

Autor **Tomáš Kulhavý**

Obor **Informační technologie**

Vedoucí práce **Ing. Tomáš Kazda, DiS.**

Školní rok **2020/2021**

Střední průmyslová škola strojní   
a elektrotechnická a Vyšší odborná škola, Liberec 1, Masarykova 3

Databáze letů pro aeroklub liberec

Maturitní práce

Anotace (Resumé)

Práce bude sloužit jako databáze pro přelety pro Aeroklub Liberec. Piloti budou nahrávat své lety a poté budou vyhodnoceny a bodovány. Poslouží jako žebříček pilotů v klubu se statistikami.

Přináší pilotům v Aeroklubu Liberec přehled o statistikách a náletech pilotů. Může také posloužit při vybírání pilotů na velké závody jako je např. Plachtařské mistroství České republiky juniorů.

Summary

The work will serve as a database for flights for Aeroclub Liberec. Pilots will record their flights and then be evaluated and scored. It will serve as a ranking of pilots in the club with statistics.

It brings the pilots in the Aeroclub Liberec an overview of the statistics and raids of the pilots. It can also be used to select pilots for large competitions, such as the Czech Junior Gliding Championship.

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne

Tomáš Kulhavý

Obsah

[Úvod 1](#_Toc65254085)

[1 Použité technologie 2](#_Toc65254086)

[1.1 ASP.NET Core 5.0 2](#_Toc65254087)

[1.2 Entity Framework Core 2](#_Toc65254088)

[1.3 React 2](#_Toc65254089)

[1.3.1 Reducer 3](#_Toc65254090)

[1.3.2 Další knihovny 3](#_Toc65254091)

[1.4 Visual Studio 2019 3](#_Toc65254092)

[1.5 Visual Studio Code 4](#_Toc65254093)

[1.6 MySQL 4](#_Toc65254094)

[2 Soutěžní bezmotorové létání 6](#_Toc65254095)

[2.1 Pravidla soutěžního bezmotorového létání 6](#_Toc65254096)

[2.1.1 AAT (Assigned area task) 6](#_Toc65254097)

[2.1.2 Rychlostní přelet (Racing) 7](#_Toc65254098)

[2.1.3 Grand Prix 7](#_Toc65254099)

[3 IGC 8](#_Toc65254100)

[3.1 IGC struktura 8](#_Toc65254101)

[3.1.1 Jednotky 8](#_Toc65254102)

[3.1.2 Pojmenování souboru 8](#_Toc65254103)

[3.1.3 A – FVU (Flight Verification Unit) 9](#_Toc65254104)

[3.1.4 G – Security (Bezpečnostní kód) 9](#_Toc65254105)

[3.1.5 H – File header (Hlavička souboru) 9](#_Toc65254106)

[3.1.6 B – Fix (Jednotlivý záznam letu) 10](#_Toc65254107)

[3.1.7 C – Task (Úloha letu) 10](#_Toc65254108)

[3.1.8 D – Differential GPS (Jiný GPS záznamník) 11](#_Toc65254109)

[3.1.9 F – Satelit 11](#_Toc65254110)

[3.1.10 L – Log book (Informace k letu) 11](#_Toc65254111)

[3.2 IGC záznamník 11](#_Toc65254112)

[4 Práce s příchozími daty 13](#_Toc65254113)

[4.1 IGC parser 13](#_Toc65254114)

[4.2 Zjištění poloměrů otočných bodů 13](#_Toc65254115)

[4.3 IGC-XC-Score 13](#_Toc65254116)

[4.4 Počítání vzdálenosti a průměrné rychlosti 13](#_Toc65254117)

[4.5 Počítání TOP 3 letů 14](#_Toc65254118)

[5 Funkční stránka 15](#_Toc65254119)

[6 Realizace aplikace 16](#_Toc65254120)

[Závěr 17](#_Toc65254121)

[Seznam obrázků 18](#_Toc65254122)

[Použitá literatura 19](#_Toc65254123)

[A. Seznam přiložených souborů 1](#_Toc65254124)

[B. Další příloha 2](#_Toc65254125)

Úvod

Tuto práci jsem si vymyslel sám, dlouho jsem uvažoval nad tím, co bych mohl dělat jako maturitní práci a pak mě napadla databáze letů a poté jejich vyhodnocení. V rámci výuky ve čtvrtém ročníku jsme začali programovat v jazyce React. Díky této práci si rozšířím znalosti řešení webových stránek v jazyce React a ASP.NET API. Zvolené téma jsem si vybral, jelikož sám jsem pilotem v Aeroklubu Liberec, tak jsem chtěl naprogramovat stránky, které budou sloužit jako žebříček pilotů v aeroklubu nebo také jako statistiky pilotů. Tato stránka by mohla pomoci členům aeroklubu při výběru pilotů na větší soutěže, jako je Mistroství České republiky nebo Mistroství regionů.

1. Použité technologie
   1. ASP.NET Core 5.0

ASP.NET Core je univerzální framework pro vytváření webových aplikací, služeb nebo aplikací pro Internet věcí (IoT) v jazyce C Sharp. Aplikace se dají nasazovat místně nebo na cloudu. Prostředí je vhodné pro vytváření webového uživatelského rozhraní a webových rozhraní API.

Nejnovější podporovaná verze frameworku .NET Core je 5.0. Tato verze je také použita v aplikaci.

* 1. Entity Framework Core

Entity Framework je nástroj pro objektově relační zobrazení nazývané tzv. ORM (Object Relational Mapper). Je vyvíjený v jazyce C# pro platformu .NET. Podporuje dotazy LINQ, aktualizace a migrace schématu. EF Core podporuje několik databází, včetně SQL Database, která se dá používat místně, ale i na Azure Cloudu. Dále podporuje SQLite, MySQL, PostgreSQL a Azure Cosmos DB.

* 1. React

React nebo také známý jako ReactJS je JavaScriptová knihovna pro tvorbu uživatelského rozhraní vyvíjená společností Facebook. Slouží převážně k tvorbě Single Page Application. Tato aplikace se nachází pouze na jedné stránce. Proto single page application. Je výhodný pro práci s daty, které se rychle mění. Jelikož virtualizaci zajišťuje DOM (Document Object Model). Virtuální DOM uchovává uživatelské rozhraní v paměti. DOM je pak synchronizován, kdy je to opravdu potřeba a tím se ušetří mnoho výpočetního času, který je potřeba na vykreslení DOMu.

Pro psaní kódu aplikace, která používá React, existuje syntaxe JSX, která umožňuje vytvářet HTML elementy jejich prostým zápisem pomocí tagů. Soubor je poté potřeba přeložit pomocí Babelu.

* + 1. Reducer

Reducer je jednoduchá funkce, která vezme dva argumenty. V kontextu nám reducer vezme aktuální stav (state) a akci (přes dispatch si vybereme action type). A vrátí nám nový stav na základě těchto dvou dat.

* + 1. Další knihovny

K dalším knihovnám patří:

* Reactstrap
* Formik
* React-leaflet
* Igc-parser
* Igc-xc-score
* Axios

Reactstrap slouží k zajištění funkčnosti knihovny Bootstrap s Reactem.

Formik je knihovna, která validuje vstupy, formátuje a vypořádává se s chybami, tím že v reálném čase vidíme, jestli náš vstup je validní.

React-leaflet je knihovna pro práci s mapami. V této aplikaci ji používám pro zobrazení úlohy letu a skutečně trasy letu.

Igc-parser je knihovna, která parsuje IGC soubor letu do JSONu.

Igc-xc-score je knihovna, která IGC soubor vyhodnotí a oboduje.

Axios je knihovna se zpětným voláním. Umožňuje asynchronně vykreslovat požadavky.

Další přehled použitých knihoven je v souboru package.json v adresáři ClientApp.

* 1. Visual Studio 2019

Visual Studio 2019 je vývojové prostředí IDE (Integrated Development Environment), které vyvíjí firma Microsoft. Visual Studio umožňuje vyvíjet aplikace v řadě programovacích jazycích. Nejpoužívanější je platforma .NET.

Obsahuje mnoho jazykových služeb. Ty slouží k tomu, aby debugger a editor podporoval ten programovací jazyk, pro který existuje daná jazyková služba. Dále obsahuje mnoho šablon nebo také grafické rozhraní GUI (Graphic User Interface). Podporuje také například vyvíjení her v Unity.

Visual Studio 2019 jsem si zvolil, jelikož mi toto prostředí vyhovuje pro vývoj back-end aplikace.

* 1. Visual Studio Code

Visual Studio Code je open-source editor zdrojových kódu. Vyvíjený je společností Microsoft. Podporuje verzovací nástroj Git. Obsahuje také kontextový našeptávač nebo zvýrazňovač syntaxe. Editor je naprogramovaný v jazyce JavaScript a TypeScript. Běží na frameworku Electron, na kterém běží i konkurenční Atom.

Visual Studio Code jsem si vybral, protože mi vyhovuje pro tvorbu aplikací v Reactu. A obsahuje velké množství doplňku z třetí strany, které zajišťují podporu mnoha jazyků. Má také lepší highlighting (zvýraznění) kódu, než Visual Studio 2019, proto je pro mě přehlednější.

* 1. MySQL

MySQL je relační databázový model vytvořený švédskou firmou MySQL AB, nyní vlastněný Oracle Corporation. Je k dispozici jako free and open-source pod licencí GPL (General Public License) a zároveň pod několika komerčních placených licencí. MySQL je velmi oblíbená a častá kombinace pro základní software webového serveru.

Komunikace probíhá pomocí jazyka SQL. Jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. Jedná se o strukturovaný dotazovací jazyk, tzn. že nepíšeme to, jak má databáze data získat, ale pouze to, jaké data chceme. MySQL je optimalizovaná především na rychlost.

Databáze je založená na tabulkách. Každá tabulka obsahuje informace jednoho typu. Data jsou v databázi uložena v řádcích a v sloupcích. Každý řádek v tabulce představuje jednu položku v databázi. Každý sloupec má pevně stanovený datový typ a každý řádek má svůj unikátní identifikátor.

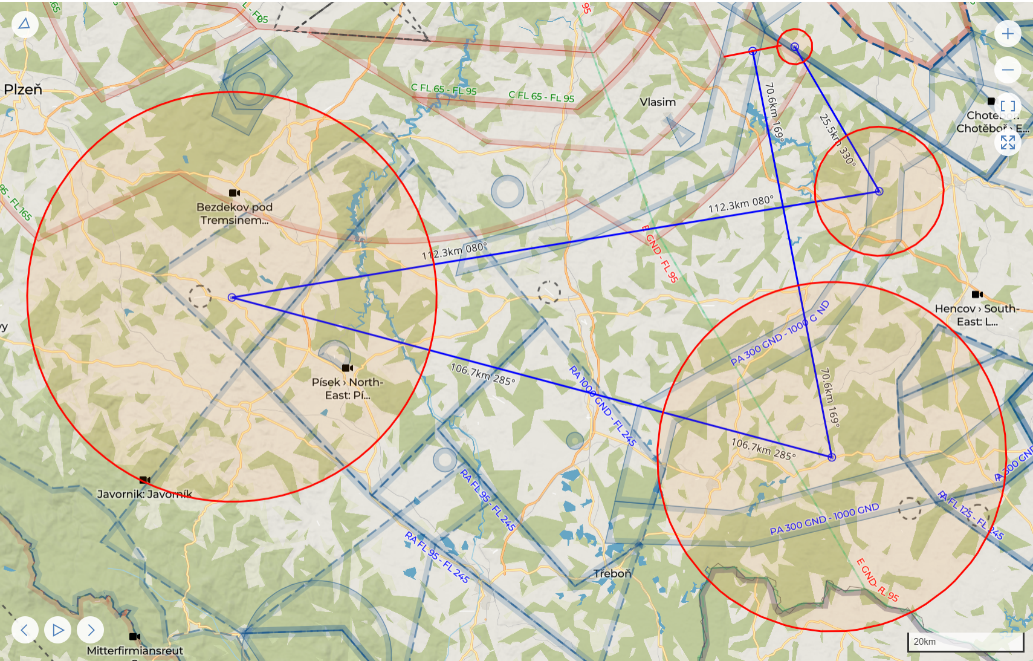
Při navrhování databáze musíme zvážit správný druh vztahů mezi tabulkami a entit. Nejjednodušší vztah představuje relace 1:1, kde Pk (Primary key) v jedné z tabulek odkazuje na data v druhé tabulce. Příkladem této relace může být vztah mezi státem a jeho hlavním městem. Další relací je 1:N. Ta představuje vztah, kde Pk v jedné tabulce odkazuje na více dat v druhé tabulce. Tento vztah může reprezentovat např. třída a studenti. A poslední relací je N:M, kde každý záznam v jedné tabulce může mít více dat ve druhé. Příklad toho vztahu je např. filmy a herci.

Název souboru odpovídá názvu tabulky. Co se týče rozlišování velkých a malých písmen, tak to závisí na platformě. V kombinaci MySQL a Windows se velikost písmen nerozlišuje. Nedoporučuje se vkládat znaky s diakritikou do názvu tabulek nebo databází.

1. Soutěžní bezmotorové létání
   1. Pravidla soutěžního bezmotorového létání

Soutěžní bezmotorové létání se řídí pravidly Mezinárodní letecké federace FAI (Fédération Aéronautique Internationale). Jelikož každý typ kluzáku, má jiný koeficient, tak se na závodech tyto kluzáky rozdělí na několik tříd, kde mají větroně podobný koeficient. Třídy se dělí např. na dvousedadlové větroně nebo na jednosedadlové větroně s rozpětím 15 m, 18 m nebo 21 m a více (tzv. otevřená třída). Koeficient se větroňům přidělil z toho důvodu, že každý typ letadla má jiné výkonnostní vlastnosti. A aby nebyla zvýhodněná letadla s výbornými letovými vlastnostmi, tak se jejich body následně spočítají s jejich koeficientem. To znamená, že piloti s lepšími letadly musí létat rychleji.

* + 1. AAT (Assigned area task)

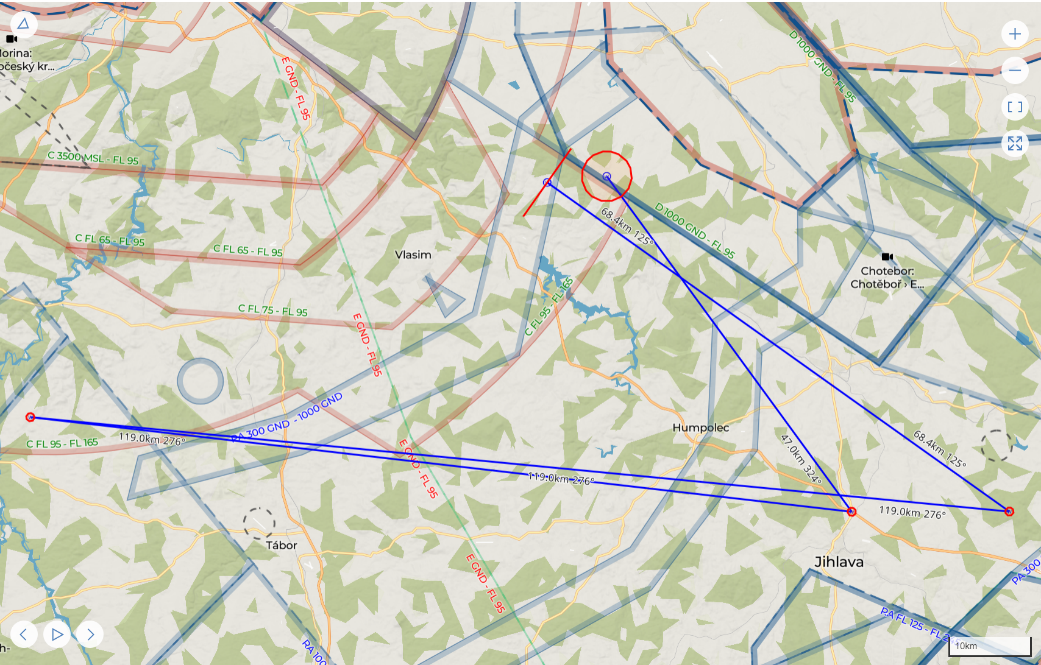
Jedním způsobem je tzv. AAT (Assigned area task), jedná se o velké kružnice o poloměru např. až 50 km, které jsou místo standardních otočných bodů stanoveny po předem naplánované trati.

Obrázek 1 AAT (Assigned Area Task)

Pilot musí těmito kružnicemi proletět a je pouze na jeho rozhodnutí, jestli si zaletí, až na konec kružnice nebo pouze na její začátek. Jeho rozhodnutí závisí na počasí a určité predikci vývoje počasí. Jelikož je předem stanovený čas na trati, pilot musí naplánovanou trať zaletět co nejrychleji, aniž by se vrátil na letiště odletu před skončením stanoveného času. Pokud by se stalo, že by pilot přiletěl do cíle např. o 30 minut dříve před skončením stanoveného času, tak jeho čas na trati se bude rovnat stále tomu stanovenému času. Tzn. že se mu tím zmenší průměrná rychlost na trati a přijde tak o spoustu bodů. Jestliže pilot přiletí po stanoveném čase, tak se mu do jeho délky letu po trati počítá skutečný jeho čas na trati. V zásadě jde o to uletět co nejvíce kilometrů za předem daný čas.

Tento typ úlohy se převážně dává soutěžícím, pokud je na letový den předpověď počasí neurčitá.

* + 1. Rychlostní přelet (Racing)

Druhým způsobem soutěžního létání je tzv. racing neboli rychlostí přelet. V tomto typu úlohy se naplánuje trať s pevně danými body na trati.

Obrázek 2 Rychlostní přelet

Pilot musí tuto celou naplánovanou trať zaletět, za co nejmenší čas a tím docílí vysoké průměrné rychlosti na trase. Poloměr otočných bodů je 500 m.

Tento typ úlohy se převážně zadává, pokud je předpověď počasí na letový den zřetelná a pořadatelé vědí jaký bude vývoj počasí na trati po celý den.

* + 1. Grand Prix

Tento typ úlohy patří k novějším a létá se na závodech, kde je menší počet letadel. Organizátor na den vypíše trať s pevně danými doby, který piloti musí obletět. V jeden přesně daný čas se otevře startovací páska a všichni piloti musí v jeden čas odletět a trať. Jsou zde dané limity na maximální výšku a rychlost při průletu startovacím oknem. Bodování je pak jednodušší, protože všichni piloti odletěli ve stejný čas na trať a vítězí ten, co do cíle přiletí jako první.

1. IGC

Tento standard datových souborů byl vyvinut ve spolupráci se softwarovými vývojáři záznamových zařízení. Standard se nazývá IGC (International Gliding Commission). Jeho účelem je usnadnit zavedení technologie GPS (Global Positioning System) do soutěžního bezmotorového létání pod sportovním řádem FAI. Mezinárodní letecké federace FAI tento standard používá nejen na soutěžní létání, ale také na FAI odznaky nebo na rekordní lety. IGC standard ověřuje pozice otočných bodů, souřadnice letadla a zabezpečuje data.

Tento standard byl představen a přijat na zasedání v září roku 1994.

* 1. IGC struktura

Každý záznam v IGC souboru začíná specifickým znakem, který identifikuje druh záznamu.

* + 1. Jednotky

Všechny data IGC záznamu jsou zaznamenávány v následujících jednotkách.

Pro záznam času se používá koordinovaný světový čas UTC (Coordinated Universal Time), je to základní čas pro všechna časová. Jedná se o souřadnicový čas a odvíjí se od základního poledníku v Greenwich.

Na měření délky letu se používají metry.

Rychlost letu měříme v kilometrech za hodinu.

Datový formát letu je DDMMYY (den, měsíc, rok)

Výška letu se udává v metrech. Zaznamenává se GPS výška, ale také výška, kterou záznamové zařízení bere ze svého barometrického senzoru, který se musí každých 5 let zkalibrovat.

* + 1. Pojmenování souboru

Každý IGC záznam má své jméno, to je dané normou ISO 9660. Používá se pro všechny oficiální záznamy letu. Ostatní názvy souborů jsou povoleny pro obecné používání, ale FAI ani další soutěžní organizace jiný název neakceptují. Všechny znaky v názvu souboru jsou alfanumerický.

„A“ reprezentuje 10, „B“ je 11 a „C“ je 12. Znaky 1-9 a A-C používáme pro měsíce.

Pro zápis roku se používá modulo 10, takže každých 10 se začíná znovu.

Pro číslo letu ten daný den se začíná číslem 1.

Sériové číslo záznamové jednotky je unikátní číslo. Pro každého výrobce záznamových zařízení je k dispozici 46656 identifikačních čísel.

* YMDCXXXF.IGC
  + Y = rok
  + M = měsíc
  + D = den
  + C = výrobní ID
  + X = sériové unikátní číslo
  + F = číslo letu za den
    1. A – FVU (Flight Verification Unit)

Tento typ záznamu začíná písmenem „A“. Specifikuje unikátní sériové číslo záznamového zařízení.

* AMNNNNN
  + A = typ záznamu
  + M = ID výrobce
  + N = unikátní ID
    1. G – Security (Bezpečnostní kód)

Záznam „G“ ověřuje, zda data ASCII nebyla během letu nebo po něm změněna. Používá se pro ověření záznamu. Bezpečnostní kód musí být vygenerován FVU. Bezpečnostní kód musí obsahovat platné znaky ASCII tabulky.

* + 1. H – File header (Hlavička souboru)

Hlavička souboru začíná písmenem „H“. Jsou zde informace o datumu, kdy byl let letěn. Dále jméno pilota, pokud je kluzák dvoumístný, uvádí se jméno druhého pilota. Také zde najdeme typ kluzáku, třídu, kam kluzák patří, registraci kluzáku a také jeho soutěžní znak. Dále jsou zde informace o verzi firmwaru záznamového zařízení, typ zařízení. Nebo označení antény pro snímání pozice a senzoru pro snímání výšky letu. Je zde také uvedena časová zóna letu.

* + 1. B – Fix (Jednotlivý záznam letu)

Fix neboli jednotlivý záznam letu, se zaznamenává pravidelně po pár vteřinách. Dá se zaznamenávat každou 1 vteřinu až 5 vteřin. Záznam začíná písmenem „B“. Obsahuje v sově informace o čase, zeměpisné šířce a délce, validaci fixu, výška z barometru, GPS výška, přesnost fixu a pokud záznamové zařízení detekuje podle hluku motor, obsahuje i otáčky motoru.

* BHHMMSSDDMMMMMNDDDMMMMMEVPPPPPGGGGGAAARRRR
  + HHMMSS = časový záznam
  + DDMMMMMN = zeměpisná šířka
  + DDDMMMMME = zeměpisná délka
  + V/A = validace fixu (A = valid, V = warning)
  + PPPPP = výška z barometru
  + GGGGG = GPS výška
  + AAA = přesnost fixu
  + RRRR = otáčky motoru (pokud je hluk detekován)
    1. C – Task (Úloha letu)

Záznam, který začíná písmenem „C“ označuje úlohu letu. Obsahuje datum a čas letu, ID úlohy, která začíná číslem 0001 (označuje pořadí deklarované tratě daný den), GPS souřadnice vzletu a přistání a také GPS souřadnice otočných bodů na trati a jako poslední je název otočného bodu.

* CDMMYYHHMMSSFDFMFYIIIITTTEXTSTRING (informace o úloze)
* CDDMMMMMNDDDMMMMMETEXTSTRING (otočný bod)
  + DDMMYY = datum
  + HHMMSS = čas
  + IIII = ID úlohy
  + DDMMMMMNDDDMMMMME = GPS souřadnice
    1. D – Differential GPS (Jiný GPS záznamník)

Záznam „D“ nám dává informace, jestli záznamové zařízení snímá GPS souřadnice pomocí jiného GPS zařízení. Pokud je absence těchto záznamů, znamená to, že záznamové zařízení používá integrovanou GPSku.

* DQSSSS
  + Q = GPS kvalifikátor
  + S = ID stanice
    1. F – Satelit

Záznam „F“ zapisuje informace o aktuálně používaném satelitu. Vždy se zaznamená, pokud se přijímací satelit změní.

* FHHMMSSAABBCCDD
  + HHMMSS = čas
  + AABBCCDD = ID satelitu
    1. L – Log book (Informace k letu)

Tento záznam „L“ umožňuje přidat několik textových řádků ve volném formátu. Jelikož záznam „L“ není pokryt bezpečnostním kódem, umožňuje nám přidat do souboru informace i po dokončení letu, např. nějaké komentáře k letu atd.

* 1. IGC záznamník

IGC záznamník neboli logger, se používá pro zaznamenávání celého letu. Jehož výstupem je následně IGC soubor. Existuje několik typů loggerů.

Mezi starší patří ty, co mají v sobě nainstalovaný hardware pouze pro zaznamenávání letu. Tento logger je připojený k baterce, jelikož v sobě baterky nemají. Dále se musí pomocí kabelů připojit k externí anténě, pokud v ji v sobě nemají. Také se dají připojit pomocí kabelu k externí navigaci, kde díky IGC loggeru uvidíme naši polohu. Tyto navigace mají i palubní počítač, který dokáže spočítat aktuální klouzavost letadla, předpokládaný přílet do cíle nebo na otočný bod. Ukazují nám také základní informace o letu jako je rychlost, výška a vario.

Mezi novější patří už IGC záznamníky, co v sobě mají logger, navigaci i palubní počítač už zabudovaný. Také mají svou vlastní baterku, takže nemusí být napojený na baterku. Tyto navigace vydrží už poměrně dlouhou dobu, takže pokud pilot letí několikahodinový přelet nebo i přelet, který trvá celý den tak mu tento logger na jedno nabití stačí. Vydrží až 12 hodin.

1. Práce s příchozími daty

Všechna příchozí data zpracovávám ihned po vložení souboru na stránku. Data ve formátu JSON pošlu pomocí axiosu do databáze. Lety si mohu následně vylistovat všechny nebo podle pilotů. K zobrazení letu používám mapy a barograf.

* 1. IGC parser

IGC parser je open-sourcová knihovna, která mi celý IGC soubor pomocí regulárních výrazů rozparsuje do JSONu. Díky tomu, že jsou data v JSONu, mohu k nim přidávat další informace o letu. A také je následně poslat do FlightLogControlleru, kde je pomocí newtonsoftu deserializuji do databáze.

* 1. Zjištění poloměrů otočných bodů

Dále celý soubor rozparsuji pomocí regulárních výrazů ještě jednou. Abych zjistil poloměry otočných bodů, které jsou potřeba pro správné počítání uletěné vzdálenosti a průměrné rychlosti letu.



Obrázek 3 Regulární výraz na zjištění poloměrů

* 1. IGC-XC-Score

Poté celý IGC soubor zpracuji ještě jednou, abych získal pomocí open-source knihovně IGC-XC-Score body za let.

Bodování letů je následující. Pokud je nadeklarovaná trasa uletěná, pilot získá více bodu, než kdyby let nedoletěl celý. Pokud pilot úlohu vůbec nezačal, nespočítá se žádná průměrná rychlost ani uletěná vzdálenost a nedostane žádné body.

* 1. Počítání vzdálenosti a průměrné rychlosti

Vzdálenost letu počítám následovně. Pomocí cyklu „for“, který mi prochází všechny otočné body, si nastavuji radius (poloměr otočných bodů, které jsem získal) a střed kružnic. Do proměnné si uložím vzdálenost mezi body v úloze a zjistím, pomocí cyklu „for“, jestli pilot proletěl daným bodem. Pokud ano, uložím si čas, kdy byl v místě toho bodu. A zvýším proměnnou „countTP“ o jedna. Do proměnné „dist“ si uložím vzdálenost mezi body. Takto tento cyklus projede celý let. Pokud se zjistí, že pilot bodem v úloze neproletěl, nezvyšuji „countTP“ ani nepřipočítávám vzdálenost. Jakmile tento cyklus projede celý let. Porovnám proměnnou „countTP“ (skutečný počet otočných bodů, které pilot proletěl) a počet otočných bodů, které byly nadeklarovány. Pokud se tyto čísla nerovnají, spočítám vzdálenost mezi body, kterou pilot neuletěl. Tu odečtu od proměnné „dist“ a přičtu vzdálenost mezi bodem, kde pilot přistál a posledním bodem nadeklarované tratě, kterým pilot proletěl. Tímto výpočtem zjistím skutečnou uletěnou vzdálenost

Průměrnou rychlost vypočítám tak, že vydělím vzdálenost letu časem na trati.

* 1. Počítání TOP 3 letů

Ihned po deserializaci letu ve FlightLogControlleru, volám metodu „GiveTopBool“. Tato metoda si vytáhne z databáze všechny lety od pilota, který let nahrává z roku, kdy je nahrávaný let letěn. Tyto záznamy seřadím sestupně podle bodů za let a vezmu si pouze tři nejlepší lety.

Pokud počet letů, které jsme si vytáhli z databáze, není větší než dva, tak automaticky přidělí letu, že „Topflight“ se rovná true. A pilotovi se přičtou body za let do jeho „TopScore“ (pilotovi celkové body za nejlepší tři lety).

Jestliže počet letů je větší než dva, tak porovná nahrávaný let s lety v databázi. A pokud má nahrávaný let více bodů, tak jeho „Topflight“ se rovná true a poslední nejlepší let je nastaven na false. Dále odečteme od „TopScore“ poslední nejlepší let a přičteme body za nový nejlepší let.

1. Funkční stránka
2. Realizace aplikace

Závěr

Tak jsem to dokázal! A teď ještě, co jsem mohl udělat jinak a jak by se dalo v práci pokračovat.

Seznam obrázků

[Obrázek 1 AAT (Assigned Area Task) 6](file:///D:\WEB\MP2021-databaze-letu-pro-aeroklub-liberec\Dokumentace,%20DB\MP2021-Kulhavý-Tomáš-P4-Databaze-letu-pro-aeroklub-liberec.docx#_Toc65252752)

[Obrázek 2 Rychlostní přelet 7](file:///D:\WEB\MP2021-databaze-letu-pro-aeroklub-liberec\Dokumentace,%20DB\MP2021-Kulhavý-Tomáš-P4-Databaze-letu-pro-aeroklub-liberec.docx#_Toc65252753)

[Obrázek 3 Regulární výraz na zjištění poloměrů 14](#_Toc65252754)

Použitá literatura

1. 1. **Microsoft.** Dokumentace k ASP.NET Core. *Microsoft Docs.* [Online] Microsoft. [Citace: 15. 02 2021.] https://docs.microsoft.com/cs-cz/aspnet/core/?view=aspnetcore-5.0.
2. 2. —. Dokumentace k EF Core. *Microsoft Docs.* [Online] Microsoft. [Citace: 16. 02 2021.] https://docs.microsoft.com/cs-cz/ef/core/.
3. 3. **Facebook.** Getting started. *React Docs.* [Online] Facebook Inc. [Citace: 16. 02 2021.] https://reactjs.org/docs/getting-started.html.
4. 4. **React & Redux. [Online] [Citace: 19. 02 2021.] https://react-redux.js.org/.**
5. **5. Visual Studio 2019. [Online] Microsoft docs. [Citace: 19. 02 2021.] https://visualstudio.microsoft.com/cs/vs/.**
6. **6. Visual Studio Code. [Online] Microsoft docs. [Citace: 19. 02 2021.] https://en.wikipedia.org/wiki/Visual\_Studio\_Code.**
7. **7. MySQL. [Online] [Citace: 19. 02 2021.] https://en.wikipedia.org/wiki/MySQL.**
8. **8. FAI. [Online] [Citace: 19. 02 2021.] https://www.fai.org/.**
9. **9. UTC time. [Online] [Citace: 19. 02 2021.] https://en.wikipedia.org/wiki/Coordinated\_Universal\_Time.**
10. **10. Gliding IGC. [Online] [Citace: 19. 02 2021.] http://www.gliding.ch/images/news/lx20/fichiers\_igc.htm#Lrec.**
11. Seznam přiložených souborů
12. Složka s řešením Visual Studio (MP2021\_LKLB)
13. Složka DB/DB diagram (návrh databáze)
14. Písemná práce MP
    1. MP2021-Kulhavý-Tomáš-P4-Databaze-letu-pro-aeroklub-liberec.docx
    2. MP2021-Kulhavý-Tomáš-P4-Databaze-letu-pro-aeroklub-liberec.pdf
15. Písemná práce SOČ
    1. MP2021-Kulhavý-Tomáš-P4-Databaze-letu-pro-aeroklub-liberec.docx
    2. MP2021-Kulhavý-Tomáš-P4-Databaze-letu-pro-aeroklub-liberec.pdf
16. Další příloha