DELTA – Střední škola informatiky a ekonomie s.r.o.

Ke Kamenci 151, PARDUBICE

**IS – LPU Pardubice**

Informační systém – OK LOKOMOTIVA PARDUBICE

Zeman Martin

4.A

Obor č. 18: Informatika

2021/22

**Zadání maturitního projektu z informatických předmětů**

Jméno a příjmení: *Martin Zeman*

Školní rok: *2021/2022*

Třída: *3.A*

Obor: *Informační technologie 18-20-M/01*

Téma práce: *Modernizace rozhraní IS LPU*

Vedoucí práce: *RNDr. Jan Koupil, Ph.D.*

**Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:**

Cílem práce je vytvořit nové uživatelské rozhraní stávajícího informačního systému oddílu orientačního běhu LPU Pardubice.

Webové uživatelské rozhraní bude navrženo jako responzivní a bude se připojovat ke stávající databázi MySQL.

V případě potřeby autor provede i modernizaci databáze a navrhne úpravy původního administračního rozhraní IS tak aby zůstalo funkční.

**Stručný časový harmonogram (s daty a konkretizovanými úkoly):**

• září: Analýza stávajícího řešení, hledání nekompatibility s aktuálními verzemi databáze

• říjen: migrace dat na aktuální verzi databáze

• listopad-prosinec: Implementace vlastního rozhraní

• leden-únor: Testování, opravy chyb, připojení původní administrace

**Prohlášení**

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze zdroje a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne 30.3.2022

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(vlastnoruční podpis)

**Poděkování**

Největší díky patří vedoucímu mého maturitního projektu RNDr. Janu Koupilovi, Ph.D. za odborné rady a nápady, které mi pomohly s tímto projektem.

A druhé díky patří mým spolužákům, kteří mě podrželi a povzbuzovali, i když projekt nešel zdaleka podle představ.

**Resumé**

Cílem projektu je vytvořit nové uživatelské rozhraní pro členy pardubického klubu OK LOKOMOTIVA.

Uživatelé zde budou mít přehled o novinkách v oddíle, plánovaných závodech na tuto sezónu a dalších akcích, které si pro ně klub připraví.

Přímo z tohoto systému se také uživatelé mohou na jednotlivé akce registrovat nebo se z nich naopak odhlásit.

Administrátoři databáze mají svojí vlastní stránku pro administraci, která nijak nesouvisí s tímto projktem

**Klíčová slova**

Webová aplikace, IS – LPU, Laravel, MySQL, PHP, Blade, Orientační běh, Maturitní práce

**Resume**

**Keywords**

Web application, IS – LPU, Laravel, MySQL, PHP, Blade, Orienteering, Graduation thesis

Obsah

# Úvod

Projekt by měl nahradit stávající uživatelské rozhraní pardubického oddílu orientačního běhu OK LOKOMOTVA PARDUBICE. Staré rozhraní bylo neresponzivní, mělo zastaralý design a díky tomu nebylo uživatelsky přívětivé.

Tento projekt má za cíl tyto neduhy vymýtit, a vytvořit hezké a uživatelsky přívětivé rozhraní, ať už se na něj koukáte na počítači nebo telefonu.

Dále je cílem zmodernizovat a zefektivnit stávající databázi. Ta v současném stavu běží na zastaralé verzi MySQL a chybí v ní jakékoliv hlídání integrity databáze například v podobě cizích klíčů.

# Technologie

## Framework

Framework je software, který slouží jako základna, na která se staví následná konkrétní aplikace.

Každý framework je vytvořený k jinému účelu a má podle toho i předpřipravené funkce, které pomohou programátorovi zpřehlednit a zjednodušit jeho práci a kód samotný. Každý programátor však musí zvolit správný framework pro to, co po aplikaci požaduje, aby dělala.

### Laravel

Laravel je open source framework pro webové aplikace pod licencí MIT a postavený na jazyce PHP. Je založený na architektuře MVC (model-view-controler). Což znamená, že kód je rozdělený do třech základních kategorií. Do modelu, který obsahuje data a funkce, view, to slouží k prezentaci dat například v HTML, nebo v případě Laravelu Blade, a controleru, který spravuje komunikace mezi modelem, view a uživatelem.

Laravel je designovaný hlavně pro komunikaci s MySQL databází. Ale dá se použít i s dalšími druhy SQL databází, a dokonce i s NoSQL. Celkově je připravený na reálné použití a má připravenou většinu věcí které jsou společné pro většinu projektů, například registraci a přihlašování uživatelů. Má pro to připravené jak funkce, tak i tabulky, které vám vytvoří jedním příkazem hned po tom, co databázi připojíte k aplikaci.

## Šablonovací engine

Šablonovací engine je software, který se stará o prezenční vrstvu databáze, o takzvaný frontend. Kód v něm zapsaný je pak pro vyhledávače překládán do čistého HTML. Proč však nepoužíváme čisté HTML? Důvod je jednoduchý. Šablonovací engine nám dává možnost používat například podmínky nebo cykly, což výrazně zpřehledňuje výsledný kód.

### Blade

Blade je jednoduchý, avšak mocný šablonovací engine, který je součástí Laravelu. Tento šablonovací engine má svojí hlavní výhodu v tom, že nám opravdu pouze zpřehledňuje kód, zatímco vše, co je v něm napsané je uložené ještě v druhém PHP souboru, kde už je to opravdu pouze čisté PHP, které se přepisuje pouze v případě změny zdrojového kódu, tedy souboru Blade. Tento způsob zajišťuje co nejmenší zátěž výpočetního výkonu serveru na kterém nám aplikace běží.

V soubory se musí pojmenovávat s koncovkou .blade.php a v Laravelu jsou ukládány do adresáře /resources/views.

## Databáze

Pokud začneme historií databází, tak nejstarší předchůdce databází jsou papírové kartotéky. Ty fungovali na stejném principu jako fungují dnešní databáze. Data zde byla členěna podle jednotlivých kritérií a mohli se na sebe navzájem odkazovat, například pacient má tuto nemoc, informace o nemoci jsou uloženy v této části kartotéky v tomto svazku.

Časem se data přesunuly z papíru na výpočetní techniku. Nejdříve to byly děrné štítky a následně, jak vývoj techniky pokračoval, se data přesouvala na výkonnější a efektivnější úložná média. Postupně vznikly hierarchické databáze, poté relační databáze, následně objektové, avšak ty nedostáli svým cílům, a místo aby vytlačily relační, tak se s nimi spojily, a tak vznikli dnešní objektově-relační databáze.

Mezi databázové objekty patří například pohledy, indexy, procedury a funkce nebo například E-R schéma

### MySQL

MySQL je databázový systém založený na relačním konceptu databáze s podporou objektů. Systém je multiplatformní. Komunikace s databází probíhá v jazyce SQL, který je však drobně modifikovaný. Není tedy možné vzít příkaz z této databáze a vložit ho například do PostgeSQL (jiný databázový systém založený také na jazyce SQL).

MySQL je v dnešní době hojně využívaná pro drobné projekty, a to hlavně díky jednoduchosti implementace. Druhým zásadním důvodem je dostupnost. MySQL je kompletně zdarma. Velice často je využívána v kombinaci s backendem, který je napsaný v jazyce PHP.

MySQL je jednou z jednodušších SQL databází, ale to má vliv na výkon. Databáze je designovaná hlavně na rychlost.

### Normalizace databáze

Normalizace databáze je postup, při kterém přeorganizujeme, roztřídíme a rozdělíme data tak, abychom co nejefektivněji využívali potenciál relačních databází.

Při normalizaci databáze řešíme konzistenci (úplnost) a redundanci (opakování) dat.

Při normalizaci databáze se postupuje podle normálových forem. Normálové formy jsou pravidla, podle kterých upravujeme strukturu databáze. Při normalizaci musíme splňovat normálové formy postupně jelikož pro splnění každé normálové formy je vyžadováno splnění všech předchozích forem.

Ve většině projektů se požaduje splnění 3NF a o další se nesnažíme. Je to způsobeno tím, že výsledek ve výkonu a udržitelnosti databáze neodpovídá vynaloženému úsilí, které je třeba, aby databáze tyto normálové formy splňovala.

#### Nenormalizovaná forma (UNF)

Nenormalizovaná forma je zde pro databáze nesplňující normálové formy.

Pro její splnění stačí, že databáze existuje.

#### Nultá normální forma (0NF)

V nulté normálové formě se tabulka vyskytne tehdy, jsou-li nějaká data uložená v jednom sloupečku tabulky dále dělitelná. Což je však přímo v rozporu s relačním modelem, který nám říká, že data v jednom poli jsou dále nedělitelná.

Tabulka splňuje buďto nultou, nebo první normálovou formu, jelikož tyto dvě si navzájem odporují.

#### První normální forma (1NF)

První normálová forma nám naopak říká, že každé pole tabulky obsahuje atomické, dále nedělitelné, hodnoty.

#### Druhá normální forma (2NF)

Druhá normálová forma nám říká, že každý neklíčový atribut je plně závislý na každém kandidátním klíči.

Jinak řečeno každý atribut, který není kandidátním klíčem tabulky, musí souviset se všemi kandidátními klíči tabulky. Pokud nesouvisí tak by měl být umístěn do jiné tabulky, kde toto pravidlo splňovat bude.

#### Třetí normální forma (3NF)

Třetí normálová forma nám říká, že všechny neklíčové atributy musí být vzájemně nezávislé.

Tedy pokud odstraníme kterýkoliv sloupeček neklíčových hodnot tabulky, neporuší se nám tím integrita dat uložených v této tabulce.

#### Boyce-Coddova normální forma (BCNF)

Boyce-Coddova normální forma nám říká že, atributy, které jsou součásti primárního klíče, musí být vzájemně nezávislé.

Tedy žádný sloupeček, který je primárním klíčem, nebo jeho součástí, nesmí být závislí na jakémkoli jiném sloupečku tabulky.

#### Čtvrtá normální forma (4NF)

Čtvrtá normálová forma nám říká, že relace popisuje pouze příčinnou souvislost mezi klíčem a atributy.

To znamená, že v jedné tabulce máme uloženy pouze atributy, které spolu souvisí. Každý atribut tedy musí souviset se všemi ostatními atributy tabulky. Nestačí, že všechny atributy souvisí s primárním klíčem.

#### Pátá normální forma (5NF)

Pátá normálová forma nám říká, že relace již není možno bezeztrátově rozložit.

<http://programujte.com/clanek/2008071900-normalizace-relacnich-databazi/> plus borkovcová