

Dokument Architektura

Zadání

Cílem zadání je vytvořit desktopovou aplikaci, která vykreslí všechny linky a spoje (zadané, nebo všechny v kraji). Cílem není hledat optimální cestu. Linky bude možné vybrat ze seznamu zaškrtnutím, stejně tak zóny, které se mají zobrazit.

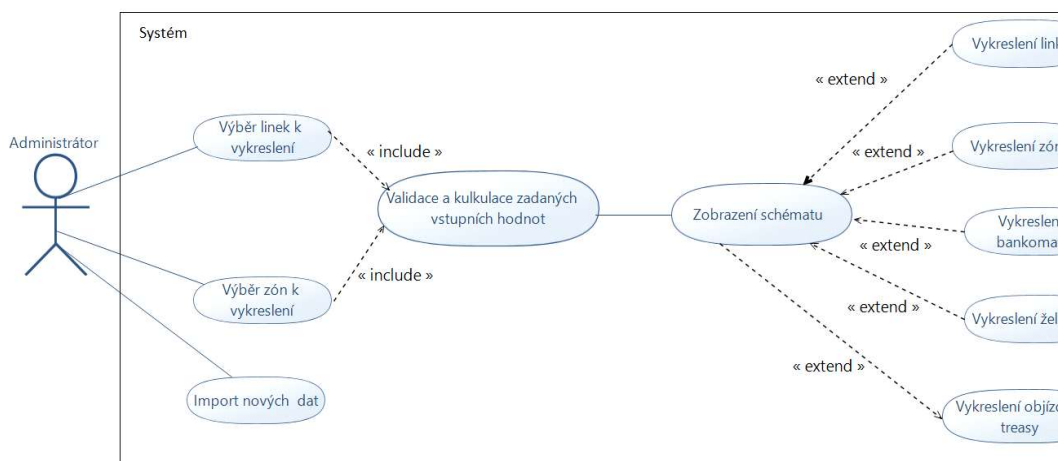
Vykreslení linek:

- Linky budou vykreslovány do mapy (lokální nebo Google Maps) a do schématu, na základě volby uživatele
- Ve schématu by linky měly být vykresleny pokud možno v osmi směrech
- Pokud to bude přehledné, linky budou vykresleny různými barvami (v mapě vždy). Pokud to nebude přehledné, budou stačit jednobarevně (jako ve schématu). U barevných linek by mělo být vždy na začátku, nebo na konci uvedeno číslo linky. Linky, které vedou ve stejném směru, mohou být vykresleny stejnou barvou.
- Železniční trati budou zakresleny včetně zastávek (značkou železnice jako v mapách)
- Zastávky v rámci města nebudou ve schématu vykresleny všechny (např. zastávky Rokycany,, aut. nádr.; Rokycany,, Železná; Rokycany,, nem. budou zakresleny jen jako Rokycany). V mapě naopak vykresleny budou.
- V mapě i ve schématu by měli být hraniční zastávky barevně odlišeny. Navíc by mapy měli obsahovat další informace (bankomat České spořitelny, výdejní místo plzeňské karty – seznam včetně ikon poskytne zadavatel) a to buď zakreslené přímo do mapy (ikona), nebo poznámka do legendy co se v daných místech nachází.

Dále, kromě vykreslení linek, by mělo být možné zakreslování objízdných tras v případě, že je na lince výluka (možnost zadat alternativní cestu linky). Aktualizace dat (linek IDP a zón) se bude provádět nahráním nového zdrojového souboru s daty. Všechny výstupy by mělo být možné exportovat do PDF a JPG.

Use-Case diagram

Use-Case diagram nám ukazuje účastníka a činnosti, které se mohou v aplikaci provádět. Účastník je pouze jeden z toho důvodu, že zde není potřeba nijaká zvláštní režije na údržbu aplikace. Sám uživatel nahraje do aplikace aktuální sadu dat (linky a zóny IDP), aplikace si data zpracuje (vazby include) a následně může uživatel provádět nad daty potřebné operace (vazby extend).



Obr. 1 Use-case diagram

Použité technologie

Aplikace bude vytvořena v programovacím jazyce Java. Databáze bude použita lokální Java DB. Aplikace musí být podporována operačními systémy Windows, které používá společnost POVED.

ERA model

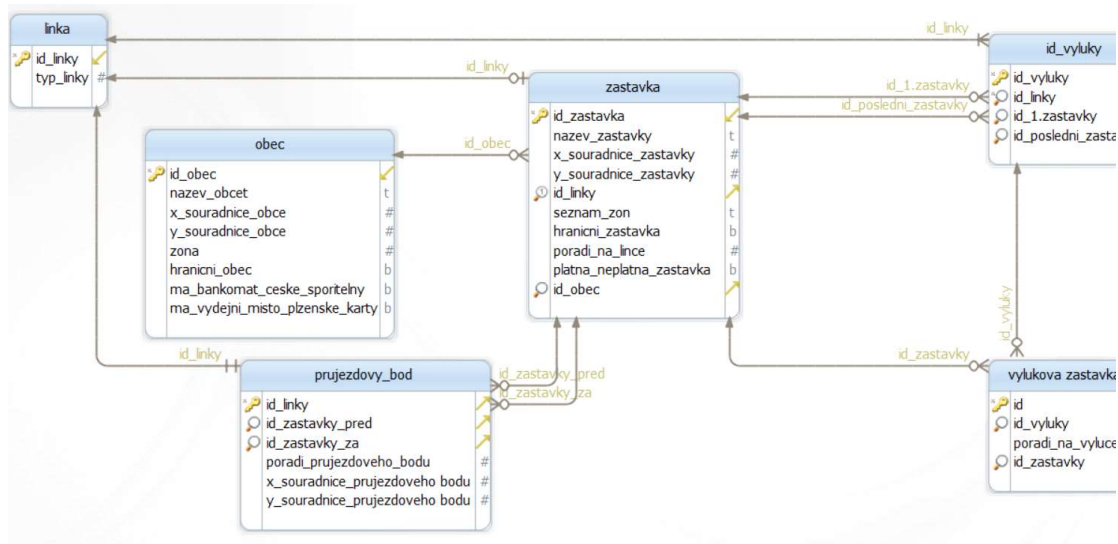
Databáze aplikace se skládá z šesti tabulek, jak můžeme vidět na obrázku níže. Význam jednotlivých tabulek bude stručně popsán.

Tabulky

- linka – Obsahuje ID a typ linky.
- obec – Obsahuje GPS polohu obce potřebnou pro správné umístění do mapy a schématu. Dále obsahuje informace o zóně, do které obec spadá a příznak, zda se jedná o hraniční zastávku nebo nikoli. Tabulka ještě obsahuje doplňkové informace jako je přítomnost bankomatu České spořitelny a výdejního místa plzeňské karty.
- prujezdovy_bod – Tato tabulka slouží pro nastavení průjezdných bodů v mapě. Protože v mapě musí linka kopírovat pozemní komunikace, je nutné nastavovat průjezdné body.
- zastávka – Informace o zastávce (poloha, zóna, zda se jedná o hraniční zastávku, ...).
- id_vyluky – Tato tabulka obsahuje informace o výlukové trase. Kromě ID výluky obsahuje ID linky, které se výluka týká a ID zastávek, kde výluka

začíná a kde končí.

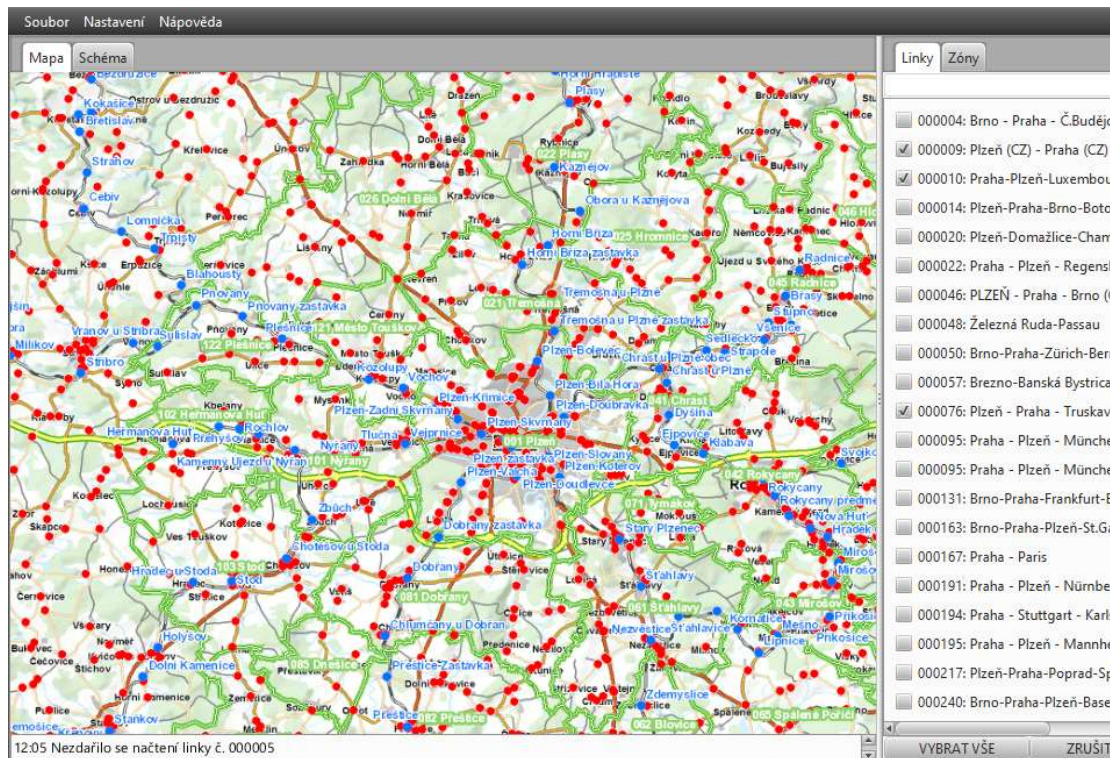
- vyluková zastavka – Tabulka obsahuje informace pro správné seřazení vylukových zastávek u konkrétní linky.



Obr. 2 – ERA model databáze

Uživatelské rozhraní

Grafické uživatelské rozhraní navržené pro aplikaci splňuje požadavky zadavatele a bylo jím schváleno. Pomocí tohoto rozhraní bude uživatel schopen ovládat celou aplikaci. Kromě vykreslování linek a zón do mapy nebo schématu se jedná o import aktuálních zdrojových dat, export mapy nebo schématu do JPG a PDF a vkládání objízdných tras.

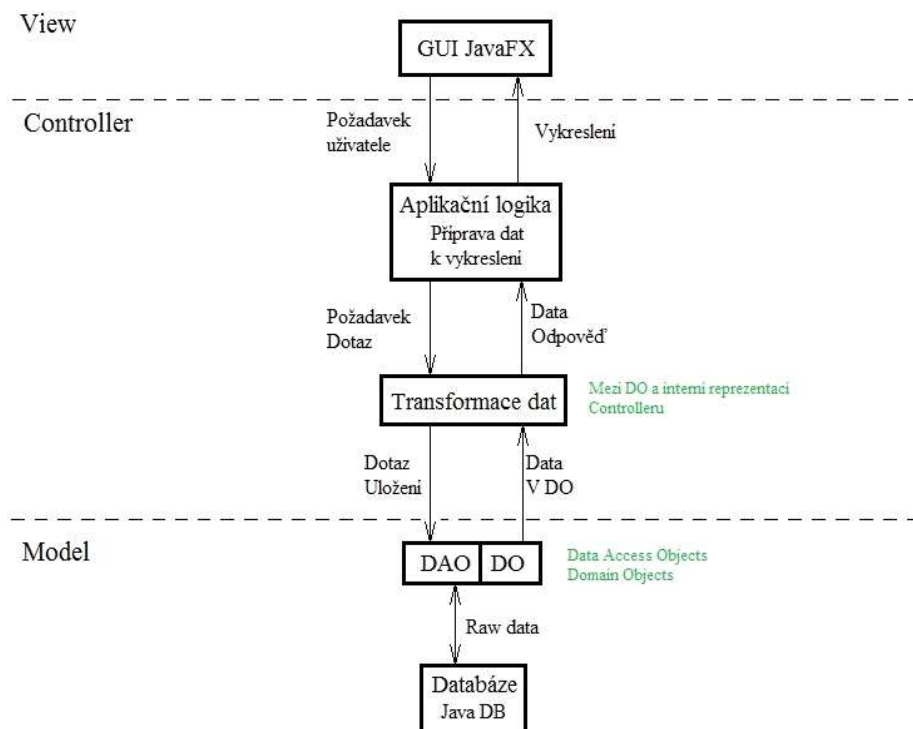


Obr. 3 – Návrh GUI aplikace

Architektura

Aplikace bude mít několik vrstev, které odpovídají vrstvám MVC architektury. Je zde však několik odchylek:

- Mezi vrstvami View a Model není přímá vazba. Požadavky atd. na obě strany chodí vždy přes Controller. To je z toho důvodu, že data je před zobrazením třeba vhodně transformovat a připravit k vykreslení.
- Vrstva Controller je rozdělena na dvě sub-vrstvy - "mozek" aplikace, který obsahuje aplikační logiku a vrstvu sloužící k transformaci dat mezi doménovými objekty a jejich reprezentací vhodnou k dalšímu zpracování controllerem před vykreslením. Tato vrstva není zařazena do modelu z toho důvodu, že proces, kterým budou data transformována, je vysoce závislý na konkrétní úloze, která se nad daty bude provádět - což se vylučuje s úlohou modelu "načti z databáze data do doménových objektů" v naší upravené MVC architektuře. Zjednodušeně řečeno Controller potřebuje data v jiné formě, než běžně Model poskytuje v doménových objektech.



Obr. 4 – Architektura aplikace