klk. do úlohy zmizí i poznámka (pouze ta původní)

Use case: jednotlivé úkoly by se měly překrývat

a) omylem dojde k překrytí stejných činností ⇒ tyto se sjednotí v jeden (poznámky se spojí)

b) překryv různých činností < warning + ruční oprava / automat. oprava tak, aby se čin. nekryly



Filtry:

a) zobrazit jen ty uživatele, kteří mají status „aktivní"

b) při zadání data dojde k filtraci uživatelů, kteří v tomto datu mají nějakou činnost (i pokud jsou „neaktivní")

c) při volbě projektu(ů) dojde k fildr. uživ., kteří jsou na daném projektu přiřazeni

- filtry se budou kombinovat
- půjde zobrazit všechny uživatele