



Hall reservation DB app

Petr Valenta

Email: petr.valenta00@email.cz

GitHub: github.com/Fudr007

Střední průmyslová škola elektrotechnická, Praha 2, Ječná 30

Projekt k předmětu Programové vybavení

9.1. 2026

Obsah

1.	Úvod	3
2.	Specifikace požadavků	3
2.1	Uživatelé	3
2.2	Funkční požadavky	3
2.3	Nefunkční požadavky	4
3.	Návrh databáze.....	5
4.	Architektura aplikace	5
4.1	Návrhové vzory	5
5.	Běh programu	7
6.	Import	8
7.	Konfigurace	9
7.1	Struktura souboru	9
8.	Instalace a spuštění	10
8.1	Releases	10
8.2	Klonování/Stažení source kódu.....	10
9.	Chybové stavy	11
10.	Knihovny	11
11.	Licence.....	12
12.	Závěr	12

1. Úvod

Cíl toho projektu byl vytvořit aplikaci, která využívá relační databázi pro ukládání dat. Aplikace je určena pro správu rezervací sportovních hal a dat s tím souvisejících. Do databáze se ukládají se rezervace samotné, pro koho je rezervace vytvořena a jeho účet ze kterého se po té rezervace hradí, sportovní haly a různé povinné či doplňkové služby. Program se ovládá přes příkazovou řádku s jednoduchým UI.

2. Specifikace požadavků

2.1 Uživatelé

Jediným navrženým uživatelem je administrátor, který má přehled o všech dostupných datech a může provádět vkládací a dotazovací operace s databází.

2.2 Funkční požadavky

FR1 – Správa sportovních hal

- Systém umožní evidovat sportovní haly

FR2 – Správa zákazníků

- Systém umožní evidovat zákazníky

FR3 – Správa zákaznických účtů

- Systém umožní evidovat účty zákazníků

FR4 – Správa služeb

- Systém umožní evidovat povinné a doplňkové služby pro rezervaci

FR5 – Správa plateb

- Systém umožní evidovat platby jednotlivých rezervací

FR6 – Vytváření rezervace

- Systém umožní uložit rezervaci pro konkrétního zákazníka, dále uloží které haly jsou v rezervaci, na kdy a do kdy rezervace je, všechny povinné a vybrané nepovinné služby a sám spočítá kolik celá rezervace stojí. Následně lze danou rezervaci zaplatit.
- Data se ukládají do příslušných a vazebních tabulek

FR7 – Report

- Systém umožní vygenerovat souhrnný report důležitých/zajímavých dat

FR8 – Import dat

- Systém umožní hromadný import základních dat (haly, zákazníci, služby) ze souborů CSV do databáze

2.3 Nefunkční požadavky

NFR1 – Konfigurace

- Aplikace musí být pomocí konfiguračního souboru

NFR2 – Validace uživatelských vstupů

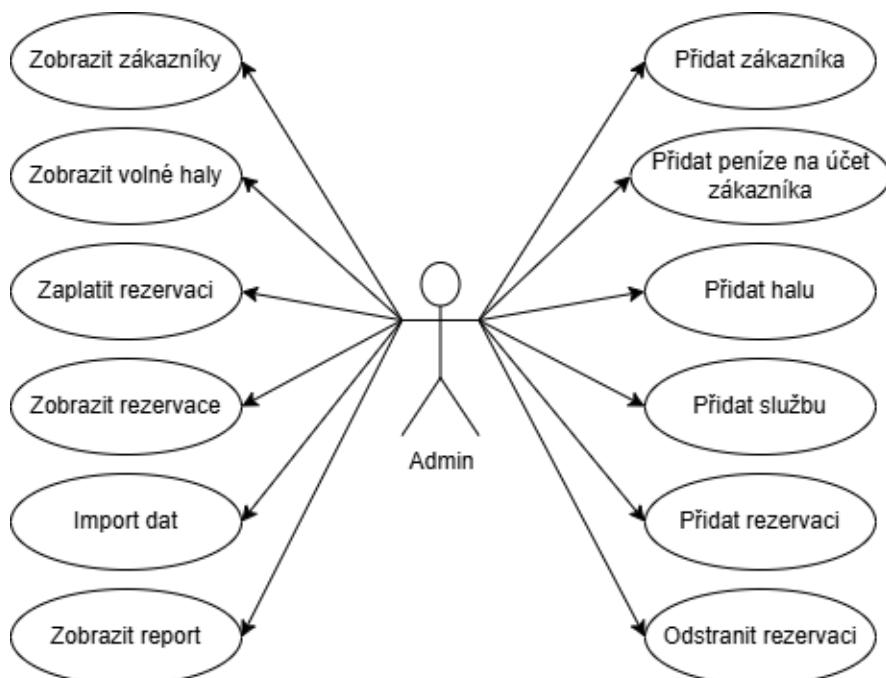
- Aplikace musí ohlídat všechny uživatelské vstupy a při případných problémech reagovat adekvátně

NFR3 – Bezpečnost dat

- Systém musí zajistit integritu dat

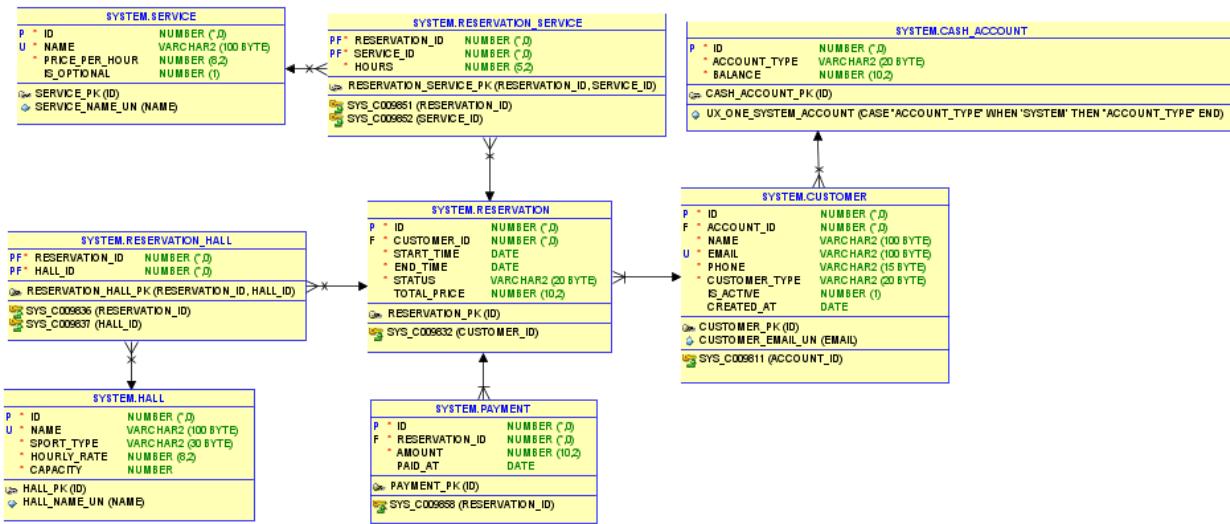
NFR4 – Uživatelské rozhraní

- Aplikace musí mít uživatelské rozhraní, přes které půjde ovládat její funkce



3. Návrh databáze

Databáze je navržená v Oracle XE databázi ve verzi 21c.



ER diagram databáze

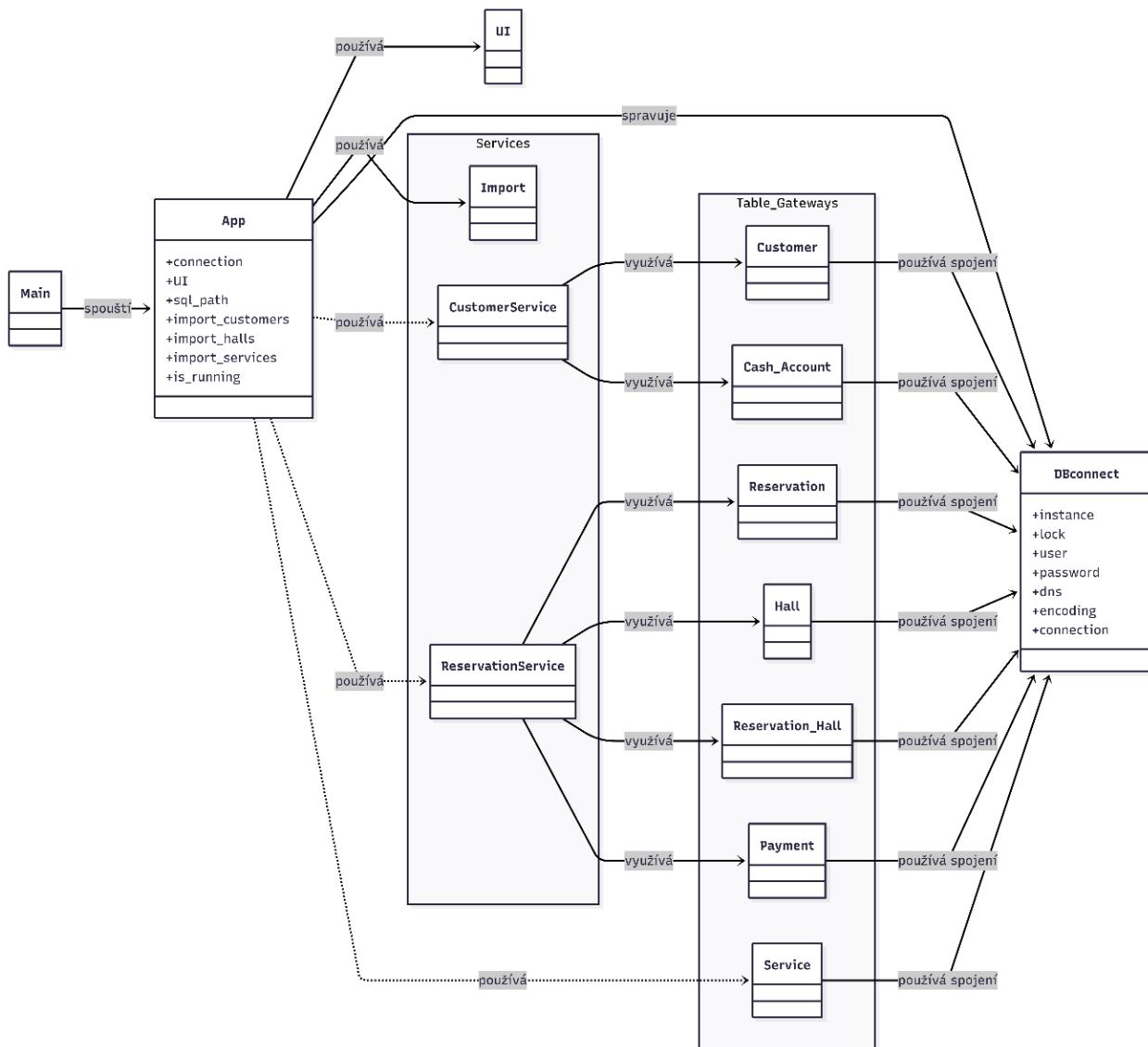
4. Architektura aplikace

Aplikace je postavená na základě Three-Tier Architecture, tzn má oddělený přístup k datům databáze od logiky průběhu programu a od uživatelského rozhraní

4.1 Návrhové vzory

Aplikace využívá následující návrhové vzory:

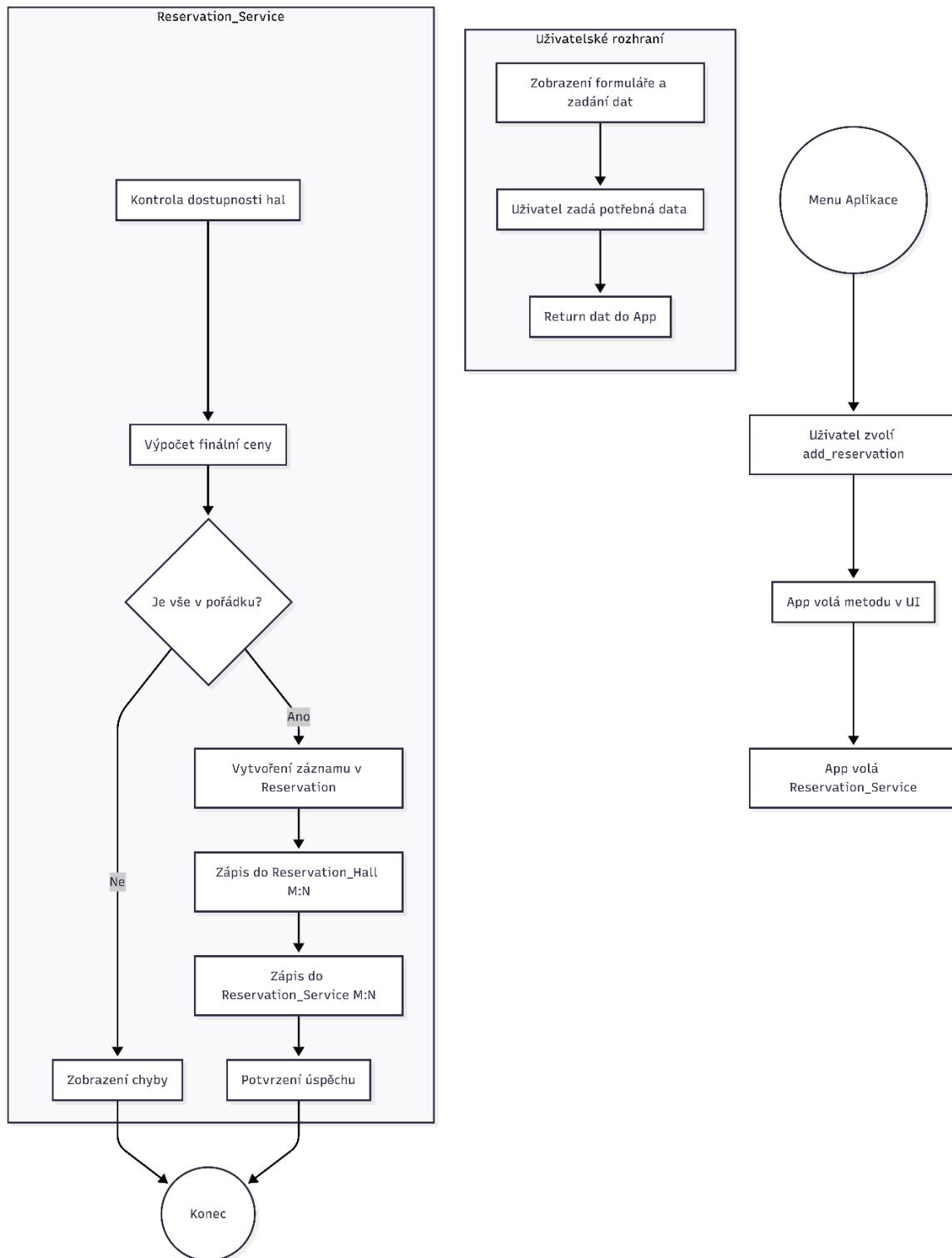
- Table Gateway – Každá tabulka má svoji třídu a CRUD operace
- Singleton – Aplikace zajišťuje že po celou dobu běhu aplikace je vytvořené a používá se pouze jedna instance připojení k databázi



Class diagram struktury aplikace

5. Běh programu

Níže je zobrazený diagram průběhu rezervace.



Activity diagram průběhu vytvoření rezervace

6. Import

Aplikace umožňuje import z CSV souborů. Po úspěšné inicializaci aplikace, kdy se nakonfigurují všechny cesty k souborům, včetně těch pro import, a vytvoří se databáze, se zobrazí menu možností programu. Číslo 11. na seznamu je Import data, toto číslo když zadáme měl by se spustit import dat ze souborů do databáze. Když se vše importuje správně systém zahlasí „Import completed successfully“ nebo "Import has already been done." pokud již import byl někdy dříve proveden se stejnými hodnotami.

Atribut	name	email	phone	customer_type	is_active
Příklad	Jan Novák	jnovak@email.cz	687529632	INDIVIDUAL	1
Povinný atribut	ano	ano	ano	ano	ne
Unikátní	ne	ano	ano	ne	ne
ENUM hodnoty				INDIVIDUAL / TEAM	1 / 0

Import do **customer** tabulky

Atribut	name	sport_type	hourly_rate	capacity
Příklad	Loko	HANDBALL	1000.0	60
Povinný atribut	ano	ano	ano	ano
Unikátní	ano	ne0	ne	ne
ENUM hodnoty		*		

*FOOTBALL, BASKETBALL, VOLLEYBALL, BADMINTON, HANDBALL, FLORBALL

Import do **hall** tabulky

Atribut	name	price_per_hour	is_optional
Příklad	Úklid	200.0	0
Povinný atribut	ano	ano	ne
Unikátní	ano	ne	ne

Import do **service** tabulky

7. Konfigurace

Aplikace se dá konfigurovat pomocí ini konfiguračního souboru. Do něj se ukládají údaje pro připojení k databázi a cesty k důležitým souborům, které databáze používá.

7.1 Struktura souboru

Soubor je rozdělen do dvou částí, tou první je sekce pro připojení pojmenovaná “database“. V té je jsou potřebné informace pro připojení k databázi.

```
[database]
user = SYSTEM
password = student
host = 127.0.0.1
port = 1521
service = xe
encoding = UTF-8
```

Struktura “database“ části a defaultní hodnoty

user = označuje username uživatele, na kterého chcete, aby se aplikace připojila

password = doplňte heslo pro daného uživatele

host = označuje ip adresu kde se nachází Oracle XE server databáze

port = je označení portu, na kterém server běží

service = typ Oracle databázového serveru

encoding = jak je zakódování SQL příkazů

Druhou částí jsou cesty k důležitým souborům, které jsou potřebné pro běh databáze. Tato část je pojmenovaná “path“.

```
[path]
db_code = db.sql
import_customer = Import/customer.csv
import_service = Import/service.csv
import_hall = Import/hall.csv
```

Struktura “path“ části a defaultní hodnoty

8. Instalace a spuštění

Aplikaci lze nainstalovat a následně spustit dvěma způsoby přes GitHub na tomto odkaze: <https://github.com/Fudr007/DBhallReservationD1>, přes Releases a přes naklonování/stažení source kódu.

8.1 Releases

První způsob před Releases na GitHub stránce tohoto projektu.

Postup:

- Na GitHub stránce tohoto projektu klikněte na nejnovější release tohoto projektu, označený Latest.
- Zde klikněte na **HallReservationDB.zip** a tím tento zip soubor stáhněte.
- Po stažení rozbalte tento soubor do zvoleného místa na disku.
- Po otevření složky s rozbaleným zip souborem uvidíte exe soubor pro spuštění aplikace a konfigurační ini soubor pro konfiguraci
- Nejprve nakonfigurujte konfigurační soubor podle kapitoly 7.1
- Poté stačí spustit main.exe soubor, pokud se vše provede správně tak by se aplikace měla spustit a zobrazit výchozí menu, pokud ne aplikace by měla sdělit co se stalo špatně

8.2 Klonování/Stažení source kódu

Druhým způsobem, jak tento projekt stáhnout a následně spustit je přes naklonování nebo stažení source kódu

Postup:

- Na GitHub stránce tohoto projektu klikněte na zelené tlačítko **Code**.
- Dále stiskněte **Download ZIP** tím stáhnete zip soubor se source kódem.
- Rozbalte zip soubor do vámi vybraného místa na disku.
- Poté otevřete rozbalenou složku ve vašem oblíbeném IDE
- Nakonfigurujte konfigurační soubor config.ini podle kapitoly 7.1
- Dále doinstalujte knihovnu pro práci s Oracle databází přes příkazovou řádku, spusťte toto: **pip install cx_Oracle**
- Vyberte Python interpreter, ideálně Python 3.13 nebo kompatibilní
- Následně stačí spustit kód v souboru main.py, pokud se vše provede správně tak by se aplikace měla spustit a zobrazit výchozí menu, pokud ne aplikace by měla sdělit co se stalo špatně

9. Chybové stavy

Aplikace rozděluje výjimky primárně na chyby v konfiguraci programu. Dále na chyby v UI, ty jsou způsobeny chybným vstupem uživatele a jsou označeny jako **Invalid input**. Dalším typem jsou chyby v databázi, ty vyhazuje přístup k databázi, když se pokusí provést nějakou akci. Vždy je zde uvedeno, z jaké části databáze tato chyba přichází (např. Customer) a co se stalo špatně.

Základní typy:

- “Jméno objektu” with duplicate data in database = označuje že se již v databázi nachází objekt s atributem, který nesmí být duplicitní a vy se jej snažíte přidat
- Invalid values = označuje že se snažíte vložit do databáze data, která nesplňují požadavky databáze (např požadavek na telefonní číslo je, aby bylo 9 míst dlouhé a obsahovalo pouze číslice)
- Cannot insert NULL values = označuje že se do databáze snažíte přidat NULL hodnotu, tam kde být nesmí
- Too large value = označuje že se snažíte do databáze vložit moc velkou hodnotu

10. Knihovny

Aplikace používá pouze externí knihovnu **cx_Oracle** pro práci s Oracle databází.

11. Licence

MIT License

Copyright (c) 2026 Petr Valenta

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

Při tvorbě tohoto projektu jsem se naučil pracovat s Oracle XE databází připojenou k vlastní Python aplikaci. Dále jsem si vyzkoušel v "praxi" použití návrhových vzorů a návrh třívrstvé architektury. Případné zlepšení bych viděl primárně ve tvorbě lepšího grafického rozhraní. Případně přidání dalších užitečných funkcí, které by koncový uživatel potřeboval. Myslím si však že aplikace v tomto stavu funguje dobře.