



RV-rental-app

Chmelík Martin

Email: chmelikmartin123@gmail.com

Github: <https://github.com/Bendy545>

Střední průmyslová škola elektrotechnická, Praha 2, Ječná 30

Školní projekt v rámci předmětu Programové vybavení

Vypracováno dne: 7.1. 2026

1. Úvod	3
2. Specifikace požadavků	3
2. 1. Popis uživatelů	3
2. 2. Funkční požadavky.....	3
2. 3. Nefunkční požadavky	4
3. Návrh databáze.....	5
4. Architektura aplikace.....	6
4. 1. Návrhové vzory.....	6
5. Běh aplikace.....	8
6. Import.....	10
6. 1. CSV.....	10
6. 2. JSON	11
7. Konfigurace programu	13
7. 1. Umístění a struktura	13
8. Instalace a spuštění	14
9. Chybové stavy	16
10.Knihovny	18
11.Licence.....	18
12.Závěr	18

1. Úvod

Cílem této práce bylo vytvořit aplikaci, která spravuje relační databázi. Aplikace slouží k evidenci karavanů, zákazníků a výpůjček uložených v relační databázi. Uživatel může přes grafické rozhraní pracovat s daty uložené v databázi.

2. Specifikace požadavků

2. 1. Popis uživatelů

- Administrátor/Operátor: má přístup ke všem funkcím systému (CRUD operace, reporty)

-

2. 2. Funkční požadavky

FR1 - Správa vozidel:

- Systém umožní evidovat obytná vozidla

FR2 - Správa zákazníků:

- Systém umožní evidenci zákazníků

FR3 - Vytvoření výpůjčky:

- Vložení dat do více tabulek najednou
- Operátor vytvoří novou výpůjčku, která spojuje konkrétního zákazníka a vozidlo. Během vytváření výpůjčky lze přidat doplňkové příslušenství

FR4 - Reporty (Agregace dat):

- Systém vygeneruje souhrnný report z několika tabulek

FR5 - Importy (CSV/JSON):

- Aplikace umožní hromadný import dat ze souboru CSV/JSON

2. 3. Nefunkční požadavky

NFR1 - Konfigurace:

- Aplikace musí být konfigurovatelná pomocí externího souboru, kde bude uloženo nastavení připojení k databázi

NFR2 - Validace vstupů:

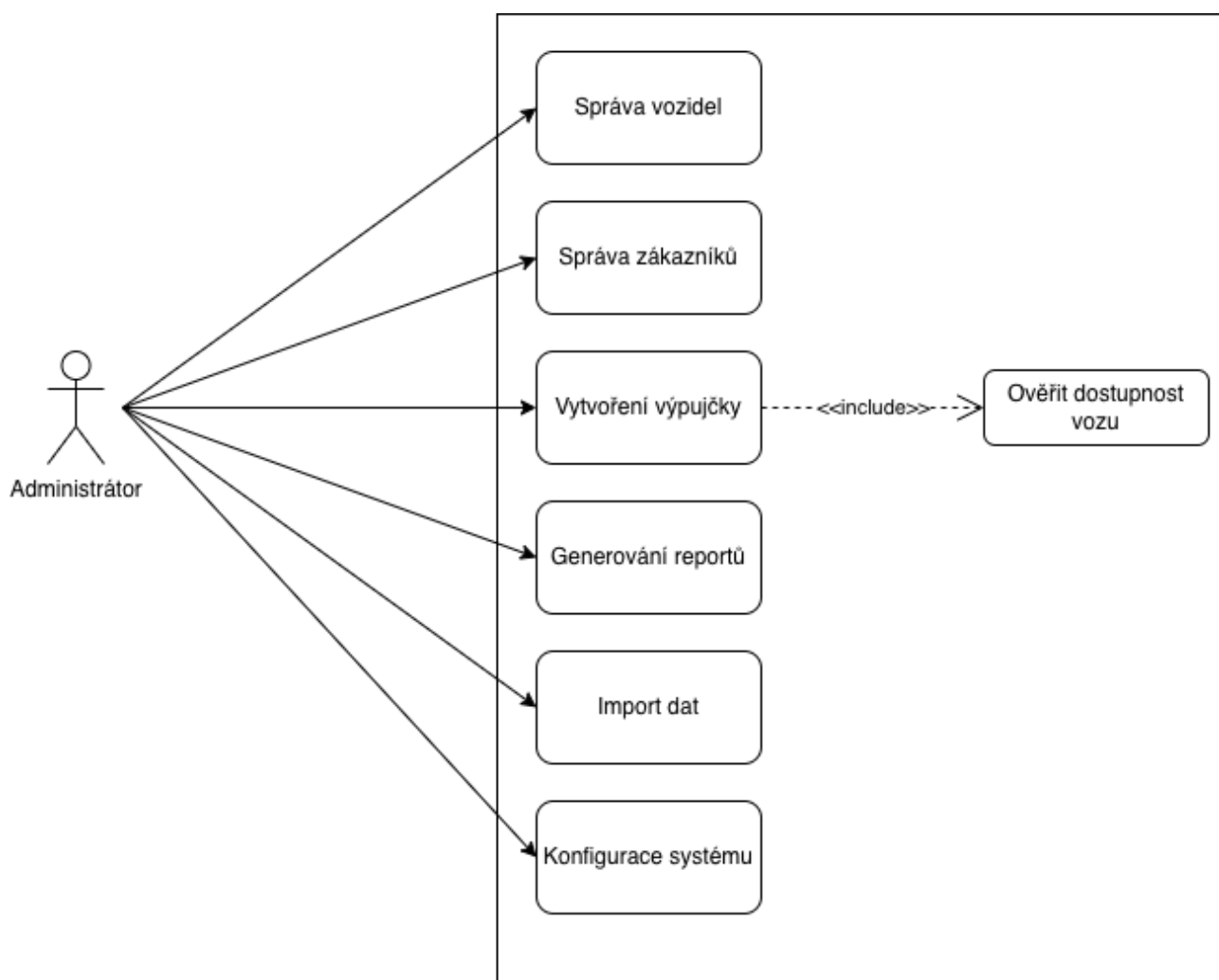
- Aplikace musí ošetřit veškeré uživatelské vstupy

NFR3 - Bezpečnost dat:

- Aplikace musí zajistit integritu dat

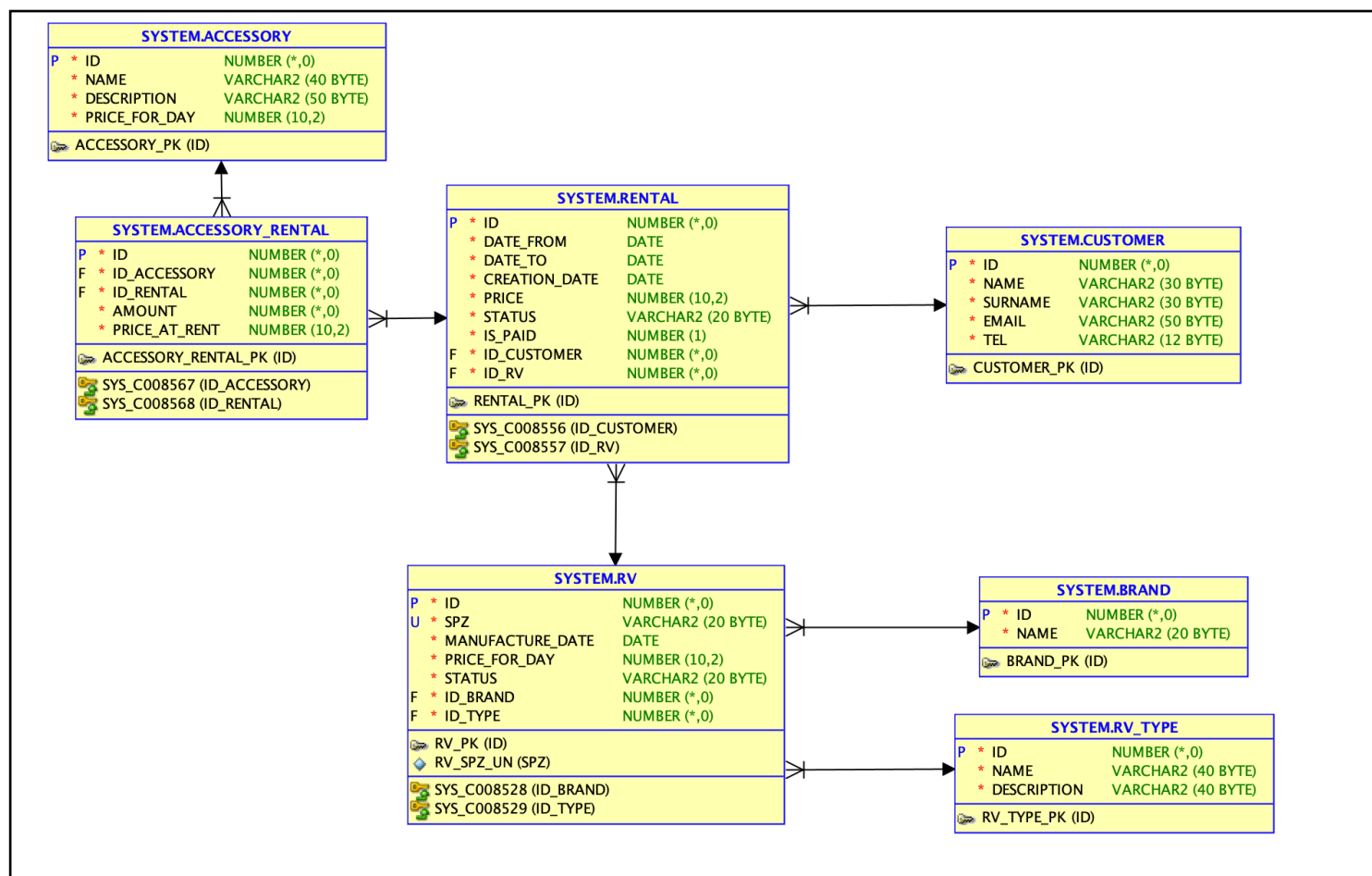
NFR4 - Uživatelské rozhraní:

- Aplikace musí být desktopová



Use case diagram

3. Návrh databáze



ER diagram

4. Architektura aplikace

Aplikace je postavena na tří vrstvě architektuře (Three-Tier Architecture), která odděluje prezentační vrstvu, business logiku a přístup k datům.

4. 1. Návrhové vzory

Aplikace využívá následující návrhové vzory:

- **DAO** (Data Access Object)
 - Každá tabulka má svůj vlastní DAO objekt
- **Singleton**
 - Zajišťuje jedinou instanci databázového připojení v celé aplikaci
- **Lazy initialization**
 - Singleton instance třídy Database se vytváří až při prvním použití

```
import cx_Oracle

class Database:
    _instance = None

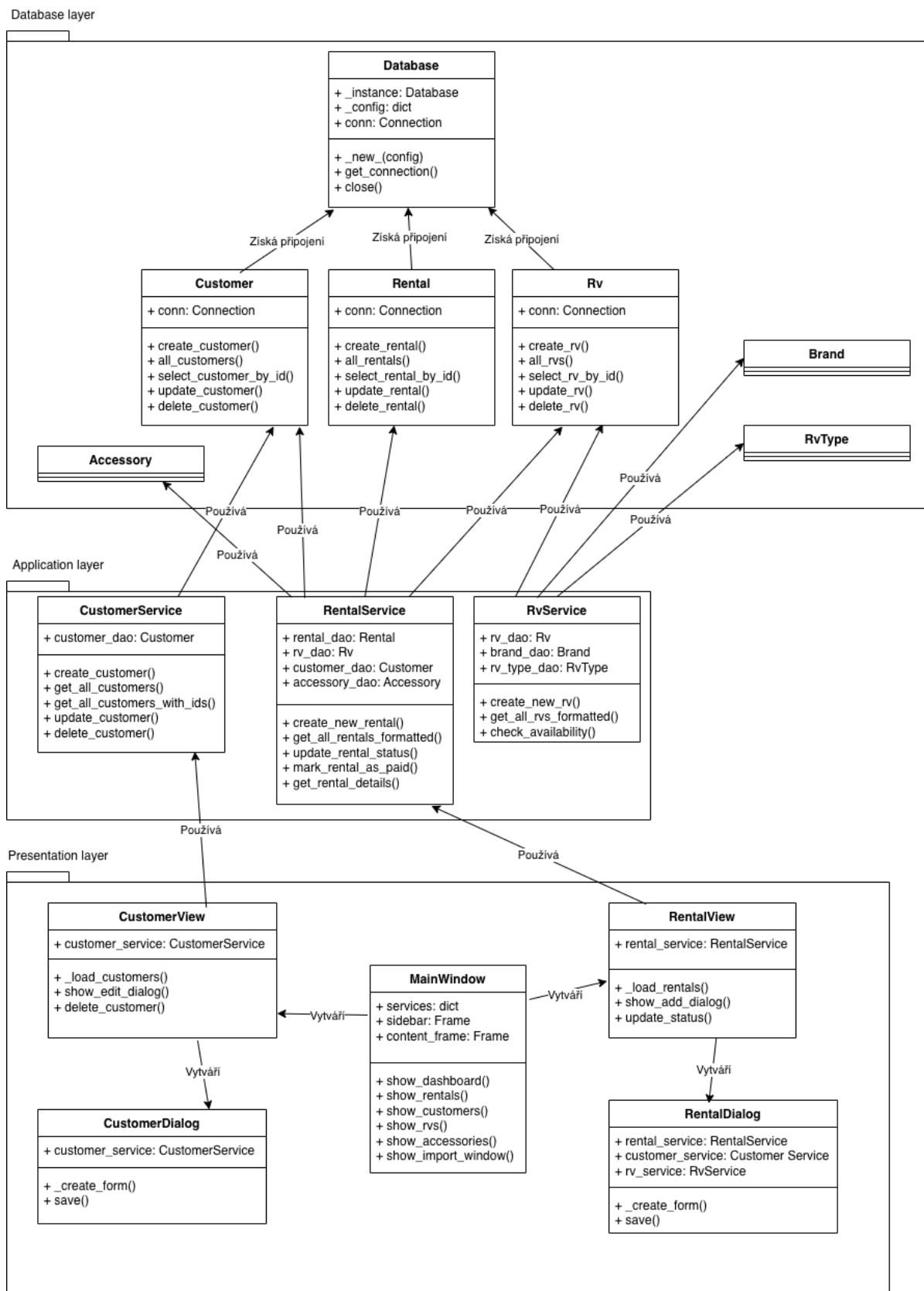
    def __new__(cls, config):
        if cls._instance is None:
            cls._instance = super(Database, cls).__new__(cls)
            cls._instance._config = config
            cls._instance.conn = None
            return cls._instance

    def _connect_if_needed(self):
        if self.conn is None:
            dsn = f"{self._config['host']}:{self._config['port']}/{self._config['service_name']}"
            self.conn = cx_Oracle.connect(
                user=self._config['user'],
                password=self._config['password'],
                dsn=dsn
            )
            print("Database connection Initialized.")

    def get_connection(self):
        self._connect_if_needed()
        return self.conn

    def close(self):
        if self.conn:
            self.conn.close()
            self.conn = None
            type(self)._instance = None
```

Kód zobrazující v jazyce Python třídu Database s použitím Singleton a Lazy Initialization vzorů

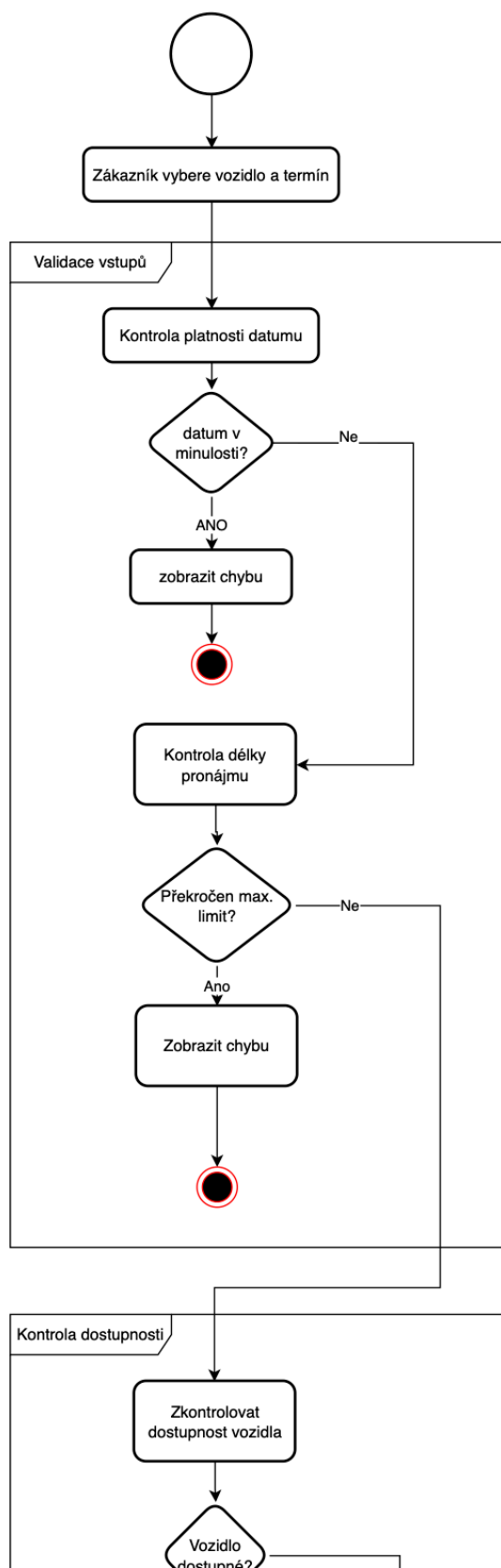


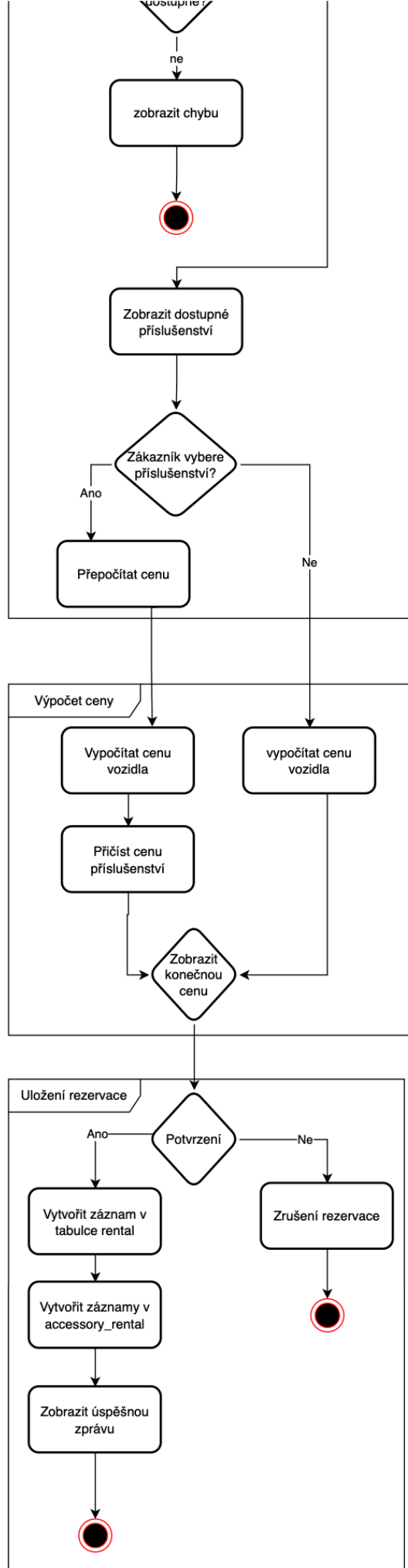
Class diagram aplikace

V diagramu jsou zobrazeny jen nejdůležitější třídy, které pracují s customer a rental.
Zbytek není zobrazen

5. Běh aplikace

Níže na dalších stránkách je popsán pomocí activity diagramu běh aplikace při rezervaci karavanu.





6. Import

Aplikace umožňuje import pomocí CSV i JSON.

6. 1. CSV

- Pomocí CSV můžeme importovat do customer nebo do brand.

Import do customer

Pořadí	Název sloupce	Typ	Povinnost	Popis	Omezení
1	name	String	POVINNÉ	Jméno zákazníka	1-30 znaků
2	surname	String	POVINNÉ	Příjmení zákazníka	1-30 znaků
3	email	String	POVINNÉ	Emailová adresa	Validní email, max 50 znaků
4	tel	String	POVINNÉ	Telefonní číslo	9-12 číslic

```
name,surname,email, tel
Jan,Novak,jan.novak@email.cz,777123456
Petra,Svobodova,petra.s@email.cz,777654321
Karel,Dvorak,karel.dvorak@email.cz,777987654
Lucie,Prochazkova,lucie.p@email.cz,777456789
Martin,Kratochvil,martin.k@email.cz,777111222
```

Příklad CSV souboru pro customer

Import do brand

Pořadí	Název sloupce	Typ	Povinnost	Popis	Omezení
1	name	String	POVINNÉ	Název značky	1-20 znaků, unikátní

```
name
Hymer
Adria
Knaus
Dethleffs
Burstner
Chausson
Sunlight
Carado
```

6. 2. JSON

- Pomocí JSON můžeme importovat do rv_type a do accessory

Import do accessory

Pole	Typ	Povinnost	Popis	Omezení
name	String	POVINNÉ	Název příslušenství	1-40 znaků
description	String	POVINNÉ	Popis příslušenství	1-50
price_for_day	Integer	POVINNÉ	Cena za den	> 0, max 2 desetinná místa

```
[
  {
    "name": "Camping table",
    "description": "Foldable camping table",
    "price_for_day": 5.00
  },
  {
    "name": "Bike rack",
    "description": "Rack for 2 bikes",
    "price_for_day": 8.50
  },
  {
    "name": "Gas grill",
    "description": "Portable gas grill",
    "price_for_day": 7.00
  },
  {
    "name": "Outdoor chairs",
    "description": "Set of 2 folding chairs",
    "price_for_day": 4.50
  }
]
```

Příklad JSON souboru pro accessory

Import do rv_type

Pole	Typ	Povinnost	Popis	Omezení
name	String	POVINNÉ	Název typu RV	1-40 znaků
description	String	POVINNÉ	Popis typu	1-40 znaků

```
[
  {
    "name": "Campervan",
    "description": "Small compact camper van"
  },
  {
    "name": "Alcove",
    "description": "Family RV with sleeping alcove"
  },
  {
    "name": "Integrated",
    "description": "Luxury fully integrated RV"
  },
  {
    "name": "Semi-integrated",
    "description": "Mid-class semi-integrated RV"
  }
]
```

Příklad JSON souboru pro rv_type

7. Konfigurace programu

- Aplikace používá konfigurační soubor pro nastavení připojení k databázi a dalších parametrů. Díky tomu nemusíme přistupovat ke zdrojovému kódu.

7. 1. Umístění a struktura

- Soubor se nachází na githubu pod jménem config.json nebo když si stáhnete .zip soubor z versions na githubu konkrétně Version2.zip

Klíč	Typ	Povinnost	Popis
user	String	POVINNÉ	Databázové uživatelské jméno
password	String	POVINNÉ	Heslo k databázovému účtu
host	String	POVINNÉ	Hostname
port	Integer	POVINNÉ	Port
service_name	String	POVINNÉ	Název oracle service

```
{
  "user": "system",
  "password": "student",
  "host": "localhost",
  "port": 1521,
  "service_name": "XE"
}
```

Příklad config.json souboru

8. Instalace a spuštění

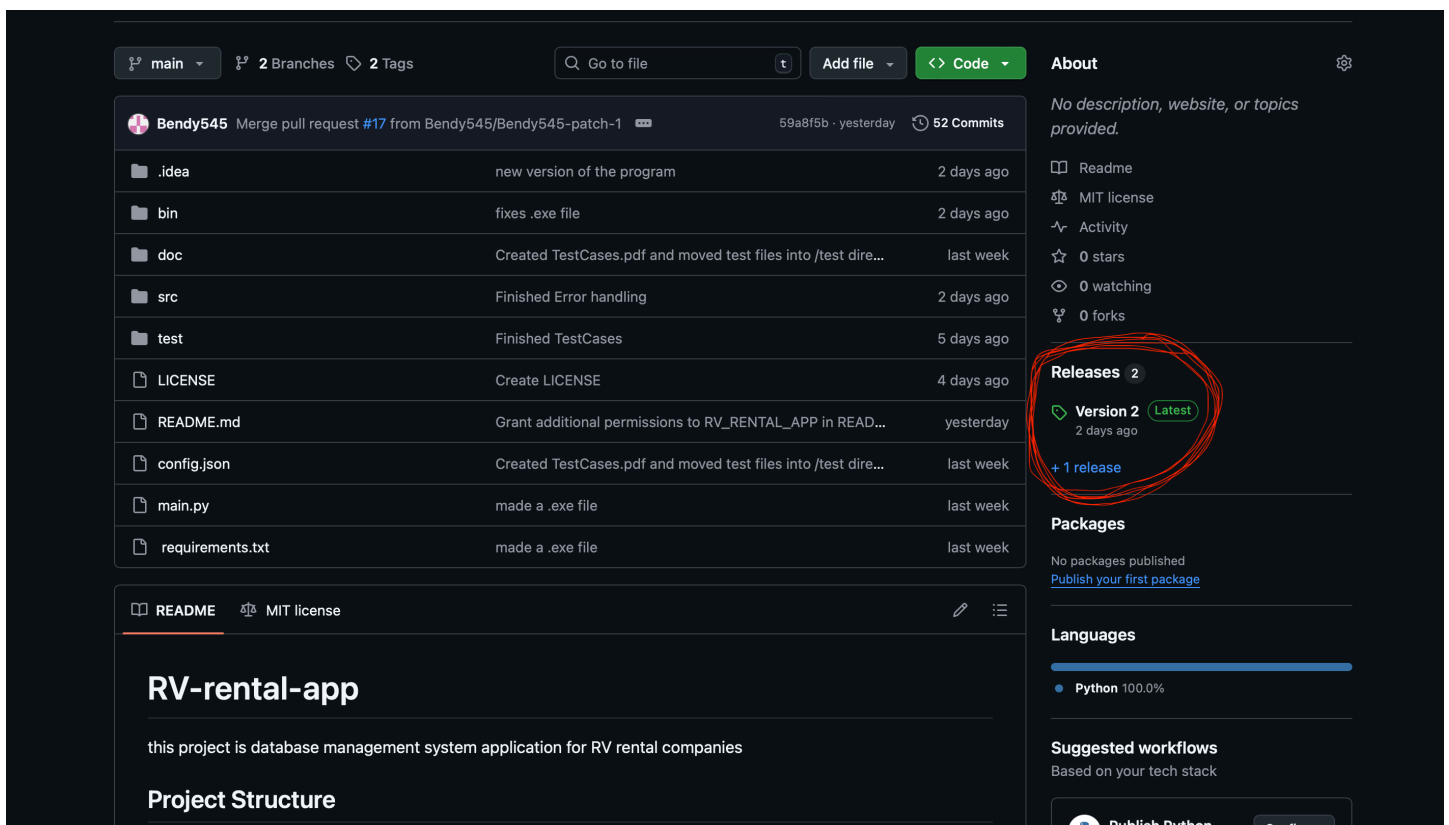
Aplikaci můžeme nainstalovat a spustit dvěma způsoby a to podle toho, jak ji chceme použít.

Klonování projektu

- Github: <https://github.com/Bendy545/RV-rental-app>
- Pokud si aplikaci/projekt naklonujeme do našeho vývojového prostředí, tak budeme postupovat podle README.md v repozitáři na githubu ke spuštění.

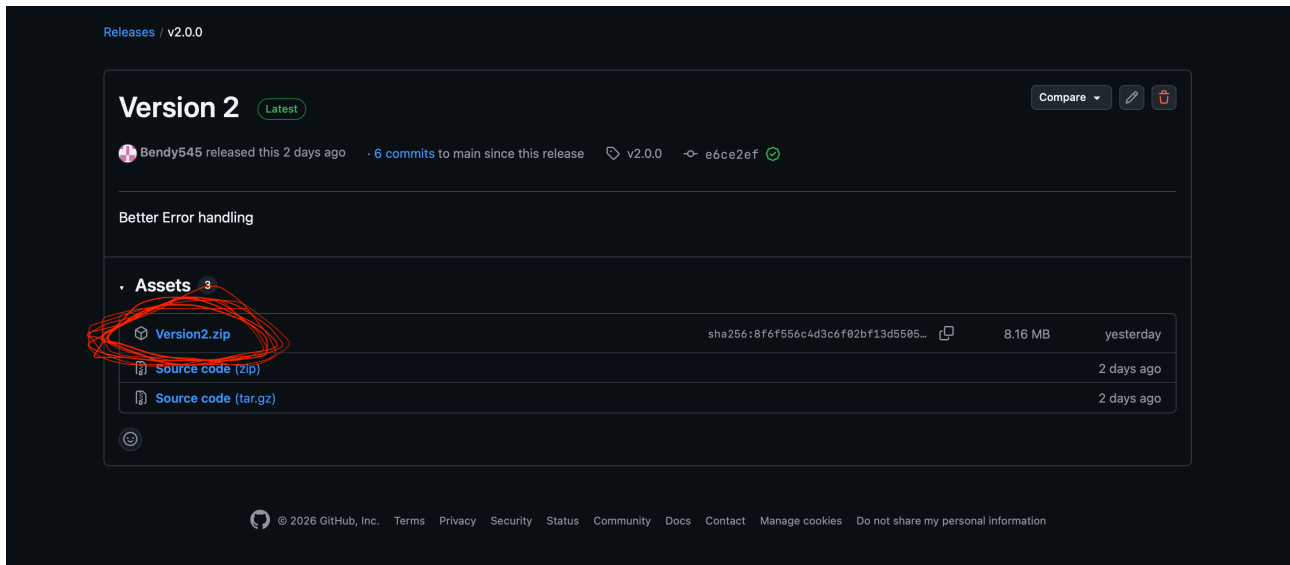
Instalace .exe souboru

- Github: <https://github.com/Bendy545/RV-rental-app>
- Na pravé straně repozitáře na githubu jsou Releases.



Obrázek znázorňující Releases

- Když klikneme na Version 2, tak se otevře nové okno.
- Tam si stáhneme Version2.zip.



Obrázek znázorňující Releases okno

- Když si Version2.zip rozbalíme tak uvidíme ve složce Version2 config.json, README.md a RvRentalApp.exe soubory.
- Pro spuštění postupujte podle README.md.
- Až vše nastavíte spustíte .exe soubor.

9. Chybové stavy

Aplikace má v každé vrstvě své výjimky což umožňuje přesné lokalizování problémů a uživatelsky přívětivé zprávy.

Každý modul v aplikaci má stejnou strukturu výjimek:

- {Module}ValidationError
- {Module}NotFoundError
- {Module}Database Error

VALIDATION ERROR (validační chyba)

- Kdy?:
 - Prázdná povinná pole
 - Nevalidní formát
 - Záporné nebo nulové hodnoty
 - Špatné datumové rozsahy
- Řešení:
 - přečtete hlášku - říká přesně, co je špatně
 - opravte pole
 - zkuste uložit znovu

NOT FOUND ERROR

- Kdy?:
 - Pokoušíte se upravit/smazat neexistující záznam
- Hláška obsahuje: „not found“
- Řešení:
 - Klikněte na tlačítko „Refresh“
 - Ověřte, že záznam existuje v seznamu

Chyby připojení k databázi:

- Hlášky typu „Database connection failed“
- Timeout při načítání dat
- Prázdné tabulky i přes existující data
- **Postup řešení:**
 - Zkontrolujte konfigurační soubor config.json
 - Ověřte že databáze běží

Databázové chyby:

- **ORA-00001** = pokus o vložení duplicitního záznamu
 - Hláška obsahuje „already exists“
 - Řešení: změňte hodnotu záznamu
- **ORA-02292** = pokus o smazání záznamu, který je referencován v jiné tabulce
 - Hláška obsahuje „referenced“, „used by“, „assigned to“, „child record“
 - Řešení: zjistěte kde je záznam referencován a buď referenci změňte nebo smažte
- **ORA-20001** = RV již pronajato v daném období

10. Knihovny

Jako jedinou externí knihovnu aplikace využívá je **cx_Oracle** pro práci s Oracle databází.

11.Licence

MIT License

Copyright (c) 2026 Martin Chmelík

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

12. Závěr

Díky tomuto projektu jsem získal zkušenosti s prací Oracle databází v jazyce Python. Dále jsem se naučil navrhnout a zkonstruovat třívrstvou architekturu, použít návrhové vzory a vytvořit grafické rozhraní pro aplikaci.