

Zápočtová úloha z předmětu KIV/ZSWI

OBJEKTOVÝ NÁVRH APLIKACE

pro Webovou aplikaci Vyhlídkové lety / Fly-by

15. 4. 2018

Tým: Fly-by tým

Členové:

Martin	Brom	bromm@zcu.cz
Miroslav	Krýsl	krysl@zcu.cz
Jaroslav	Chejn	jchejn@zcu.cz

Obsah [\[Na tomto místě znovu vygenerujte obsah.\]](#)

1. ÚVOD.....	3
1.1 ÚČEL SYSTÉMU.....	3
1.2 SLOVNÍČEK DEFINIC, POJMŮ A ZKRATEK.....	3
1.3 ODKAZY NA DALŠÍ DOKUMENTY.....	3
2. KONTEXT A ARCHITEKTURA SYSTÉMU.....	3
2.1 KONTEXT SYSTÉMU.....	3
2.2 ARCHITEKTURA SYSTÉMU, PŘEHLED PODSYSTÉMŮ.....	4
2.3 ZVOLENÁ TECHNOLOGIE, PROGRAMOVACÍ JAZYK AD., DŮVODY.....	4
3. TYPY INFORMACÍ ZPRACOVÁVANÉ SYSTÉMEM.....	4
4. PODSYSTÉMY.....	4
4.1 <PODSYSTÉM 1>.....	4
4.1.1 <i>Přehled tříd</i>	4
4.1.2 <i>Třída <název třídy></i>	5
4. Přiřazení tříd/modulů programátorům.....	5

1. Úvod

Tento dokument popisuje návrh softwarového systému – webové aplikace Fly-by. Obsahuje návrhové diagramy databáze, tříd i celkové provázanosti systému.

1.1 Účel systému

Nový systém pro plánování vyhlídkových letů je určen pro společnost provozující tyto lety. Je cílená pro zákazníky, kteří si budou moci pomoci této aplikace plánovat a objednávat vyhlídkové lety. Dále bude možná administrace obsahu aplikace.

1.2 Slovníček definic, pojmů a zkratk

- MVC - Návrhový vzor – modely, pohledy a kontrolery

1.3 Odkazy na další dokumenty

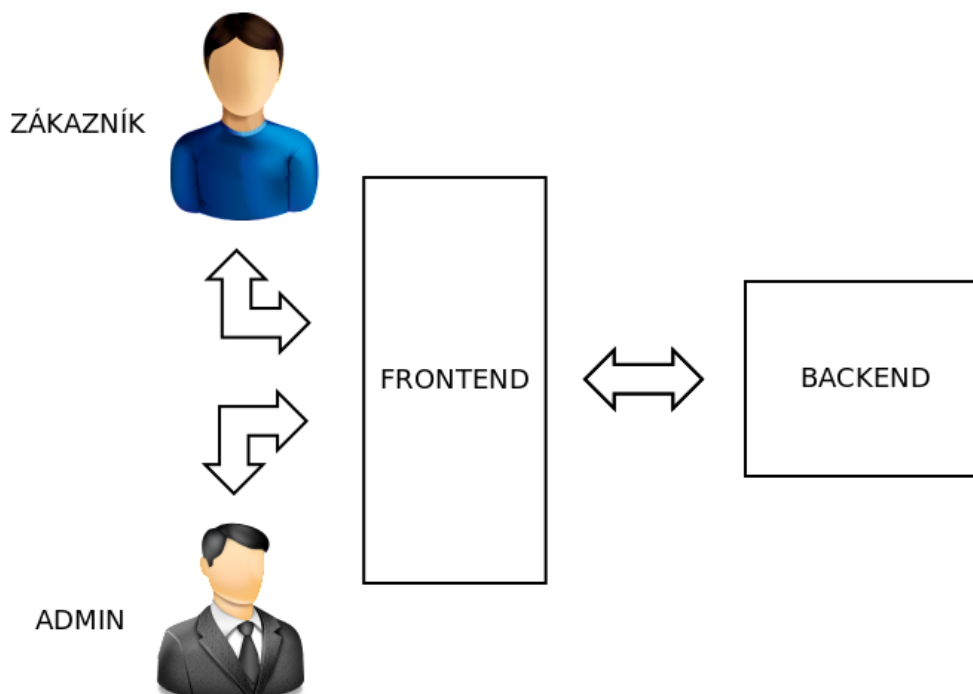
Dokumentace Laravelu 5.5 - <https://laravel.com/docs/5.5>

2. Kontext a architektura systému

2.1 Kontext systému

Uživatel bude pomocí aplikace vytvářet objednávky vyhlídkových letů. Správce webu je poté může potvrzovat a dále s uživatelem komunikovat ohledně objednávky.

2.2

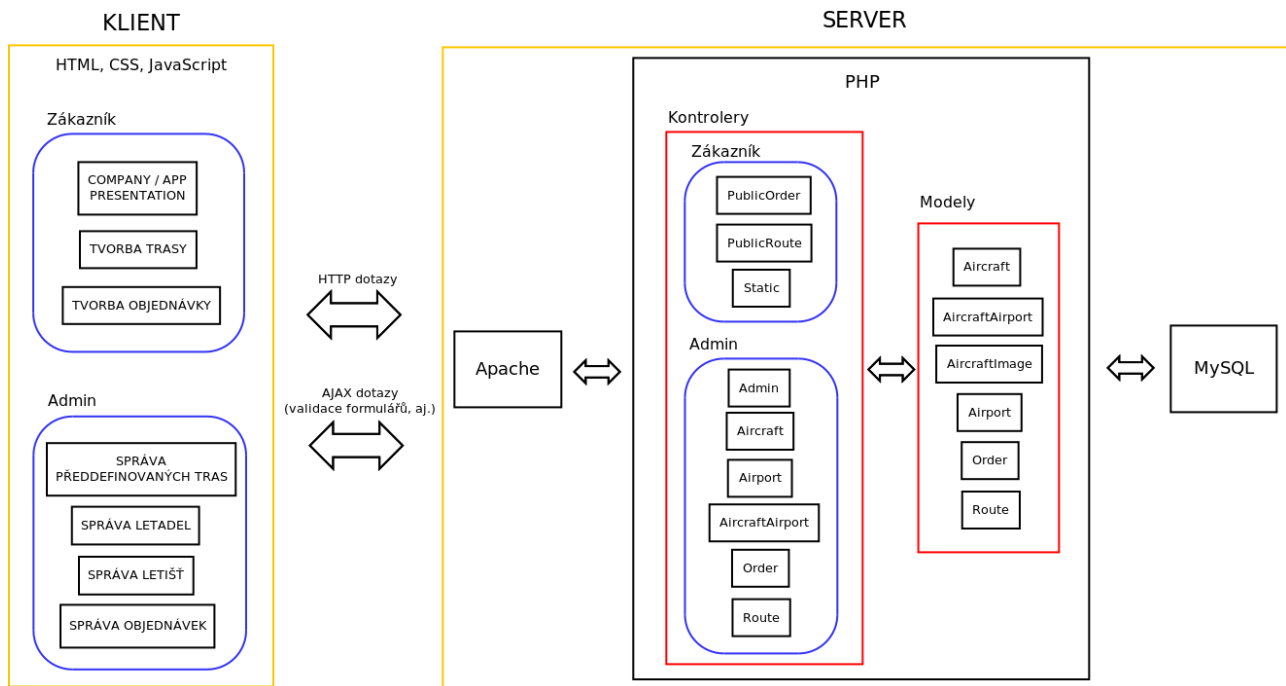


Architektura systému, přehled podsystémů

Aplikace se skládá z těchto částí:

- **Frontend**
 - Interakce uživatelů s aplikací
 - Oddělená zákaznická a administrační část

- Komunikace s backendem pomocí HTTP protokolu
- **Backend**
 - Přijímá HTTP požadavky a vrací odpovědi
 - Webový server Apache (nebo jiný) zavolá hlavní script aplikace, která zpracuje požadavek
 - Samotná backendová část aplikace je rozdělena podle návrhového vzoru MVC:
 - Kontrolery se starají o logiku aplikace a dotazují se modelů na data, která předávají do pohledů
 - Modely se starají o získávání dat z databáze
 - Pohledy slouží pouze jako zobrazení dat a obsahují prvky pro zpětnou interakci s aplikací
 - Databáze MySQL obsahuje všechna data potřebná pro běh aplikace



2.3 Zvolená technologie, programovací jazyk ad., důvody

Frontend

- HTML5, CSS3, JavaScript - Jedná se v podstatě o standard technologií webového frontendu
- Leaflet.js – javascriptová knihovna pro zobrazování map a interakci s nimi, využívá OpenStreet mapy nebo i jiné poskytovatele map
- Bootstrap 4 – CSS / JavaScript framework pro snadnou tvorbu webového uživatelského rozhraní

Backend

- PHP 7.0 – Většina webových hostingů běží se scriptovacím jazykem PHP, verzi 7.0 podporují všechny nejpoužívanější hostingy
- MySQL – Taktéž jako u PHP, většina hostingů běží s databázovým systémem MySQL
- Framework Laravel – PHP framework pro snadné definování cest, akcí a interakcí s databází na backendu

3. Typy informací zpracovávané systémem

3.1 Vstupní a výstupní soubory

Aplikace bude zpracovávat jako vstupní soubory pouze soubory popisující neletové zóny. Souvby budou ve formátu SUA nebo OpenAir.

Jako výstupní soubory bude aplikace generovat pouze pdf soubory s informacemi o objednaném letu a s unikátním klíčem objednávky.

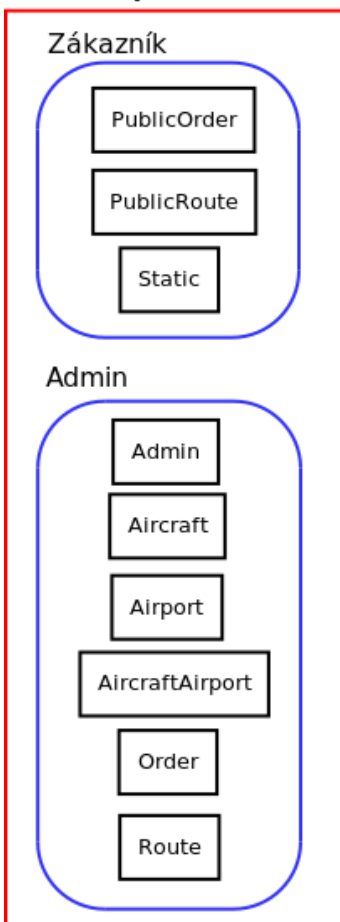
3.2 Databáze

Databáze bude vytvořená v MySQL. Modely vytvořené pomocí laravelu budou většinou odpovídat tabulkám v databázi.

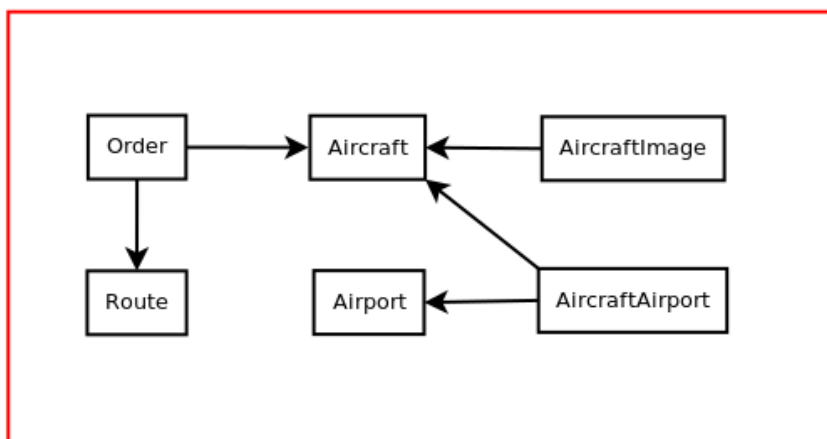


4. Návrh systému

Kontrolery



Modely



4.1 Modely

Modely jsou pouze obalovací třída okolo databázových tabulek, mají tedy stejné atributy.

Funkčnost všech modelů včetně relací mezi nimi je popsána v velmi dobře zpracované dokumentaci Laravelu. Bylo by tedy zbytečné je zde uvádět.

4.2 Kontrolery

Kontrolery obsahují logiku zpracování dotazů. Stejně jako u modelů je očekávaná funkčnost kontrolerů popsána v dokumentaci Laravelu.

Kontrolery budou odpovídat tzv. resource kontrolerům – pro manipulaci se zdroji (letadla, letiště, objednávky atd.) Budou obsahovat následující metody pro provedení CRUD akcí.

4.2.1 index

Akce pro základní zobrazení zdroje, většinou se jedná o zobrazení seznamu všech prvků konkrétního zdroje.

4.2.2 create

Akce pro zobrazení stránky s možností vytvoření nového prvku zdroje.

4.2.3 store

Akce pro uložení nového prvku zdroje.

4.2.4 edit

Akce pro zobrazení stránky s možností editace prvku zdroje.

4.2.5 update

Akce pro provedení/potvrzení editace prvku zdroje.

4.2.6 destroy

Akce pro odstranění prvku zdroje.

5. Přiřazení částí aplikace programátorům

Frontend - Miroslav Krýsl

Backend - Martin Brom

Databáze, Testování - Jaroslav Chejn