

# Dokumentácia inžinierskeho diela

## 1. Úvod

Tento dokument opisuje realizáciu nášho zadania v rámci predmetu Tímový projekt na tému “Predikcia ceny vozidla na základe inzerátu [CarAd]”. Projekt vyvíjame pod názvom Predicar a ide o produkt, ktorý má na základe textu inzerátu odhadnúť cenu vozidla a poskytnúť iné užitočné štatistiky alebo odhady spojené práve s predajom daného vozidla.

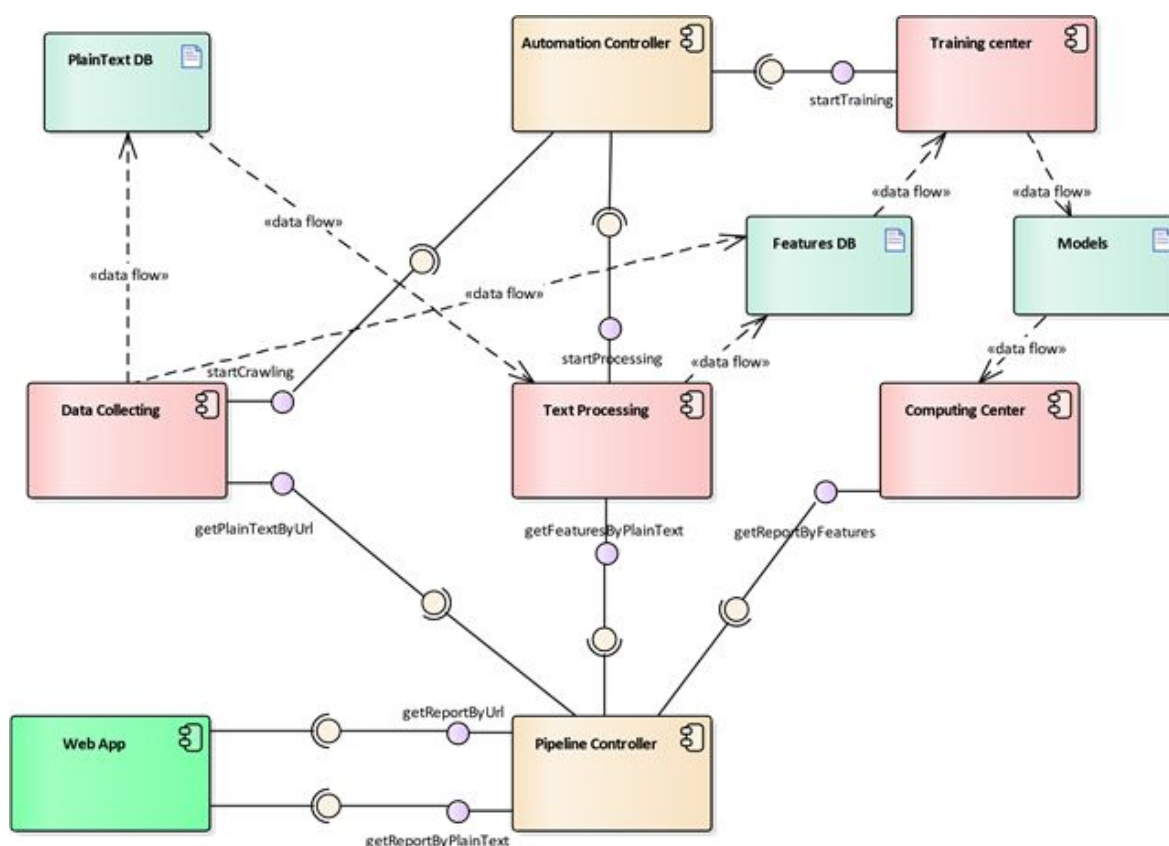
### 1.1 Globálny cieľ na ZS

**“Chceme vidieť vypočítanú predikciu ako odhad ceny v našom rozhraní pre štandardný vstup.”**

## 2. Základná štruktúra systému

V tejto kapitole poskytneme sumár nami navrhutej architektúry systému. Okrem toho zhrnieme aj dátový model, podľa ktorého uchováваме v našom systéme dáta o vozidlách a jednotlivých inzerátoch.

### 2.1 Architektúra



## Obrázok 1 Základný pohľad na architektúru systému Predicar

Náš systém je zameraný na strojové učenie, pomocou čoho dokáže odhadnúť ceny vozidiel. Kvôli tomuto si architektúra vyžaduje časť zameranú na **zber dát**. Tieto dáta potom uchováame v dokumentovej databáze.

Strojové učenie sa v našej architektúre objavuje v dvoch inštanciách: **spracovanie textu** pre extrakciu črt z textu inzerátu a samotný **model pre odhad cien**. Zatiaľ čo spracovanie textu nepotrebujeme sústavne trénovať, ak dokáže správne interpretovať texty inzerátov, odhad cien musíme pravidelne trénovať, aby odhady cien vždy brali do úvahy aktuálne dáta. Preto je súčasťou našej architektúry aj **trénovacie centrum**.

Interakcia so systémom je zabezpečená cez jednoduché **webové rozhranie**. Pomocou tohto rozhrania môže používateľ zadať odkaz na inzerát alebo jeho samotný text a pozrieť sa na odhady a štatistiky od nášho produktu.

## 2.2 Dátový model

Dátový model nášho systému pozostáva z jednej entity, keďže sme zvolili dokumentovú databázu ako úložisko našich dát. Tieto dáta odrážajú vlastnosti vozidla a kolekciu inzerátov, ktoré sa naň vzťahujú. Údaje, ktoré o vozidlách uchováame sú nasledovné:

- **VIN:** VIN číslo predstavuje unikátny identifikátor vozidla
- **licence\_plate:** zoznam ŠPZ značiek pre vozidlo s daným VIN číslom
- **price:** výsledná cena vozidla agregovaná z cien nájdených v inzerátoch pre dané vozidlo
- **features:** vlastnosti vozidla, ktorých presný zoznam určíme pri zbere dát
- **ads:** zoznam objektov s nasledovnou štruktúrou, ktoré reprezentujú nájdené inzeráty pre vozidlo:
  - **url:** unikátny odkaz na inzerát
  - **version:** verzia inzerátu
  - **created\_at:** čas vytvorenia inzerátu
  - **ad\_price:** cena uvedená v inzeráte
  - **sold:** dátum predaja alebo prázdna hodnota, ak ešte predané nebolo
  - **ad\_features:** vlastnosti vozidla získané z daného inzerátu
  - **parsed\_raw\_text:** znenie inzerátu
  - **raw\_html:** html obsah stránky s inzerátom

## 2.3 Zoznam priložených elektronických dokumentov

- **Príloha A - GUI Design Document** - Návrh rozhrania webovej stránky pre interakciu so systémom
- **Príloha B - Dokumentácia modulu Crawler** - Prehľad štruktúry modulu určeného na zber dát
- **Prílohy C-G - Exporty jednotlivých šprintov (1-5)**

### 3. Popis modulov systému

#### 3.1 Data Collection

Analýza	Produkt si vyžaduje, aby sme vždy pracovali s aktuálnymi dátami. Preto potrebujeme zabezpečiť neustály zber dát z viacerých portálov na predaj áut cez inzerát.
Návrh	Časť systému zodpovedná za zber dát sa skladá z viacerých komponentov: <ul style="list-style-type: none"><li>• Processing queue - udržiava odkazy na inzeráty, ktoré treba spracovať</li><li>• New URL Discovery - objavuje nové odkazy na inzeráty</li><li>• Data Updater - aktualizuje stav inzerátov v databáze</li><li>• Downloader - stiahne obsah stránky</li><li>• Parser - vytiahne zo stránky znenie inzerátu</li></ul>
Implementácia	Zber dát, teda crawler, je implementovaný ako mikroslužba volaná riadiacim modulom. Jazyk: Python

#### 3.2 Text Processing

Analýza	Pre spracovanie textu potrebujeme vhodnou metódou implementovaný model pre spracovanie textu v slovenčine. Modul Text processor bude do databázy ukladať nájdené črty, cez ktoré budeme môcť trénovať predikčný model.
Návrh	Pre spracovanie jazyka v slovenčine sú pre nás relevantné 2 možnosti: slovníková metóda a strojové učenie s použitím word2vec.
Implementácia	Text processing je implementovaný ako mikroslužba volaná riadiacim modulom. Jazyk: Python

#### 3.3 Training Center

Analýza	Model pre odhad cien musí vždy odrážať aktuálny stav trhu, preto musíme zabezpečiť, že tento model sa bude pravidelne učiť na aktuálnych dátach. Toto tréningovanie chceme spúšťať v pravidelných intervaloch.
---------	--

### 3.4 Computing Center

Analýza	Pre odhad cien je potrebný regresný model, ktorý pomocou strojového učenia dokáže tieto odhady generovať. Okrem toho potrebujeme iné modely, aby sme získali dáta pre vizualizácie zhrnuté v priloženom elektronickom dokumente.
Návrh	Natrénované modely máme uložené v pamäti, pričom ich tréningom získané váhy uchováame v databáze.
Implementácia	Výpočtové centrum je implementované ako mikroslužba volaná riadiacim modulom. Jazyk: Python

### 3.5 Web App

Analýza	Pre intuitívnu interakciu so systémom chceme implementovať používateľské rozhranie vo forme webovej stránky, ktorá poskytne prehľadnú časť, kam používateľ môže zadať samotné znenie alebo odkaz na inzerát vozidla. Potom mu taktiež musí prezentovať vypočítané dáta.
Návrh	Návrh rozhrania webovej stránky nájdete medzi priloženými elektronickými dokumentmi.

## 3.6 Nasadenie modulov

Pri tvorbe systému sme sa rozhodli pre použitie prostredia Docker z dôvodu jednoduchosti nasadenia mikroslužieb na server a taktiež aj na vytvorenie jednotného vývojového prostredia.

Pre nasadenie na serveri je potrebné nainštalovať **docker** a **docker-compose**. Následne treba skontrolovať, či je docker service zapnutý **systemctl status docker.service** a v prípade, že nie je, použiť **systemctl start docker.service**. Je nutné pridať používateľa, ktorý v ďalšom kroku bude spúšťať docker image do skupiny docker.

Pre samotné spustenie stačí prejsť do predicar\_manager priečinka, ktorý je stiahnuteľný z nášho github repozitára, a tam spustiť **docker-compose up**.