

**Softwarový robot pro Discord server**

**Ročníkový projekt**

|  |  |
| --- | --- |
| *Studijní program:* | B2646 – Informační technologie |
| *Studijní obor:* | 1802R007 – Informační technologie |
| *Autor práce:* | **Tomáš Novotný** |
| *Vedoucí práce:* | Ing. Lukáš Matějů, Ph.D. |

Anotace

Text anotace.

Annotation

Text of annotation.

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Poděkování

Děkuji všem, kteří mi pomohli.

Obsah

[Seznam zkratek 6](#_Toc112616523)

[1 Použité technologie 8](#_Toc112616524)

[1.1 Microsoft Visual Studio 8](#_Toc112616525)

[1.2 .NET 8](#_Toc112616526)

[1.3 Entity Framework Core 8](#_Toc112616527)

[1.4 DSharpPlus 9](#_Toc112616528)

[1.5 SQLite 9](#_Toc112616529)

[1.6 Discord 9](#_Toc112616530)

[1.7 DB Browser for SQLite 10](#_Toc112616531)

[1.8 Pokémon GO 11](#_Toc112616532)

[2 Metodika aplikace 11](#_Toc112616533)

[2.1 Softwarový robot 12](#_Toc112616534)

[2.2 Vlastnosti 12](#_Toc112616535)

[2.2.1 Pravidla umístění bodů 12](#_Toc112616536)

[2.3 13](#_Toc112616537)

[3 Implementace aplikace 13](#_Toc112616538)

[3.1.1 Databáze 13](#_Toc112616539)

[3.1.2 Příkazy bota 13](#_Toc112616540)

[3.1.2.1 Nahrávání dat 13](#_Toc112616541)

[3.1.2.2 Kontrola dat 13](#_Toc112616542)

[3.1.2.3 Změna typu bodu 13](#_Toc112616543)

[3.1.2.4 Změna názvu bodu 13](#_Toc112616544)

[3.1.2.5 Procházení bodů 13](#_Toc112616545)

[3.1.2.6 Hlášení raidů 13](#_Toc112616546)

[3.1.2.7 Nastavení srazu 13](#_Toc112616547)

# Seznam zkratek

TUL Technická univerzita v Liberci

FM Fakulta mechatroniky, informatiky a mezioborových studií Technické

univerzity v Liberci

API Application Programming Interface

PvP Player vs Player (hráč proti hráči)

PvE Player Versus Environment (hráč proti hře)

Úvod

Text úvodu.

# Použité technologie

## Microsoft Visual Studio

Microsoft Visual Studio (dále Visual Studio) je vývojové prostředí vytvořené společností Microsoft. Využívá se pro vývoj konzolových, webových, mobilních aplikací nebo webových stránek. Nejnovější verze vydání Visual Studio 2022 obsahuje vylepšenou verzi IntelliSense, kterou je výkonná sada nástrojů IntelliCode. Ta na základě již napsaného kódu, strojového učení a získané znalosti z půl miliónu kódových vzorů dokáže vyhodnotit, co bude chtít programátor napsat a vyplnit na něj celý řádek, případně celý cyklus. Visual Studio integruje možnost používat Git pro klonování kódu na cloudové úložiště. Díky chytrému zabudovanému ladění lze kód jednoduše procházet a odhalit chyby. Existují 3 verze: Community pro studenty a menší vývojáře, Professional a Enterprise pro náročnější projekty.[[1]](#footnote-1)

## .NET

.NET je bezplatná, open source vývojová platforma, která je určena pro vytváření různých aplikací. Jedná se o spojení programovacích jazyků a knihoven a CLR (Common Language Runtime) což je běhové prostředí, které zajišťuje běh a kompilaci aplikací. .NET podporuje programovací jazyky C# (zvolen pro tuto práci), F# a Visual Basic. C# je jednoduchý, moderní, objektově orientovaný a typově bezpečný jazyk. Pomocí .NETu lze vyvíjet aplikace pro web, osobní počítače, mobilní telefony, hry či cloud. Jde se o multiplatformní technologii, s kterou můžeme stejný kód spouštět na systému Windows, macOS, Linux, Android nebo iOS. Aplikace využívá nejnovější verzi .NET 6.0.[[2]](#footnote-2)[[3]](#footnote-3)

## Entity Framework Core

Entity Framework Core (dále EF Core) zajišťuje přenos dat mezi aplikací s databází. Je součástí .NET a umožňuje nám pracovat s entitami a atributy jako s třídami a vlastnostmi. Usnadňuje sledování změn, zavoláním metody SaveChanges se provede automatická aktualizace přidaných, změněných, odebraných objektů a následně tyto změny promítne v databázi. Po definování tříd entit můžeme vytvořit migraci, ze které se vygeneruje počáteční sada tabulek.[[4]](#footnote-4) EF Core pracuje s dotazovacím jazykem LINQ (Language-Integrated Query), který je kompatibilní s různými typy zdrojů a formáty dat. Programátor tedy pracuje se stejnými objekty bez ohledu na cílovou databázi.[[5]](#footnote-5)

## DSharpPlus

DSharpPlus je neoficiální multiplatformní .NET knihovna, tzv. „wrapper“, založený ze zrušeného projektu DiscordSharp, který se používá k propojení s oficiálním Discord API. Je přepsaná tak, aby odpovídala standardům API a zaměřuje se na širší škálu implementace .NETu. Ověření je zajištěno pomocí soukromého klíče (token), který pro přihlášení vygeneruje Discord API a je uložen na straně klienta. Aplikace využívá verzi 3.2.3.[[6]](#footnote-6)

## SQLite

SQLite je nejpoužívanější knihovna nástrojů, databázový systém bez použití serveru nebo služby. Jednotlivé databáze se ukládají jako soubory. Výhodou je malá velikost souborů (jednotky MB), rychlé načítání, žádná nebo jen minimální konfigurace, snadná přenositelnost, multiplatformní použití. SQLite je transakční, pokud se tedy transakci nepovede úspěšně dokončit např. pádem programu nebo operačního systému, tak se neprovede a databáze se vrátí do původního validního stavu. Přístup k databázi je umožněn z více instancí. Nevýhodou je naopak nemožnost čtení a zapisování zároveň, protože při ukládání informací se databáze uzamkne. Postrádá některé datové typy, které lze nahradit základními (Bool → Int nebo DateTime → String). Není možná správa oprávnění a uživatelů na úrovni databáze, ale je možné zabezpečit databázový soubor pomocí operačního systému. Podle vývojářů je záměr podporovat SQLite do roku 2050.[[7]](#footnote-7)

## Discord

Komunikační platforma Discord bezplatná aplikace pro hlasovou, video a textovou komunikaci. Probíhá na jednotlivých serverech, které jsou většinou zaměřeny na jedno téma nebo zastřešují určitou komunitu lidí např. hry *Pokémon GO* nebo známé hudební skupiny. Přístupnost serveru může být veřejná nebo soukromá. Pouze registrovaní uživatelé se mohou připojit, a to na několik serverů na jednou. Výměna informací, probíhá v hlasových a textových kanálech. Uživatelé mohou dostat určité role, které specifikují, jaké budou mít pravomoci na serveru. Jde např. o omezený přístup k jednotlivým hlasovým a textovým kanálům, využívání kanálu pouze pro čtení nebo možnost zablokování ostatních uživatelů. Softwarový robot (déle bot) je testován na soukromém serveru autora práce, který je částečnou kopií hlavního serveru *Pogo Raid Liberec*, na který by měl být bot aplikován. Pro jeho práci je nutné mu udělit nejvyšší práva, aby měl přístup ke všem kanálům a mohl do nich psát zprávy či blokovat uživatele a odepsat jim role. Díky komplexnosti Discordu je testovací server plně dostačující a výsledky na něm lze prezentovat jako konečné.[[8]](#footnote-8)

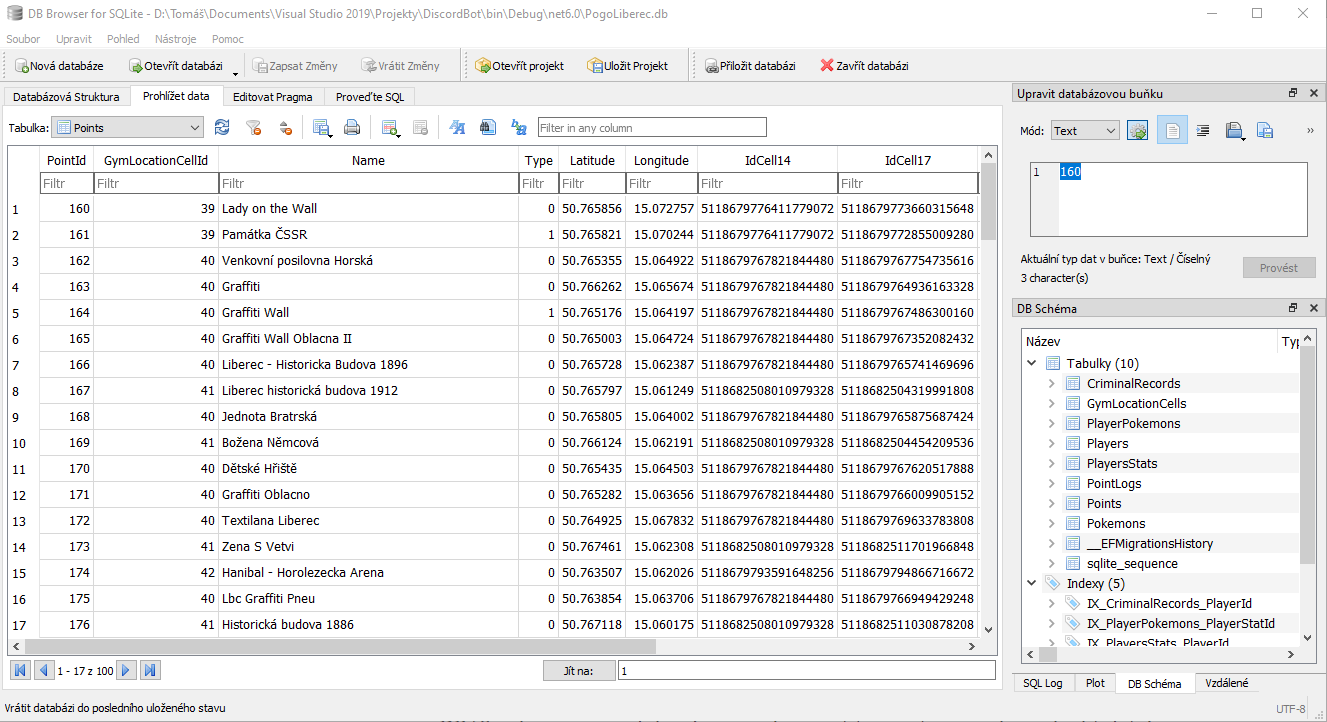
Obsah obrázku text, monitor, snímek obrazovky, obrazovka

Popis byl vytvořen automaticky

Obrázek 1 Grafické rozhraní testovacího serveru na Discordu

## DB Browser for SQLite

DB Browser for SQLite (dále DB Browser) je open source nástroj pro vytváření, navrhování, čtení a editaci databázových souborů kompatibilních s SQLite. S jeho pomocí můžeme vyhledávat uložené záznamy a kontrolovat jejich správné zapsání do souboru. Software zároveň umožňuje jednoduchý import a export tabulek do souborů formátu txt nebo csv. Aplikace je intuitivní a přehledná.[[9]](#footnote-9)



Obrázek 2 Grafické rozhraní programu DB Browser

## Pokémon GO

Oblíbená mobilní hra, jejíž cílem je nasbírat všechny pokémony. Ty se objevují po celé mapě, nejčastěji v parcích, na pěších cestách nebo u pokéstopů a gymů, kde hráči sbírají pokébally a berries. Své pokémony si hráč může za pomocí nasbíraného stardustu a candies vyvíjet a vylepšovat. Stardust a candies hráč získává za chytání, candies i za prodej. Zájmové body ve hře reflektují místa v reálném světe (např. památník, dětské hřiště, historické či neobvyklé stavby, zdobené výklenky atd.). Na gymech se může objevit (spawnout) raid s určitým pokémonem (boss). Nejprve se odhalí vajíčko, ze kterého se v určitý čas vyklube boss. Na jeho poražení a následné získání je potřeba 1 až více hráčů (podle obtížnosti), kteří se musí domluvit na čase, dojít na místo (případně dostat pozvánku) a společné bosse porazit. Z pokéstopů hráč dostává úkoly, z kterých může dostat další pokébally, berries nebo přímo spawn pokémona. Hráči mohou sílu svých nasbíraných pokémonů změřit v PvP nebo PvE souboji. Pokud se hráči osobně setkají, mohou své pokémony vyměňovat. Hráči jsou rozděleni do tří týmů (modrý Mystic, červený Valor, žlutý Instinct). Hra je vyvíjena společností Niantic.

# Metodika aplikace

## Softwarový robot

## Vlastnosti

### Ingress

Mobilní hra podobná známé hře Pokémon GO. Spravuje ji stejný vývojář, Niantic. Ve hře jsou 2 týmy neboli frakce (zelení Osvícení a modrý Odpor), kteří bojují proti sobě. Místo pokéstopů a gymů hráči zabírají portály pro svůj tým a snaží se je spojit do geometrického útvaru. Podle velikosti se poté týmu a hráči připisují body.

### S2 buňky

S2 buňky fungují pomocí matematických výpočtů, které umožňují převádět 3D tvar Země do 2D geometrie. Technologii vynalezl v roce 2005 inženýr Googlu Eric Veachem. Každá buňka má jedinečné 64bitové číslo, které umožňuje vývojářům jednoduchou identifikaci míst na Zemi. Mají různé úrovně podle velikosti (vyšší úroveň = menší velikost). Jsou hierarchické, větší buňku lze rozdělit na menší. Největší buňka úrovně 0 má rozměr 85 011 012 km², naopak nejmenší buňka úrovně 30 má rozlohu 0,48 cm² až 0,93 cm².[[10]](#footnote-10)

Pokémon GO využívá tyto buňky v několika úrovních. Buňky úrovně 10 se používají k určení polohy při chycení pokémona. Úrovně 10 a 11 slouží k určení počasí ve hře. Úroveň 14 se používají k určení počtu gymů v dané buňce. Body, které jsou v úrovni 16 kolem aktuální polohy hráče se vykreslují ve hře. Buňky úrovně 17 se používají k určení, zda se bod objeví jen ve hře Ingress nebo i ve hře Pokémon GO.[[11]](#footnote-11)

### Pravidla zobrazení bodů ve hře

Obsah obrázku bílá tabule

Popis byl vytvořen automatickyPokéstopy a gymy mají určitá pravidla, podle kterých se ve hře objevují. Většina bodů (portálů) vychází ze hry Ingress. Dříve se nové body mohly přidávat pouze přes něj. Před dvěma lety se umožnila možnost přidávat body i ze hry Pokémon GO.

Obrázek 3 Mapa znázorňující zájmové body ve hře Pokémon GO a úrovně 14 a 17 S2 buněk

V buňce úrovně 17 může být ve hře Pokémon GO pouze jeden pokéstop nebo gym nehledě na počet portálů ve hře Ingress. Na obr. č. 3 jsou znázorněny světle zelenou barvou. V buňce úrovně 14, na obr. č. 3 znázorněny silnou zelenou čarou. Podle počtu zaplnění buněk úrovně 14, kterých je 64, se vypočítává počet gymů. Přesněji řečeno, při zaplnění 2-5 buněk úrovně 17 (na obr. č 3 znázorněno šedým vyplněním) se jeden z bodů změní na gym. Při vyplnění 6-19 buněk 2 gymy. Pokud bude počet bodů mezi 20-34, v buňce budou 3 gymy. Při 35 a více bodech 4 gymy.[[12]](#footnote-12) Na obr. č. 3 je vidět, že v buňce úrovně 14 jsou vyplněná 3 políčka, tudíž se jeden z nich změní na gym.

### Získání dat o bodech

Obsah obrázku stůl

Popis byl vytvořen automatickyIngress ani Pokémon GO nemá žádné oficiální API, ze kterého by se dala získat data o bodech ve hře. Existují webové stránky, kde mohou lidé ručně body přidávat. Nevýhodou těchto stránek je náročné ověřování informací a neúplnost dat. Pro řešení se nabízí oficiální Ingress mapa, která je dostupná pro přihlášené hráče na adrese [*www.intel.ingress.com*](http://www.intel.ingress.com). V ní jsou vidět všechny portály ve hře a jaký tým je má pod kontrolou. Data ve výchozím stavu stále nejdou exportovat. Je nutné použít doplněk do prohlížeče *Ingress Inter Total Conversion* (dále IITC). Pomocí něj lze jednotlivé portály označit za pokéstopy nebo gymy a následně exportovat do souboru .csv nebo .json. Pro aplikaci je využit soubor .csv, který obsahuje 3 nejpodstatnější informace – jméno a zeměpisná délka a šířka.

Obrázek 4 Ukázka exportovaného souboru pomocí rozšíření IITC

### Zpracování dat o bodech

Získaná data bot zpracuje a vytvoří databázi bodů, ke kterým přidá nové informace, které lze dopočítat nebo přidat ručně. Ze souřadnic lze vypočítat unikátní číslo podle potřebné úrovně (více v kapitole 2.2.2), automaticky se vygeneruje datum a čas vytvoření a aktualizace. Pokud vznikne konflikt, přidá se příznak nutné ruční kontroly a poznámka. Ručně se musí vyplnit, zda se jedná o pokéstop nebo gym.

### Vznik konfliktů

Při vkládání bodů do databáze může dojít ke konfliktu, který není možné vyřešit automaticky, je nutný ruční zásah uživatele. Ve hře Pokémon GO totiž nejsou názvy pokéstopů a gymů unikátní. Určité místo je častokrát jedinečné pouze u blízkém okolí a lze předpokládat, že se může objevit i na jiné časti města/státu. Unikátnost je zaručena souřadnicemi a již zmíněními S2 buňkami úrovně 17 (kapitola 2.2.2). Body před zapsáním do hry prochází procesem kontroly, kde uživatelé kontrolují, zda se nejedná o duplikát. Protože body vytvářejí a schvalují lidé, tak často nemusí být názvy gramaticky správně (např. chybná interpunkce nebo špatná velká písmena). Problémem jsou obecné názvy bez jakékoli personifikace ke konkrétnímu místu. Nejčastější názvy jsou: Dětské hřiště, Graffiti, Památník, Pískoviště, Venkovní posilovna.

Bot musí vzniklé konflikty odchytit, odhadnout o jaký konflikt se jedná a zapsat do databáze pro pozdější ruční kontrolu. Pro případy, kdy se nepovede konflikt odhalit, musí bot umět pomocí příkazu změnit název, souřadnice nebo typ bodu a aktualizovat databázi. Pro kontrolu zapsaných bodů by měl umět vypsat list bodů v databázi po napsání názvu nebo části názvu.

### Hlášení raidů

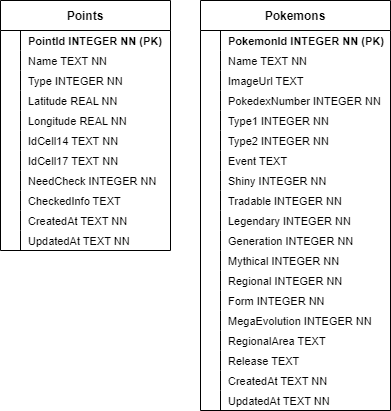
Uživatel napíše do příslušného kanálu *#hlášení-raidů* název gymu, název pokémona a čas a jiný kanál, kde bude probíhat domluva mezi ostatním hráči ve který čas se uskuteční sraz. Hráči tyto informace mohou zapsat v jakémkoli pořadí a jakýmkoli oddělovačem. Příklad: čas – pokémon – místo – kanál nebo místo – čas – pokémon – kanál. Název pokémona může být nahrazen emotikonem. Čas může reprezentovat čas líhnutí (začátku raidu) nebo obvykle počet minut do konce. Po úspěšné domluvě se do kanálu *#domluvné-srazy* napíše místo, pokémon (emotikon) a čas srazu.

Bot by měl umět rozpoznat zprávu hlášení raidu a zobrazit jí v přehledné podobě. Uživatelům umožnit pomocí emotikonů hlasovat o účasti a přehledně tak zobrazit počet zájemců. Následně opět pomocí emotikonu vytvořit sraz a vypsat přehlednou zprávu.

### Možné budoucí funkce

# Implementace aplikace

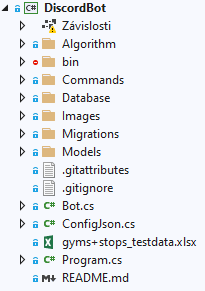
## Databáze

V databázi potřebujeme ukládat informace o bodech a pokémonech. Každá entita má primární klíč (Id) a atribut vytvoření a aktualizace záznamu. Dalšími atributy v tabulce *Points* jsou: název bodu, typ (portál, pokéstop, gym), zeměpisná délka a šířka, identifikátor buňky S2 úrovně 14 a 17, položku, nutnost kontroly záznamu a informace o chybě. Entita *Pokemons* obsahuje všechny potřebné informace o pokémonech viz Obrázek 5. I když pro využití v této práci nám dostačuje pouze název a odkaz na obrázek, je dobré mít další informace již uložené pro možné budoucí rozšíření.

Obrázek 5 Schéma databáze

## Souborová struktura

Soubory programu jsou rozděleny do několika kategorií

Složka Algorithm obsahuje třídy pro práci s daty. Složka bin slouží pro zdrojové soubory softwaru v provozu. Ve složce Commands se nachází třídy obsluhující příkazy bota. Ve složce Database je context, který spojuje kód s databází. Ve složce Images jsou uložené obrázky pokémonů pro možné pozdější využití. Složka Migrations obsahuje vygenerované migrační soubory. Ve složce Models se nachází modely, jako například Points nebo Pokemons. Soubor Bot.cs je třída, která reprezentuje instanci bota. Soubor ConfigJson.cs obsahuje konfigurační data. Program.cs je hlavní spouštěcí soubor aplikace. README.md je vytvořen GithHubem, slouží pro vložení důležitých informací.

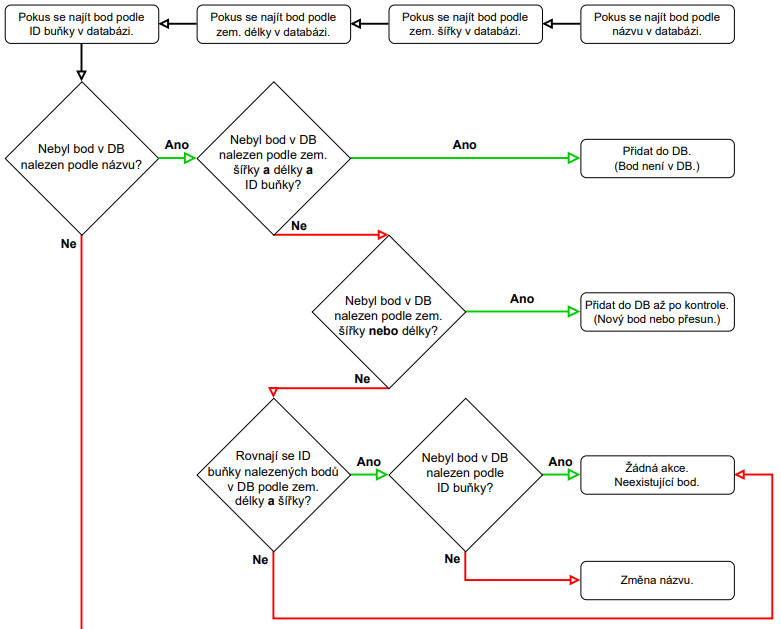
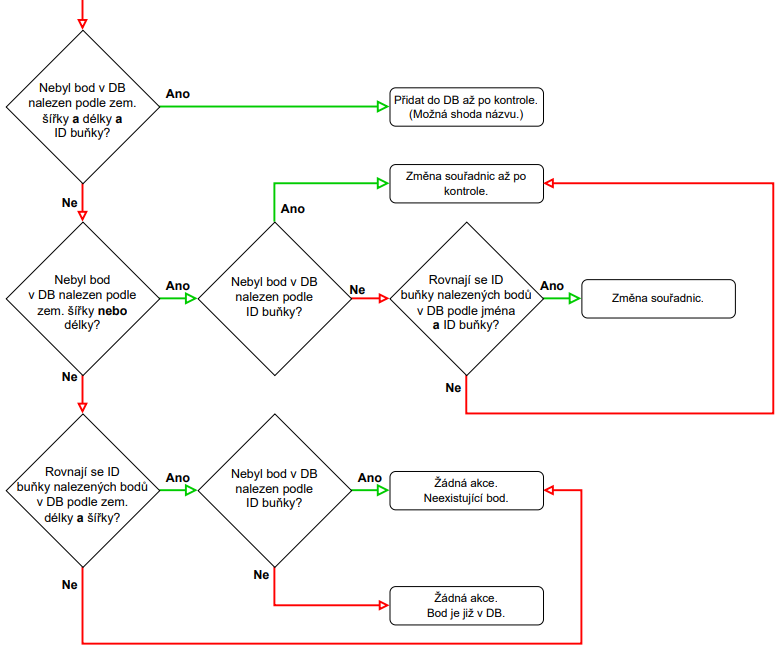
Obrázek 6 Databázová struktura programu

## Příkazy bota

### Nahrávání dat

Vstupní data byla získána z mapy hry Ingress pomocí rozšíření IITC viz kapitola 2.2.4. Třída *CsvHandler* se stará o přijmutí souboru .csv a následné zpracování, které probíhá pomocí integrované třídy *TextFieldParser* pro parsování strukturovaných textových souborů. Funkce vrátí list bodů připravených pro vložení do databáze.

Výsledek převezme třída *DatabaseHandler*, která zajistí správné zpracování dat. Nejdůležitější je kontrola, v jakém stavu je vstupní bod s databází viz kapitola 2.2.6. Zpracování *CheckIfExists()* je zobrazuje Obrázek 6 a Obrázek 7. Metoda vrací výčtový typ Stav viz Obrázek 6.



Obrázek 7 Vývojový diagram postupu vyhodnocení existence bodu

Obrázek 8 Vývojový diagram postupu vyhodnocení existence bodu – pokračování

#### Kontrola dat

#### Změna typu bodu

#### Změna názvu bodu

#### Procházení bodů

#### Obsah obrázku text Popis byl vytvořen automatickyHlášení raidů

#### Nastavení srazu

Obrázek 9 Výčtový typ State

1. Microsoft [online]. 2022 [cit. 2022-08-20]. Dostupné z: https://visualstudio.microsoft.com/cs/vs/ [↑](#footnote-ref-1)
2. HERCEG, Tomáš. Úvod do .net frameworku. Dotnetportal.cz [online]. 03.04.2009 [cit. 2022-08-20]. Dostupné z: https://www.dotnetportal.cz/clanek/125/Uvod-do-NET-Frameworku [↑](#footnote-ref-2)
3. Microsoft [online]. [cit. 2022-08-20]. Dostupné z: https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet [↑](#footnote-ref-3)
4. Microsoft [online]. 2022 [cit. 2022-08-20]. Dostupné z: https://docs.microsoft.com/cs-cz/ef/core/ [↑](#footnote-ref-4)
5. Microsoft [online]. 2022 [cit. 2022-08-20]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/csharp/linq/ [↑](#footnote-ref-5)
6. DSharpPlus Contributors [online]. [cit. 2022-08-21]. Dostupné z: https://dsharpplus.github.io/ [↑](#footnote-ref-6)
7. SQLite [online]. [cit. 2022-08-20]. Dostupné z: https://www.sqlite.org/about.html [↑](#footnote-ref-7)
8. Discord [online]. [cit. 2022-08-20]. Dostupné z: https://discord.com/safety/360044149331-What-is-Discord [↑](#footnote-ref-8)
9. DB Browser for SQLite [online]. [cit. 2022-08-20]. Dostupné z: https://sqlitebrowser.org/about/ [↑](#footnote-ref-9)
10. PROCOPIUC, Octavian. Geometry on the Sphere: Google's S2 Library [online]. [cit. 2022-08-22]. Citováno z: https://docs.google.com/presentation/d/1Hl4KapfAENAOf4gv-pSngKwvS\_jwNVHRPZTTDzXXn6Q/view [↑](#footnote-ref-10)
11. ZEROGHAN. A Comprehensive Guide to S2 Cells and Pokémon GO. Pokémon GO Hub [online], 27. 2. 2019 [cit. 2022-08-22]. Dostupné z: https://pokemongohub.net/post/article/comprehensive-guide-s2-cells-pokemon-go/ [↑](#footnote-ref-11)
12. ZEROGHAN. A Comprehensive Guide to S2 Cells and Pokémon GO. Pokémon GO Hub [online], 27. 2. 2019 [cit. 2022-08-22]. Dostupné z: https://pokemongohub.net/post/article/comprehensive-guide-s2-cells-pokemon-go/ [↑](#footnote-ref-12)