



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ

Bc. Václav Vidoň

**ZPRACOVÁNÍ A ANALÝZA KONDENZAČNÍCH STOP
ZA POUŽITÍ DATABÁZOVÝCH SYSTÉMŮ**

Diplomová práce

2019



K621..... Ústav letecké dopravy

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Václav Vidoň

Kód studijního programu a studijní obor studenta:

N 3710 – PL – Provoz a řízení letecké dopravy

Název tématu (česky): **Zpracování a analýza kondenzačních stop za použití
databázových systémů**

Název tématu (anglicky): Processing and Analysis of Condensation Trails Data Using
Database Systems

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Popis příčin vzniku a chování kondenzačních stop
- Analýza a zpracování dat z ADS-B přijímačů FD ČVUT
- Vytvoření vhodné databázové struktury
- Zpracování dat do databázové struktury
- Vypacování jednoduchého uživatelského rozhraní pro použití této databáze



- Rozsah grafických prací: dle pokynů vedoucího diplomové práce
- Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)
- Seznam odborné literatury: D. J. Auer, D. M. Kroenke: Databáze
L. Palečková: Analýza kondenzačních stop prostřednictvím dat vysílaných letadly
Krška a kol.: Meteorologie

Vedoucí diplomové práce: **Ing. Sébastien Lán**
doc. Ing. Jakub Hospodka, Ph.D.

Datum zadání diplomové práce: **27. července 2018**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **28. května 2019**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

doc. Ing. Jakub Kraus, Ph.D.
vedoucí
Ústavu letecké dopravy



doc. Ing. Pavel Hrubeš, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Václav Vidoň
jméno a podpis studenta

V Praze dne27. července 2018

Poděkování

Mé poděkování patří Ing. Sébastienu Lánovi za odborné vedení, ochotu, trpělivost, rady, čas a pomoc, kterou mi při vypracovávání této diplomové práce věnoval. Taktéž bych rád poděkoval mé rodině a slečně Svatavě Šímové za podporu při psaní této diplomové práce.

Prohlášení

„Nemám závazný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).“

„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“

V Praze dne 28. května 2019



.....
podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA DOPRAVNÍ

ZPRACOVÁNÍ A ANALÝZA KONDENZAČNÍCH STOP ZA POUŽITÍ
DATABÁZOVÝCH SYSTÉMŮ

Bc. Václav Vidoň
Diplomová práce
2019

Abstrakt

Práce se zabývá popisem příčin vzniku a chování kondenzačních stop za letadly, leteckými radary, komunikací mezi letadly a pozemními řídicími stanicemi, popisem funkce, analýzou a zpracováním dat z ADS-B přijímačů a registrů. Dále potom návrhem a vytvořením vhodné webové aplikace s jednoduchým uživatelským rozhraním a databází pro skladování a práci s těmito daty.

Klíčová slova

Kondenzační stopa, Letecký radar, Odpovídač, Mód S, ADS-B, HTML, PHP, SQL

CZECH TECHNICAL UNIVERISTY IN PRAGUE
FACULTY OF TRANSPORTATION SCIENCES

PROCESSING AND ANALYSIS OF CONDENSATION TRAILS DATA USING
DATABASE SYSTEMS

Bc. Václav Vidoň
Master's Thesis
2019

Abstrakt

This Master's thesis describes the causes and behavior of aircraft condensation trails, aircraft radar technology, communication between aircraft and ground control stations, description of the function, analysis and processing of the ADS-B receivers and registers data. It also deals with the design and creation of a suitable web application with a simple user interface and a database for storing and processing this data.

Keywords

Condensation trail, Aircraft radar, Responder, Mode S, ADS-B, HTML, PHP, SQL

Obsah

SEZNAM POUŽITÝCH ZKRATEK	8
1 ÚVOD	11
2 KONDENZAČNÍ STOPY.....	13
2.1 VZNIK KONDENZAČNÍCH STOP.....	13
2.2 POZOROVÁNÍ KONDENZAČNÍCH STOP.....	14
2.3 SCHMIDT – APPLMENOVA TEORIE TVORBY KONDENZAČNÍCH STOP.....	16
3 LETECKÉ RADARY.....	18
3.1 PRIMÁRNÍ PŘEHLEDOVÝ RADAR (PSR)	18
3.2 SEKUNDÁRNÍ PŘEHLEDOVÝ RADAR (SSR).....	19
3.3 NEADRESNÁ KOMUNIKACE SEKUNDÁRNÍHO RADARU SSR	20
3.3.1 <i>Dotaz</i>	20
3.3.2 <i>Odpověď</i>	21
3.4 ADRESNÁ KOMUNIKACE SEKUNDÁRNÍHO RADARU SSR.....	23
3.4.1 <i>Dotaz</i>	23
3.4.2 <i>Všeobecný dotaz typu A/C módu S</i>	23
3.4.3 <i>Všeobecný dotaz typu A/C/S módu S</i>	24
3.4.4 <i>Adresný dotaz módu S</i>	24
3.4.5 <i>Odpověď</i>	24
3.5 PROBLÉMY V KOMUNIKACI RADAR – LETADLO.....	25
3.5.1 <i>Problematické určení azimutu</i>	25
3.5.2 <i>Odpovědi na dotazy z jiného zdroje</i>	27
3.5.3 <i>Překrývání odpovědí</i>	27
4 ADS-B.....	28
4.1 PŘENOS INFORMACÍ POMOCÍ ADS-B	28
4.1.1 <i>ADS-B OUT</i>	28
4.1.2 <i>ADS-B IN</i>	29
4.1.3 <i>Modulační schéma PPM</i>	29
4.2 VYBAVENÍ PRO ADS-B	30
4.2.1 <i>Povinnost ADS-B vybavení v Evropě</i>	31
4.2.2 <i>Povinnost ADS-B vybavení v USA a ve světě</i>	32
4.3 AMATÉRSKÉ VYUŽITÍ ADS-B	33
4.3.1 <i>Hardwarové vybavení pro amatérský příjem ADS-B dat</i>	34
4.3.2 <i>Softwarové vybavení pro amatérský příjem ADS-B dat</i>	36
4.3.3 <i>Sdílení ADS-B dat</i>	37

5	DEFINICE, POPIS A STRUKTURA PŘIJÍMANÝCH DAT.....	38
5.1	UNIX TIME	39
5.2	MESSAGE	39
5.3	TIME	41
5.4	DF - DOWNLINK FORMAT	41
5.5	TYPE CODE/BDS NUMBER.....	42
5.6	LATITUDE.....	42
5.7	LONGITUDE.....	42
5.8	ALTITUDE.....	42
5.9	GROUND SPEED	43
5.10	BARO DIFFERENCE.....	44
5.11	HEADING	44
5.12	TAS	44
5.13	MACH	44
5.14	FOM/SOURCE	44
5.15	METEOROLOGICKÁ DATA - WIND SPEED, WIND DIRECTION, TEMPERATURE, PRESSURE, TURBULENCE, HUMIDITY	45
5.16	POUŽITÍ DAT V PRAKTICKÉ ČÁSTI PRÁCE	46
6	DATABÁZOVÉ SYSTÉMY A JEJICH VYUŽITÍ VE ZPRACOVÁNÍ ADS-B DAT.....	47
6.1	VÍCEVRSTVÁ A TŘÍVRSTVÁ ARCHITEKTURA.....	48
6.2	DATOVÁ VRSTVA V TŘÍVRSTVÉ ARCHITEKTUŘE	49
6.3	DATOVÁ STRUKTURA DATABÁZE	50
6.3.1	<i>Struktura vytvořené relační databáze</i>	<i>51</i>
6.4	APLIKAČNÍ VRSTVA V TŘÍVRSTVÉ ARCHITEKTUŘE	52
6.5	PREZENTAČNÍ VRSTVA V TŘÍVRSTVÉ ARCHITEKTUŘE.....	54
6.5.1	<i>Jazyk HTML.....</i>	<i>54</i>
7	JAZYK SQL A PHP.....	56
7.1	DATA QUERY LANGUAGE (DQL)	56
7.2	DATA DEFINITION LANGUAGE (DDL).....	57
7.3	DATA CONTROL LANGUAGE (DCL)	58
7.4	DATA MANIPULATION LANGUAGE (DML)	58
7.5	JAZYK PHP	59
7.5.1	<i>Technická specifika jazyka PHP.....</i>	<i>59</i>
7.5.2	<i>Proměnné a jejich datové typy.....</i>	<i>60</i>
8	MANUÁL K OBSLUZE WEBOVÉ APLIKACE	61
8.1	PŘÍSTUP K APLIKACI.....	61

8.2	FILTER RESULTS BY DATE	61
8.3	EXPORT ADS-B DATA.....	63
8.4	UPLOAD FILES TO DATABASE	65
9	ZDROJOVÝ KÓD WEBOVÉ APLIKACE	67
9.1	ZDROJOVÝ KÓD APLIKAČNÍ VRSTVY	67
9.2	ZDROJOVÝ KÓD PREZENČNÍ VRSTVY	87
9.3	ZDROJOVÝ KÓD DATOVÉ VRSTVY.....	92
10	ZÁVĚR	93
	SEZNAM POUŽITÝCH ZDROJŮ.....	94
	SEZNAM OBRÁZKŮ	99
	SEZNAM TABULEK.....	101

Seznam použitých zkratk

ADS-B	Automatic dependent surveillance— broadcast	Automatický závislý přehledový systém - vysílání
ADS-B IN	Automatic dependent surveillance— broadcast IN system	Automatický závislý přehledový systém – vysílání IN
ADS-B OUT	Automatic dependent surveillance— broadcast OUT system	Automatický závislý přehledový systém – vysílání OUT
ASP.NET	Active Server Pages .net framework	Framework „Active Server Pages .net“
AWR	Airborne Weather Radar	Letadlový meteorologický radar
BDS	Binary Data Selector	Selektor binárních dat
BE	Backend	Backend
BUKOP	Secondary radar located on the „Buchtův Kopec“	Sekundární radar na Buchtově kopci
CFR	Code of Federal Regulations	Kodex federálních předpisů
CSS	Cascading Style Sheets	Kaskádové styly
ČHMÚ	Czech hydrometeorological Institute	Český hydrometeorologický ústav
ČVUT	Czech Technical University in Prague	České vysoké učení technické v Praze
DCL	Data control language	Soubor příkazů pro práci s právy uživatelé
DDL	Data definition language	Soubor příkazů pro práci s objekty databáze
DF	Downlink format	Formát pro stahování
DML	Data Manipulation Language	Soubor příkazů pro manipulaci daty databáze
DQL	Data query language	Soubor příkazů pro práci s daty databáze
DVB-T	Digital Video Broadcasting — Terrestrial	Digitální pozemní přenos vysílání
EASA	European Aviation Safety Agency	Evropská agentura pro bezpečnost letectví
EU	European Union	Evropská unie
FAA	Federal Aviation Administration	Federální letecká správa

FD ČVUT	Faculty of Transportation Sciences, Czech Technical University in Prague	Fakulta dopravní Českého vysokého učení technického v Praze
FE	Front-end	Front-end
FIS-B	Flight Information System Broadcast	System vysílání letových informací
FRUIT	False Response Un-correlated In Time	Odezva, časově neodpovídající
FTP	File Transfer Protocol	Protokol pro přenos souborů mezi počítači
GNSS	Global Navigation Satellite System	Globální družicový polohový systém
GPS	Global Positioning System	Globální polohový systém
GS	Ground Speed	Rychlos vůči zemskému povrchu
HTML	Hypertext Markup Language	Značkovací jazyk pro tvorbu webových stránek
IFR	Instrument Flight Rules	Let podle přístrojů
ICAO	International Civil Aviation Organization	Mezinárodní organizace pro civilní letectví
IoT	Internet of Things	Internet věcí
KOS ČVUT	CTU Study Information System	Studijní informační systém ČVUT – „KOmpONENTA Student“
LTE	Long Term Evolution	Technologie určená pro vysokorychlostní internet v mobilních sítích
MHz	Megahertz	Megahertz
MSSR	Monopulse Secondary Surveillance Radar	Monopulzní sekundární radar
NM	Nautical mile	Námořní míle
OS	Operating system	Operační systém
Obr	Picture	Obrázek
PC	Personal Computer	Osobní počítač
PHP	Hypertext Preprocessor	Hypertextový preprocesor
POE	Power over Ethernet	Napájení po datovém síťovém kabelu

POP3	Post Office Protocol version 3	Post Office Protocol verze 3
PPM	Pulse Position Modulation	Modulace pulzní polohy
PSR	Primary Surveillance Radar	Primární přehledový radar
ŘLP ČR s.p.	Air Navigation Services of the Czech Republic	Řízení letového provozu České republiky, s. p.
SESAR	Single European Sky ATM Research	Technologická součást iniciativy EU „Jednotné evropské nebe“
SMTP	Simple Mail Transfer Protocol	Internetový protokol určený pro přenos zpráv elektronické pošty
SQL	Structured Query Language	Standardizovaný strukturovaný dotazovací jazyk
SSR	Secondary surveillance radar	Sekundární přehledový radar
TAS	True Air Speed	Pravá vzdušná rychlost
TIS-B	Traffic information service – broadcast	Systém vysílání leteckých dopravních informací
ÚCL	Civil Aviation Authority (Czech Republic)	Úřad pro civilní letectví
UKV	Ultra High Frequency	Ultra krátké vlny
USB	Universal Serial Bus	Univerzální sériová sběrnice
UX	User Experience	Uživatelská zkušenost/rozhraní
HTTP	Hypertext Transfer Protocol	Internetový protokol určený pro komunikaci s WWW servery

1 ÚVOD

Za posledních 100 let udělalo lidstvo obrovský pokrok ve všech technologických směrech. Mnoho z vynálezů lidí je viditelných v každodenním životě, jedním z nejvíce viditelných je pokrok v letectví a leteckém průmyslu. Od prvního vzletu motorem poháněného letadla na kopci Kitty Hawk v roce 1903 uplynulo již přes sto let a díky pokroku jsme v současnosti schopni létat násobky rychlostí zvuku, přepravit stovky cestujících přes velké vzdálenosti a cestovat za ceny, odpovídající zlomkům jejich měsíčních příjmů.

S technologickým pokrokem ale přichází i nové výzvy a komplikace. V případě letectví šlo nejdříve například o vývoj motoru, který je schopen letadlo s pilotem pohánět a následně takového motoru, který je schopen přepravovat větší množství cestujících nebo zboží. Lidstvo se tedy postupně dostalo od jednoduchého pístového motoru přes první německé tryskové motory až po moderní turbínové motory pohánějící mnohotunová letadla.

Se zavedením moderních motorů v civilním letectví vznikl i fenomén kondenzačních stop. Kondenzační stopy jsou úkazy, které za určitých podmínek vznikají za letícím letadlem (respektive za motory letícího letadla) a bylo rozhodnuto tyto stopy sledovat, studovat a analyzovat. Zkoumáním těchto vzdušných čar můžeme zjistit mnoho nového o atmosféře planety Země, o reakcích teplých výfukových plynů s atmosférou při průletu a o chování atmosféry jako takové. Kondenzační stopy jsou také důležitý prostředek k sledování stavu atmosféry pro meteorology, kteří ovšem nejsou v současnosti schopni se stoprocentní jistotou určit přesné složení atmosféry ve výšce a mnohdy jim tvar, chování a struktura kondenzačních stop pomáhá k analýze atmosféry ve výšce letícího letadla.

Dalším obsáhlým tématem této diplomové práce jsou data z ADS-B přijímačů a jejich analýza. ADS-B přijímače jsou zařízení pro profesionální i amatérské sledování letového provozu pomocí přijímání dat z odpovídačů na palubách letadel – jsou například využívány pro provoz serveru Flightradar24, kde slouží pro vytvoření interaktivní mapy, na které můžeme živě sledovat celosvětový letecký provoz. Tato data shromažďuje i Fakulta dopravní ČVUT v Praze pomocí svých přijímačů v Praze a kamer pro pozorování kondenzačních stop, které jsou umístěny v západočeském Děčíně. Diplomová práce zpracovává takto získaná data z přijímačů do webové aplikace s databázovým úložištěm, v níž se s nimi dá mnohem lépe pracovat, než když jsou v tabulkové formě strojově vygenerovaných dat.

Tato webová aplikace bude sloužit jako interaktivní prostředek pro pracovníky a studenty fakulty, kteří si tak mohou z aplikace jednoduše vyfiltrovat nahrané tabulky a k těmto datům přidat vlastní poznámky k původu, chování a životnosti stopy pro tabulku daného letadla. Součástí aplikace je i možnost několika druhů exportu dat a možnost nahrát jednotky nebo balíčky nových tabulek, které jsou určeny pro zpracování. K interakci bude uživateli

sloužit jednoduché prostředí webové stránky, jehož vytvoření je také součástí této práce. Aplikace je vytvořena pomocí databázového jazyka SQL, řídicí a výpočetní část je napsána v jazyce PHP a samotné uživatelské prostředí bude napsáno kombinací jazyků PHP, HTML, Javascript a CSS, přičemž samotná struktura tohoto funkčního celku bude mít třívrstvou architekturu.

Úkolem této práce je popsat vznik kondenzačních stop, objasnit aktuální implementaci komunikačních systémů se zaměřením na systém ADS-B, který je používán v letectví v České republice i ve světě a zároveň vytvořit jednoduchý systém pro správu záznamů komunikace mezi palubním vysílačem a pozemním přijímačem. Tato komunikace je seskupována do tabulek, což ulehčí pracovníkům Ústavu letecké dopravy FD ČVUT jejich práci při sledování a analýze kondenzačních stop pro jednotlivé letouny.

2 KONDENZAČNÍ STOPY

Kondenzační stopy (v anglickém jazyce *condensation trails*) jsou bílé stopy za motory nebo křídly letadel, které mají tvar čar. Zpravidla jsou pozorovatelné okem. Samotná tvorba těchto čar závisí na několika technických i meteorologických podmínkách [1, 2, 3].

Z technologických podmínek je to hlavně:

- Let je prováděn v horní troposféře, ve výškách 8–12 km,
- Letadlo je osazeno motory, spalujícími fosilní palivo,
- Letecké motory vypouštějí výfukové plyny do atmosféry.

Z meteorologických podmínek je to hlavně:

- Správná kombinace vlhkosti a teploty vzduchu okolní atmosféry,
- Teplota silně pod bodem mrazu (minus 40 °C až 60 °C).

2.1 Vznik kondenzačních stop

Samotný vznik kondenzačních čar je přibližně následovný: letadlo, letící ve správné výšce a je osazeno motory, spalujícími fosilní paliva při letu vypouští velmi horké výfukové plyny, o teplotách stovek stupňů celsia [3, 6].

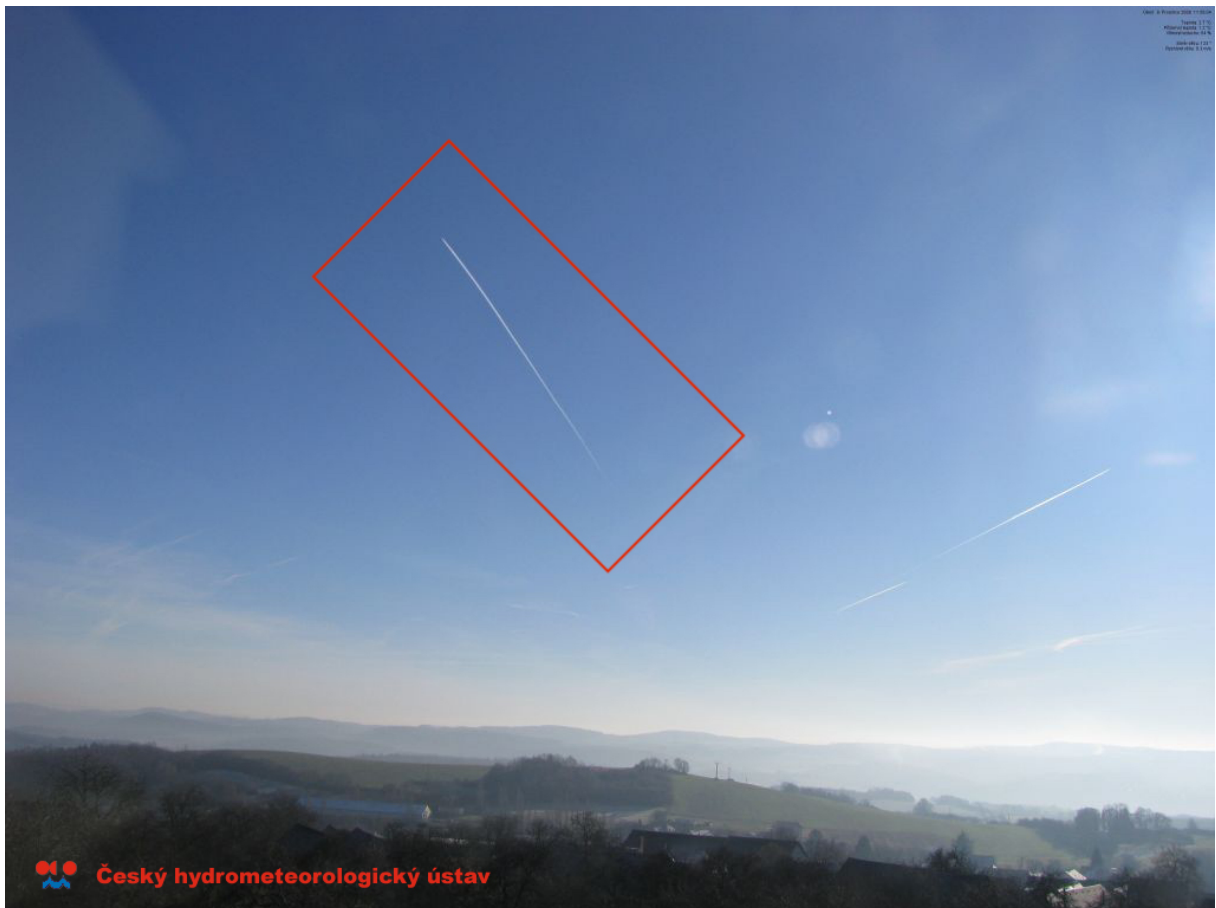
Tyto plyny obsahují i relativně velké množství vodní páry, která má podobnou teplotu, jako ostatní vypouštěné výfukové plyny. Ve výfukových plynech jsou také obsaženy mikročástice pevné hmoty (pozůstatky nespáleného paliva, nečistoty, pozůstatky hoření a další), která můžeme považovat za tzv. kondenzační jádra. Tato ledová jádra jsou postupně obalována námrazou z vodní páry a jelikož jejich stopa přímo koresponduje s trasou, kterou za sebou nechává letadlo, jsou pro obvyčejného pozorovatele na obloze viditelné jako čáry. Nevznikají však okamžitě po vypuštění z výfukového systému, jelikož kondenzační jádra mají příliš vysokou teplotu a vznikající námraza se na nich okamžitě taje. Táním se ovšem kondenzační jádro ochlazuje až na teplotu, při které se na jádru námraza drží a vznikají tak viditelné kondenzační čáry. Mezi další důležité podmínky tvorby kondenzačních čar patří okolní vlhkost vzduchu – **relativní (poměrná) vlhkost** [3, 4].

Relativní vlhkost je poměr skutečného obsahu vodních par (a) k maximálnímu možnému obsahu vodní páry (A) v daného objemu vzduchu za dané teploty okolního vzduchu. Je označována f a vyjádřena je v procenty. Pro výpočet používáme vzorec 1 [4, 5].

$$f = \frac{a}{A} \cdot 100\% \quad (1)$$

Může tedy nastat situace, kdy okolní vzduch není dostatečně vlhký (nízká relativní vlhkost) nebo nejsou splněny další podmínky pro tvorbu stop a kondenzační stopa sice vznikne, ale okamžitě po průletu letadla zaniká. Naopak při příznivých podmínkách (hlavně pokud je relativní vlhkost dostatečně velká) přetrvává kondenzační stopa po průletu letadla velmi dlouho. V případě dlouhé životnosti kondenzační stopy se po nějaké době stopa za letadlem postupně rozpadá a je rozfukována okolním pohybem vzduchu [4, 5].

2.2 Pozorování kondenzačních stop



Obr. 1: Čerstvá kondenzační stopa za letadlem [3]

Na obrázku 1 výše vidíme „čerstvou“ kondenzační stopu za letadlem. Tato stopa ještě není rozfukována okolními větry. Na obrázku 2 naopak vidíme starší kondenzační stopu,

kteřá je rozfoukána větrem. Barva takovýchto vyfocených kondenzačních stop je bílá, nicméně pozorovateli se může jevit i jako červená či černá. Tento jev je dán denní dobou, respektive úhlem, pod kterým na stopy dopadají sluneční paprsky. Obecně platí, že čím pozdější denní doba, tím tmavší se kondenzační stopy jeví. Například těsně před západem Slunce (nízká výška Slunce nad obzorem) se kondenzační stopy jeví jako černé (obrázek 3) [3].

Tvar, chování, způsob rozpadání a barva tedy závisí na mnoha parametrech a prakticky každá pozorovatelná kondenzační stopa je unikátní [3, 7].



Obr. 2: Rozfoukaná kondenzační stopa [3]

Jak již bylo řečeno, samotný vznik kondenzačních čar je ve velké míře závislý na meteorologických a technických podmínkách, v určitých dnech nemůžeme kondenzační stopy pozorovat vůbec, jelikož nejsou splněny podmínky meteorologické, naopak v jiné dny jsou kondenzační stopy pozorovatelné hojně a za všemi pozorovanými letadly. Mohou nastat i situace, kdy u části pozorovaných letadel kondenzační stopa viditelná není a u části letadel tato stopa viditelná je. Tento jev je většinou způsoben rozdílnými letovými hladinami, ve kterých letadla letí, přesněji rozdílnými meteorologickými podmínkami v těchto hladinách, případně podmínkami technologickými (například malé množství kondenzačních jader).

Výška, ve které letadla letí, není obyčejným pozorovatelem jednoduše určitelná, proto pro určování výšky určitého letadla a meteorologické situace v dané výšce používáme informace z vysílačů ADS-B (blíže v kapitole 4), případně můžeme použít i meteorologická data z jiných letadel, která letí ve stejné výšce a blízko sledovaného letadla, pokud nejsou meteorologická data pro daný let dostupná [3, 8, 9, 20].



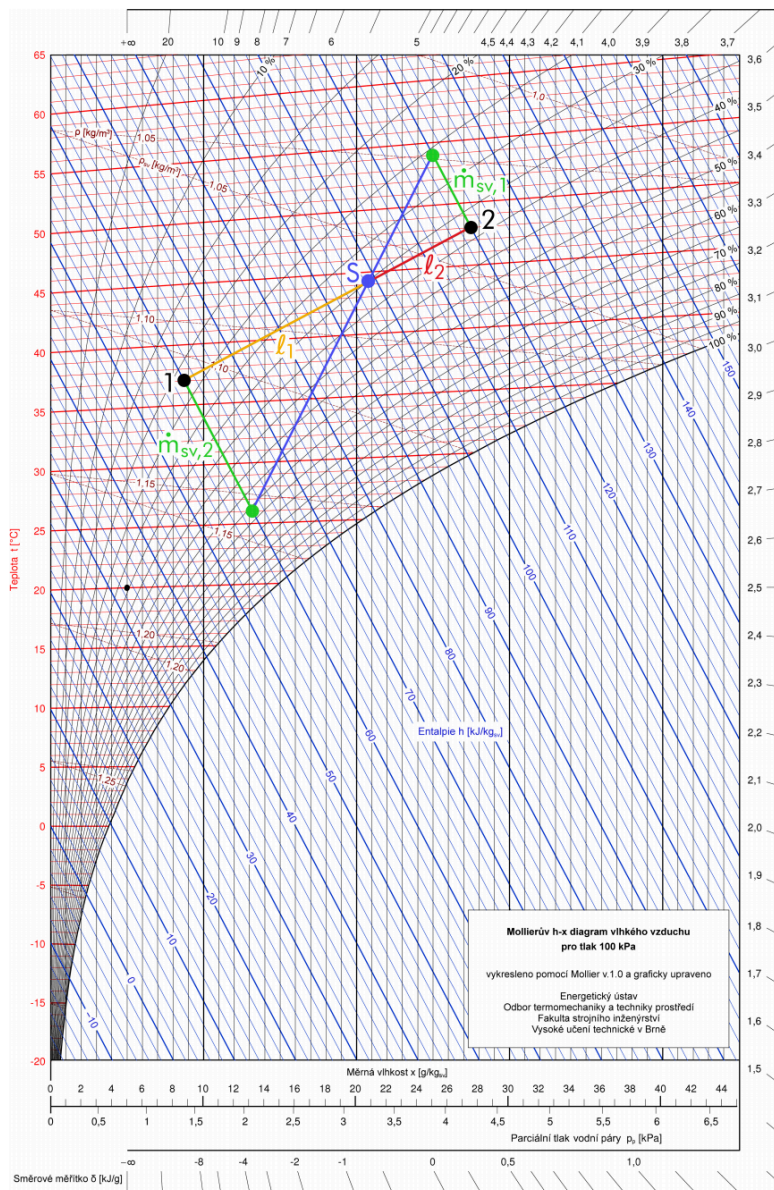
Obr. 3: „Černé“ kondenzační stopy na obloze, způsobené západem Slunce [3]

2.3 Schmidt – Applmenova teorie tvorby kondenzačních stop

Schmidt – Applmenova teorie určuje meteorologické podmínky pro vznik kondenzačních stop. Předpokládá míšení okolního vzduchu a výfukových plynů za letícím letadlem izobaricky.

Izobarické míšení probíhá následovně: Mějme dva proudy o hmotnostních průtocích m_1 a m_2 , teplotách t_1 a t_2 a měrných entalpiích h_1 a h_2 . Měrná entalpie je součet entalpií jednotlivých částí směsi vlhkého vzduchu (entalpie je fyzikální veličina, vyjadřující energii, uloženou v termodynamickém systému). Smícháním těchto proudů vznikne směs o hmotnostním průtoku $m_s = m_1 + m_2$, teplotě t_s a měrné entalpii h_s .

V Mollerově diagramu vlhkého vzduchu pro tlak 100 kPa (obr. 4) potom vidíme bod S, který určuje stav vzduchu po smíšení obou výše uvedených proudů [4, 5, 6].



Obr. 4: Mollerův h-x diagram vlhkého vzduchu pro tlak 100 kPa [4]

Kondenzační čára tedy vzniká smíšením velmi teplého a vlhkého vzduchu z výfukového systému motoru a suchého a studeného vzduchu okolního, přičemž okolní vzduch musí být plně nasycen tak, aby se voda v okolním vzduchu kondenzovala do vodních kapek a nikoli do malých, okamžitě mrznoucích krystalků. Po vychladnutí směsi po průletu letadla vzniklá (vodou nasycená) směs chladne a vznikají krystalky vody, které tvoří samotnou kondenzační stopu. Samotná stopa vzniká nejčastěji v řádu desítek metrů za letícím letadlem [4, 5, 6].

3 LETECKÉ RADARY

Úkolem diplomové práce je vytvoření databáze odpovědí palubních vysílačů a sekundárních přehledových radarů, které spolu komunikují v módu S. Nejdříve je ale nutno vysvětlit funkci radaru, používaných v letectví.

Letecké přehledové radary si můžeme obecně rozdělit na dvě podskupiny:

- Primární přehledový radar (PSR),
- Sekundární přehledový radar (SSR).

3.1 Primární přehledový radar (PSR)

Primární přehledový radar (dále jen PSR) funguje na jednoduchém principu odražení rádiových vln od cíle. Vysílač tohoto radaru vyšle rádiový paprsek a zaznamenává odrazy od letadel (v letecké terminologii je používán výraz „cílů“) a tyto odrazy posléze promítá na zobrazovací obrazovku řídicího/pozorovatele. PSR je také několik druhů, vesměs se liší použitím, dosahem a detailem zobrazení. PSR je používán pro činnosti, jako je například řízení pojiždění na velkých letištích, nebo vojenské sledování cílů, případně jako AWR (letadlový meteorologický radar) [10, 12].

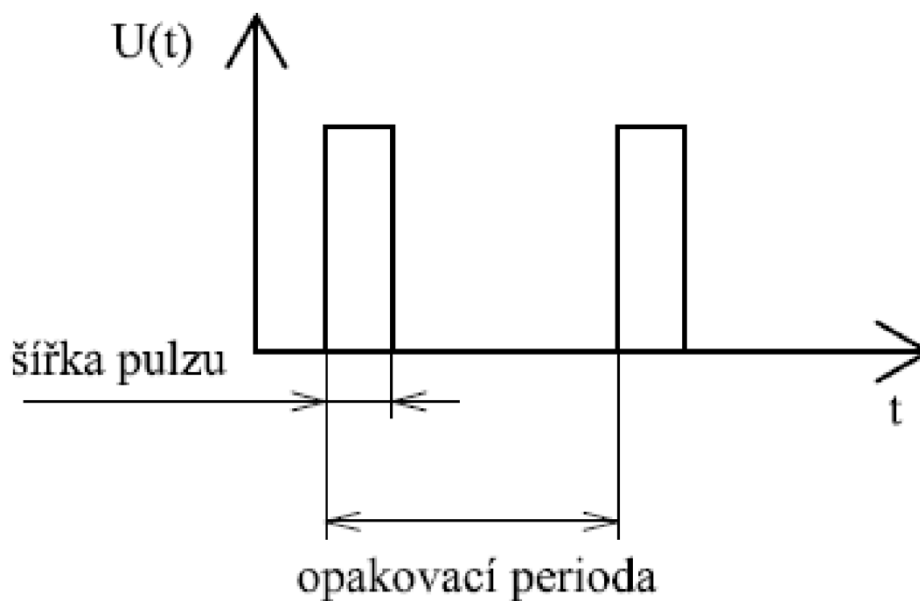
Tento druh radaru má ale relativně velké nevýhody – například potřebu výkonného vysílače. Standardní PSR taktéž dokáže zobrazit pouze údaje o poloze cíle, o jeho rychlosti a směru ve kterém letí, ne však detailnější informace o např. identifikaci letounu a jeho výšce. Pro výpočet dosahu radaru používáme *radiolokační rovnici*, viz vzorec 2.

$$r_{max} = \sqrt[4]{\frac{P_v D_v S_e S_p D}{(4\pi)^2 P_{p\ min}}} \quad (2)$$

V této rovnici vyjadřuje r_{max} maximální dosah radaru, P_v výkon impulsu, S_p efektivní plochu antény, $P_{p\ min}$ citlivost přijímače radiolokátoru, S_e plochu cíle a D_v funkci úhlu antény a cíle.

Z této rovnice nám po zjednodušení vyplývá, že chceme-li zvětšit dosah radaru 2x, musíme zvýšit jeho výkon 16x, případně zvětšit přijímací plochu přibližně dvojnásobně [10, 11, 13].

Na obrázku 5 níže jsou zobrazeny obálky vysokofrekvenčních signálů vysílače PSR.



Obr. 5: Obálky vysokofrekvenčních signálů vysílače PSR [21]

3.2 Sekundární přehledový radar (SSR)

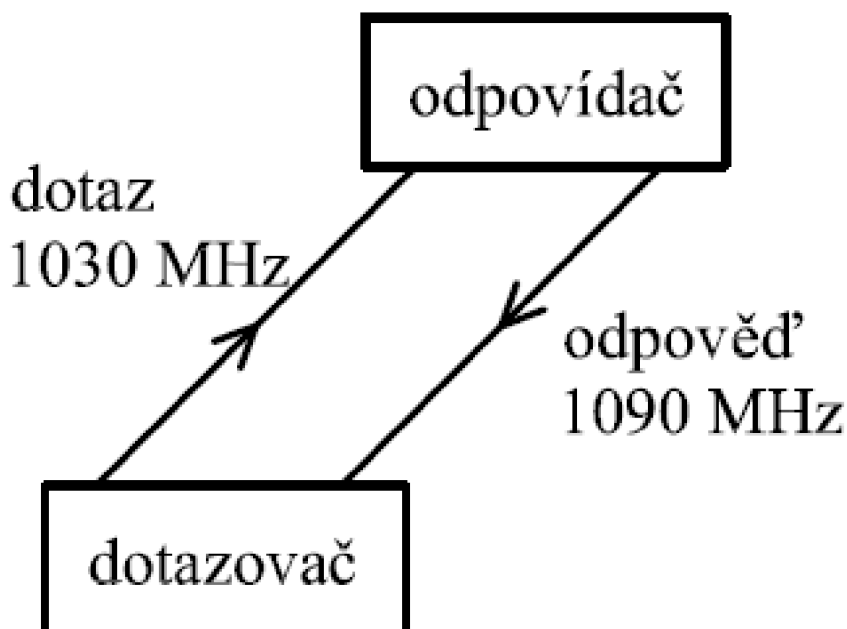
Sekundární přehledový radar (dále jen SSR), na rozdíl od PSR, nespolehá na odrazy od cíle, ale aktivně komunikuje s odpovídači (transpondéry), umístěnými na palubách letadel a jejich jednotlivými módy [10, 17].

Vysílač SSR pracuje v pásmu UKV, na frekvenci 1030 MHz. Vysílač pošle do svého okolí dotaz, letecké stanice tento dotaz přijmou, rozbalí a pošlou zpátky svoji odpověď. Odpověď může být různá, záleží na druhu dotazu a pracovním módu radaru a odpovídače. Zpráva je zpět k vysílači odeslána také v pásmu UKV, přesněji na frekvenci 1090 MHz. Přijímač (SSR) tuto odpověď přijme, přečte a výsledná data zobrazí na promítací obrazovku řídicího letového provozu [11, 17].

Odpovědi vysílané letadly jsou vesměs tří typů: **A**, **C**, **S**. Každý z módů má jiné průběhy, druhy a data odpovědí [21].

Rozlišujeme mezi komunikací adresnou a neadresnou – adresná komunikace probíhá v módu S, kdy je letoun radarem oslovován přímo. Při neadresné komunikaci není rozlišování letounů bráno v potaz a letouny na obrazovce rozlišujeme jen podle směru a vzdálenosti, ze které nám dojde odpověď. Na obrázku 6 níže je vyobrazena komunikace dotazovače s odpovídačem a k tomuto používané frekvence [14, 21].

Česká služba řízení letového provozu (v České republice zařizuje tuto službu ŘLP ČR s.p.) má v provozu tři SSR – Praha, Písek a BUKOP. Praha má dosah přibližně 170 NM, Písek 200 NM a BUKOP 200 NM [21].



Obr. 6: Komunikace dotazovač-odpovídač [21]

3.3 Neadresná komunikace sekundárního radaru SSR

Neadresná komunikace mezi letadlem a přehledovým radarem SSR se skládá ze dvou částí – z dotazu a odpovědi. Sekundární radar bereme jako řídicí prvek při komunikaci letadlo – radar a podle toho i definujeme jejich komunikaci [21, 22].

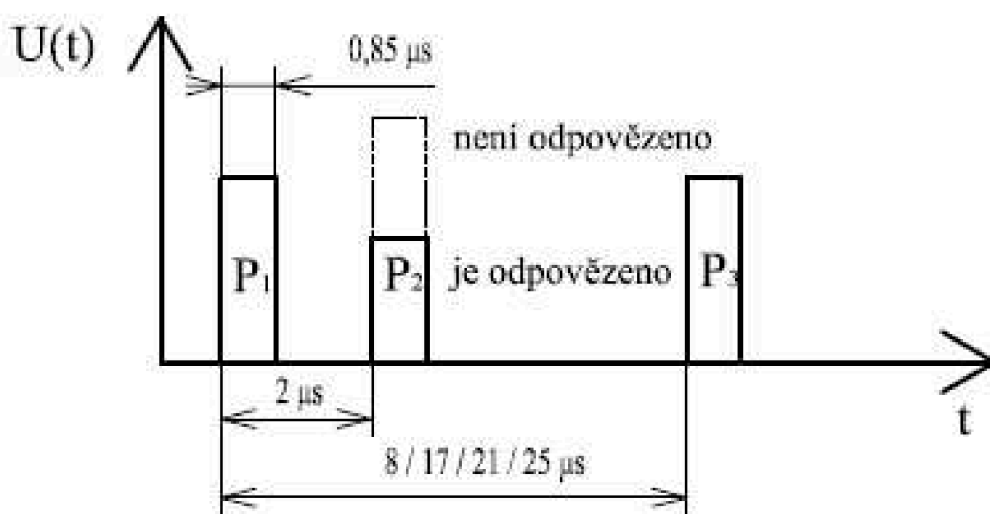
3.3.1 Dotaz

Dotaz je vysílán ze sekundárního radaru SSR a je přijímán přijímačem v letadle. Dotaz tvoří dva elektromagnetické impulsy P_1 a P_3 , které jsou vysílány po sobě a jejichž odstup určuje pracovní mód radaru. V současnosti jsou používány 4 pracovní módy – A, B, C, D. Liší se použitím a funkcí a jsou určeny rozestupem jednotlivých impulsů o příslušný počet mikrosekund [21, 22].

- Mód A je v komunikaci je používán pro sledování a identifikaci – 8 μs ,
- Mód B je v komunikaci je rezervován pro další využití – 17 μs ,
- Mód C je v komunikaci je používán pro automatické dotazování tlakové výšky letounu – 21 μs ,
- Mód D je v komunikaci je rezervován pro budoucí využití – 25 μs .

Při zjišťování módu dotazů je měřen rozestup mezi počátečními úvratěmi signálu P_1 a P_3 .

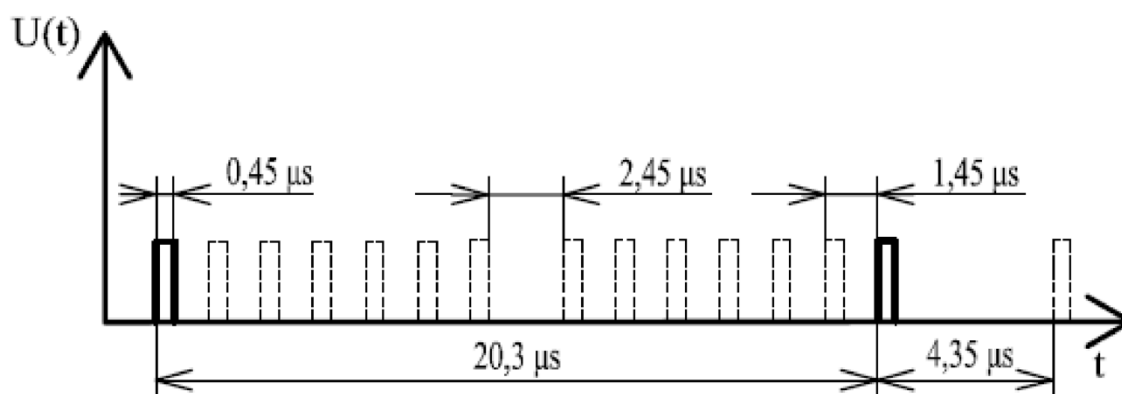
Dotaz všesměrového radaru SSR má ovšem na vyzařovacím diagramu kromě hlavního laloku i laloky vedlejší (postranní). Tyto laloky mohou „zmást“ odpovídač v letadle natolik, že na přijatý signál z těchto postranních laloků odpovídá, proto do komunikace mezi radarem a letadlem zavádíme i signál P_2 . Tento signál slouží k potlačení odpovědi na falešné dotazy postranních laloků a je vysílán 2 μs po signálu P_1 . Pokud je P_2 přijat s amplitudou větší nebo rovnou amplitudě signálu P_1 jedná se o dotaz od postranního laloku a není na něj odpovídačem odpovězeno. Vztah a časový průběh P signálů je zobrazen na obrázku 7 [21].



Obr. 7: Dotaz radaru SSR [21]

3.3.2 Odpověď

Odpověď letadla sekundárnímu radaru je tedy odeslána jen tehdy, když je amplituda signálu P_2 menší než amplituda signálu P_1 . Odpovídač letadla vyšle signál všesměrově a ten je zachycen a vyhodnocen sekundárním radarem [21].



Obr. 8: Odpověď na dotaz SSR [21]

Odpověď, vysílaná palubním odpovídačem je zobrazena na obrázku 8 výše. Vidíme na něm dva hlavní (bránové) impulsy, které jsou od sebe vzdáleny 20,3 μs . Mezi tyto hraniční impulsy je dle potřeby vloženo 12 informačních impulsů, které mohou nabývat pouze binárních hodnot (jejich hodnota může být pouze 0 nebo 1) [21, 31].

Tyto informační impulsy slouží pro přenos jednoduchých informací (kódů) a velikost přenášené informace může být maximálně 4096 bitů, jelikož jsou možné jen dvě hodnoty informačního impulsu (0/1) a máme dvanáct jednotlivých informačních impulsů = $2^{12} = 4096$ bitů informací [21, 31].

Při neadresné komunikaci radar – letadlo je využíváno především komunikace v módu A/C a to ve tvaru dotazu radaru „ACACACAC...“ a k tomuto korespondující odpovědi. V odpovědi palubního odpovídače jsou tedy informace o identifikaci letounu v módu A, vysílané po jednotlivých číslech („squawk“ letounu = číselný, informační kód, který je letounu přidělen službou řízení leteckého provozu) a v módu C informace o aktuální výšce letounu podle standardní tlakové atmosféry [21, 31].

Kódy squawk jsou důležité pro identifikaci letadel, které nemají k dispozici odpovídač, schopný odpovídat v adresném módu S. Letadlu je přidělen unikátní kód a s tímto kódem je nadále řízeno. Squawkem letadla je možno i hlásit nenadálé nebo nebezpečné situace na palubě letadel, nejznámějšími nouzovými kódy jsou [22]:

- 7500 – Aircraft hijacking – nezákonné vměšování do letu na palubě letadla,
- 7600 – Radio failure – porucha komunikačních prostředků na palubě letadla,
- 7700 – Emergency – obecný nouzový kód.

Po odvysílání informačních impulsů následuje druhý bránový impuls, kterým je přenos informačních impulsů ukončen. Za tímto uzavírajícím bránovým impulsem může následovat impuls typu „ident“, který je od něj vzdálen 4,35 μ s a je aktivován na vyžádání služeb řízení letového provozu. Slouží pro potvrzení navázání komunikace letadlo – řídicí a nejčastěji se aktivuje stisknutím tlačítka na odpovídači v pilotní kabině letounu. Signál „ident“ je potom vysílán po dobu 20 sekund [21].

3.4 Adresná komunikace sekundárního radaru SSR

Adresná komunikace je druh komunikace mezi radarem SSR a odpovídačem letadla, při které je odpovídačem letadla vysílána identifikace letadla (přidělená při registraci do leteckého rejstříku) se kterou přijímač radaru pracuje a zobrazuje ji (spolu s dalšími informacemi) na obrazovku řídicího letového provozu. Pro tento druh komunikace se využívá mód S palubního odpovídače a radaru, který ovšem (zvláště starší odpovídače) neumí používat a pracují tedy nadále v módu A/C [16, 17, 21].

S plošným zavedením tohoto módu se značně usnadnila práce řídicích letového provozu, kteří už nadále nemusí spoléhat jenom na přidělované a předávané kódy squawk a signál ident. Mód S byl také zaveden z důvodu občasného nebezpečného překrytí odpovědí letadel, která letí stejnou dráhu ale v rozdílných výškách. Radarový systém potom takovéto odpovědi vyhodnotil jako nesprávné a sledování takto letících letadel bylo prakticky nemožné [15, 17, 21].

3.4.1 Dotaz

Sekundární radar, používající adresnou komunikaci (tedy komunikující v módu S), používá pro komunikaci s letadly dva hlavní režimy, které se liší druhy dotazů [21]:

- Všeobecný dotaz typu A/C módu S,
- Všeobecný dotaz typu A/C/S módu S,
- Adresný dotaz módu S.

3.4.2 Všeobecný dotaz typu A/C módu S

Všeobecný dotaz typu A/C módu S se skládá se z impulsů P_1 , P_2 , P_3 a krátkého P_4 , přičemž P_2 je kontrolní signál, který zabraňuje odpovědím palubního odpovídače na vyzařované postranní laloky. Po odvysílání tohoto dotazu je odpovězeno všemi odpovídači, které pracují v módu A/C, odpovídače v módu S na tento dotaz neodpovídají [17, 18, 21].

3.4.3 Všeobecný dotaz typu A/C/S módu S

Tento typ dotazu se skládá se z impulsů P_1 , P_2 , P_3 a dlouhého P_4 . Po odvysílání tohoto dotazu odpovídají odpovídače všech letadel v dosahu – odpovídače typu S odesílají svoji 24bitovou adresu [17, 18, 21].

3.4.4 Adresný dotaz módu S

Adresný dotaz módu S se skládá se z impulsů P_1 , P_2 , P_6 a P_5 , kdy P_5 je kontrolní signál [17, 18, 21].

Má tři různé režimy použití:

- Všeobecný dotaz módu S – na tento dotaz odpovídají všechny odpovídače módu S, ostatní odpovídače mlčí,
- Adresný dotaz módu S – radar se dotazuje pouze letounu s určitou adresou a tento mu odpovídá, ostatní odpovídače mlčí. Taktéž nazýván „selektivní“,
- Rozhlasový dotaz módu S – SSR vysílá určité informace všem odpovídačům, které pracují v módu S, ale neočekává na ně odpověď.

3.4.5 Odpověď

Komunikační mód S umožňuje 25 různých druhů komunikace radar-odpovídač. Teoreticky je tedy možno dotazem zjistit 25 různých informací o dotazovaném letadle jakožto i jejich kombinaci [17, 18, 21].

Samotný přenos informací z palubního odpovídače se skládá z několika částí:

- 24bitová adresa pro identifikaci letounu,
- Informační část zprávy: 56/112/1280bitů informací z letounu pro řízení letového provozu.

Použití módu S umožňuje tedy rychlou a efektivní komunikaci mezi radarovým systémem a palubním odpovídačem. Řídící letového provozu a automatické systémy řízení jsou z komunikace schopny dostat například lepší informace o poloze letadla nebo aktuální meteorologické podmínky v sektoru, ve kterém se letadlo nachází, což přispívá k větší bezpečnosti leteckého provozu obecně [21].

3.5 Problémy v komunikaci radar – letadlo

Stejně jako všechny systémy, používané v letectví, i spojení pozemního radarového systému a palubního odpovídače vykazuje ve své činnosti určité chyby a nedostatky, jmenovitě potom [17, 21, 31]:

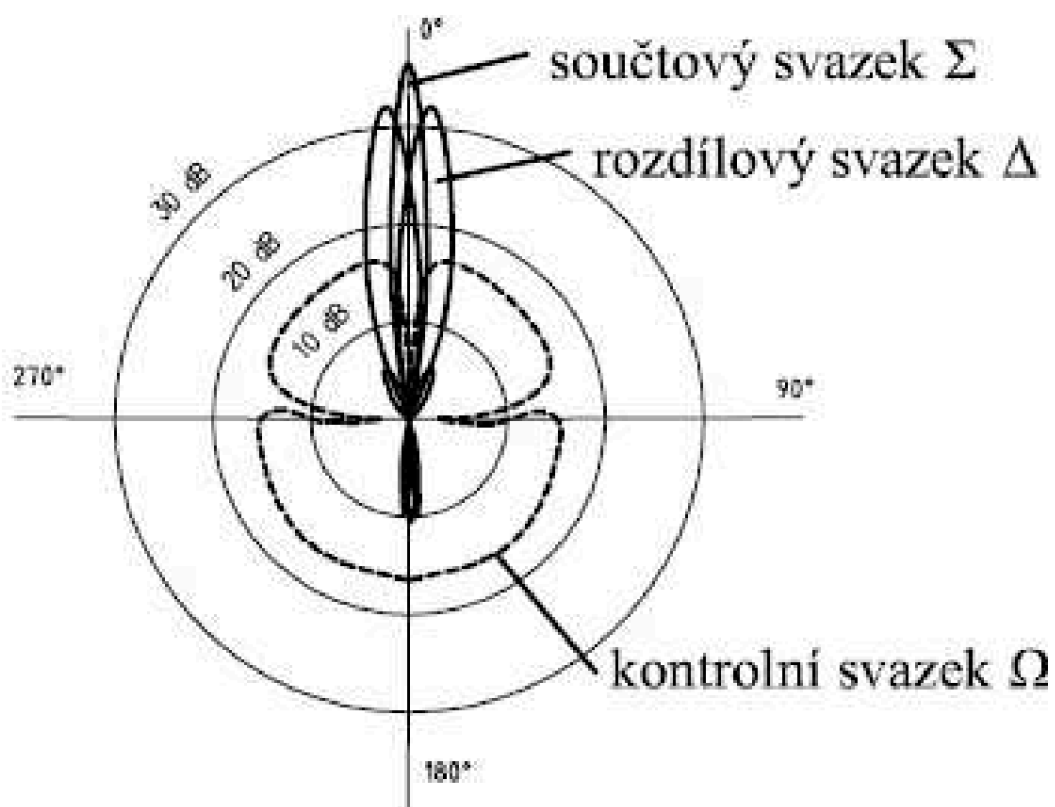
- Problematické určení azimutu,
- Odpovědi na dotazy z jiného zdroje,
- Překrývání odpovědí.

3.5.1 Problematické určení azimutu

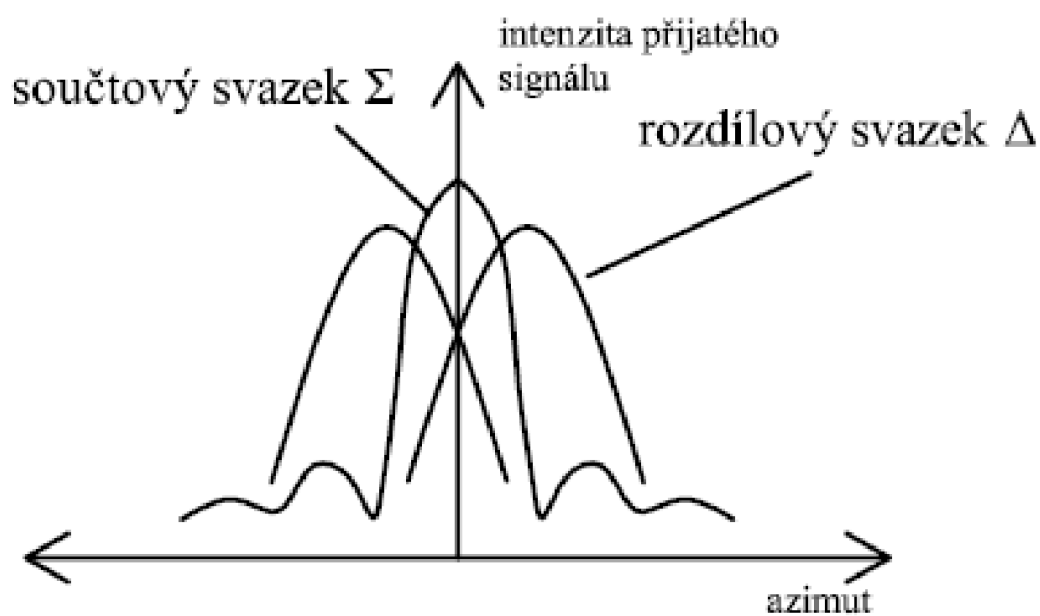
U sekundárního radaru je určení azimutu letadla jedním z nejdůležitějších úkolů. Bohužel, vzhledem k velké šířce vyzařovaného anténního svazku je toto určení někdy chybové. Jako řešení se nabízí používání MSSR – monopolzního sekundárního radaru, který chybové chování do značné míry eliminuje. Jeho vyzařovací diagram se skládá z celkem tří anténních svazků: součtového, rozdílového a kontrolního [11, 21].

Vyzařované laloky jsou umístěny do směru osy antény, přičemž symetrický s osou je svazek součtový. Rozdílové svazky mají nejlepší zisk v malých úhlech od této osy a se zmenšujícím se úhlem tento zisk klesá. Kontrolní svazek je de facto impulz P_2 , který slouží ke kontrole, zda odpověď na dotaz nebyla vyvolána postranními laloky. Čím více se intenzita přijatého signálu součtového a rozdílového svazku liší, tím přesněji můžeme změřit azimut letadla [21].

Na obrázcích 9 a 10 níže vidíme vyzařovací diagram pro jednotlivé laloky a diagram intenzity přijatého signálu.



Obr. 9: Vyzařovací diagram monopulzního sekundárního radaru [21]



Obr. 10: Zobrazení intenzity přijímaného signálu monopulzního sekundárního radaru [21]

3.5.2 Odpovědi na dotazy z jiného zdroje

Tato chyba souvisí s formou vyzařování odpovědí palubního odpovídače na dotazy radaru. Tyto odpovědi jsou totiž vyzařovány všesměrově a za jistých podmínek se může stát, že odpověď letadla, vyvolaná prvním radarem bude zachycena a zpracována radarem druhým. Nejedná se ovšem o stejnou situaci jako při vyvolání „falešných“ odpovědí postranními laloky radaru – v případě odpovědi na dotaz z jiného zdroje je totiž platnost odpovědi potvrzena (pomocí kontrolního impulsu P_2) a odpověď, vyvolaná jiným radarem, je brána jako odpověď platná [21].

Takovéto odpovědi označujeme jako FRUIT (False Response Un-correlated In Time, případně False Replies Unsynchronized with Interrogator Transmissions) a při použití moderních radarových systémů, vybavených digitálním zpracováním dat, se tyto chyby dají téměř stoprocentně odfiltrout [21, 22].

3.5.3 Překrývání odpovědí

Letí-li dvě letadla ve stejném letovém koridoru, stejným směrem v různé výšce, může nastat jev, který nazýváme „překrývání odpovědí“. Zjednodušeně – v neadresném módu odpovědi nastane překrytí odpovědí dvou letadel na dotaz radaru. Ten sice takovéto odpovědi přijme, ale musí je ignorovat a může potom docházet např. ke ztrátám spojení letadlo – radar. Tato chyba byla značně omezena zavedením adresného dotazování v módu S, kdy na dotaz radaru odpovídá jen konkrétní palubní odpovídač [21, 22].

4 ADS-B

ADS-B je radiokomunikační systém, který je používán pro přenos přesné polohy a dalších informací o letu, zejména identifikačních údajů. Zkratka ADS-B znamená „Automatic Dependent Surveillance: Broadcast“, tedy „systém přehledového automatického vysílání“. Jedná se tedy o radiokomunikační systém, který využívá frekvencí 978 MHz a 1090 MHz ke komunikaci a k samotnému zpracování dat využívá používá modulační schéma PPM (Pulse Position Modulation). Nejčastěji je používáno pásmo 1090 MHz, nicméně vzhledem k velkému provozu na tomto a v okolí tohoto pásma se používá i pásmo 978 MHz. Řadíme ho mezi tzv. systémy třetí generace – systémy, které značně zvyšují bezpečnost v leteckém provozu. Velkou výhodou je relativní nízkonákladovost a to především z důvodu jednoduchého a levného plánování letů a nutnosti minimálního pozemního vybavení [24].

Celý systém je plně automatický, v normálním stavu nevyžaduje žádné zásahy pilota, čímž se značně redukuje chybovost, zapříčiněná lidským faktorem [24].

4.1 Přenos informací pomocí ADS-B

Pro přenos ADS-B informací je používáno modulační schéma PPM, frekvence 978 MHz nebo 1090 MHz a funguje na principu zjišťování aktuální polohy letadla a odesílání této informace letadlům v okolí, přičemž značně omezuje chyby, nepřesnosti a časové prodlevy určování polohy pomocí komunikace radar-letadlo. Příjemcem vysílání ADS-B dat může být jak letadlo, které je vybaveno příslušným přijímacím systémem, tak pozemní stanice. Vzhledem k rostoucí amatérské komunitě nadšenců do letového provozu roste v posledních letech počet pozemních přijímačů a na jimi přijímaná data jsou vázány mnohé servery pro sledování nebo vyhodnocování letového provozu. Data jsou následně ukládána a zpracovávána pro různé účely (data pro webové stránky, zobrazující reálný provoz, tréninkové účely a další) [24, 26].

ADS-B se skládá ze dvou služeb:

- ADS-B OUT,
- ADS-B IN.

4.1.1 ADS-B OUT

Jedná se o službu ADS-B systému, která pravidelně vysílá informace o letu pomocí palubního vysílače okolním letadlům, nebo pozemním stanicím. Tento balíček informací obsahuje: polohu letadla (dle GPS), volací znak letadla, výšku a rychlosti

(horizontální a vertikální). Obecně se skládá ze dvou částí – palubního vysílače a GNSS jednotky. GNSS jednotka poskytne vysílači data (poloha, rychlost...), palubní vysílač data modularizuje pomocí schématu PPM a vyšle tyto informace do svého okolí. Za předpokladu bezchybné funkčnosti systému GNSS jsou tato data o letounu brána jako velmi přesná, přesnější než data z primárního nebo sekundárního radaru [24, 26, 29].

4.1.2 ADS-B IN

Služba systému ADS-B, zajišťující příjem informací z okolních vysílačů (příjem informací ze služby ADS-B OUT), zpracování přijatých dat do použitelné formy pro palubní počítač a nakonec zobrazení těchto dat na obrazovku pro pilota. Mimo to by měl být systém ADS-B IN schopen přijímat a zobrazovat data ze systémů FIS-B a TIS-B [24, 32].

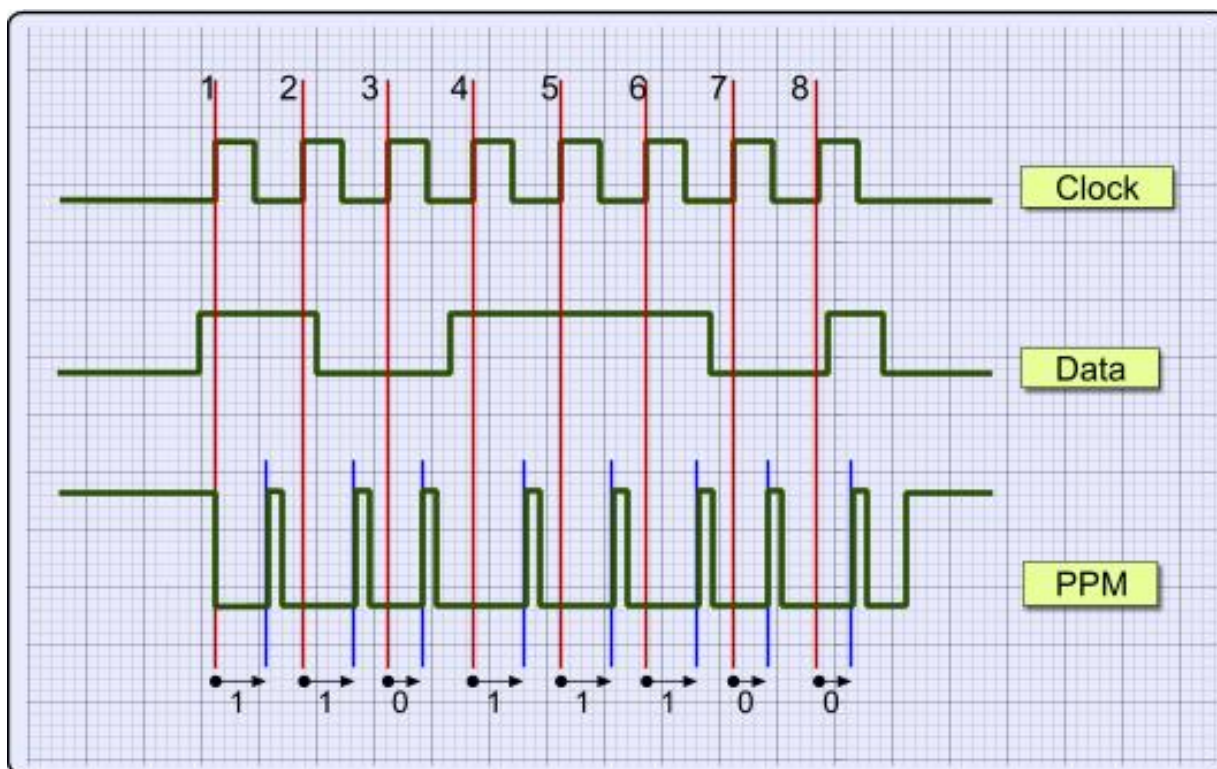
Data TIS-B jsou data od pozemních stanic, které přijímají, předávají si a vysílají informace ze systému ADS-B OUT okolnímu leteckému provozu. FIS-B předává na paluby letadel hlavně informace o počasí, taktéž za pomoci pozemních stanic. Oba tyto systémy jsou pro majitele letadla, osazeného ADS-B systémem, bezplatné [24, 32].

4.1.3 Modulační schéma PPM

Modulační schéma PPM funguje na jednoduchém principu modulace dat do krátkých impulsů, které jsou stejně široké a mají stejnou amplitudu, liší se však vzájemnou vzdáleností. Impulsy jsou následně přenášeny analogově nebo digitálně. Mimo ADS-B je toto schéma používáno například pro dálkové ovládání a přenos informací optickými kabely. První pokusy a počátky zavádění tohoto druhu modulace se datují do 70. a 80. let 20. století [23].

PPM je využíváno z několika hlavních důvodů – přenos pomocí PPM je úsporný, jednoduchý, rychlý a překvapivě spolehlivý. Vedlejším důvodem by byly náklady na zavedení nového způsobu modulace a přenosu dat, jelikož například certifikační náklady v letectví by byly pro takovýto systém enormní [23].

Na obrázku 11 níže vidíme modulaci dat podle do PPM. K modularizaci potřebujeme dva signály – signál časoměrného čipu (na obrázku jako vrchní signál „Clock“) a data, která chceme přenášet (prostřední křivka „Data“). Na spodní křivce (výsledný signál, modularizovaný do PPM) potom čteme binární data (0/1), kdy velký časový interval mezi jednotlivými impulsy znamená 1 a krátký znamená 0. Časové intervaly se liší podle výrobce – například firma Sony (ve svém protokolu „Sony SIRC Protokol“) používá časový interval 1.8 milisekundy pro vyjádření hodnoty 1. Obdobně funguje modularizace ADS-B dat [23].



Obr. 11: Modulace dat do schématu PPM pro přenos [23]

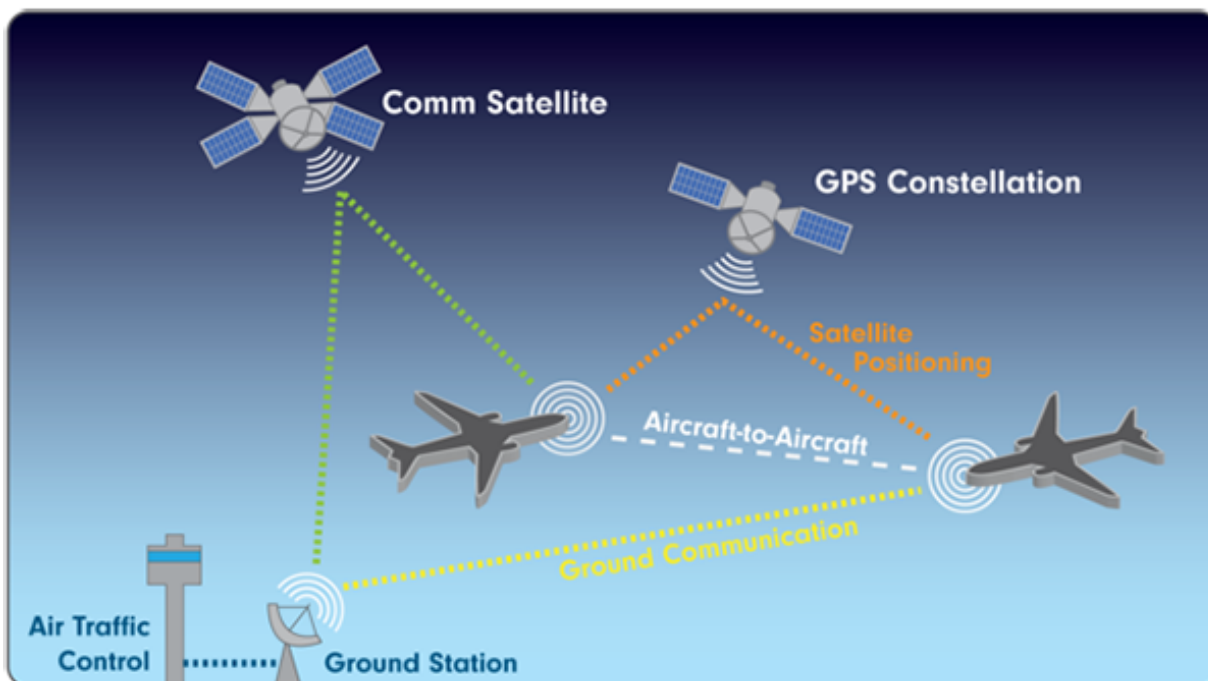
4.2 Vybavení pro ADS-B

Systém ADS-B se skládá z pozemní a palubní části – v této kapitole jsou shrnuty požadavky na palubní a pozemní vybavení. Zjednodušeně tedy – v **palubní části** fungují dva samostatné systémy – ADS-B OUT a ADS-B IN. Systém ADS-B OUT se skládá ze systému, který je schopný poskytnout GNSS (nejčastěji GPS) data a vysílače (vysílajícího na 1090 MHz, někdy označovaný jako 1090ES), který tato data zpracovává a vysílá je okolním letounům a pozemním stanicím. Systém ADS-B IN se skládá z přijímače dat od okolních letadel, pozemních stanic, nebo satelitních systémů, který přichází data přijme a zpracuje do zobrazitelné podoby na elektronickou obrazovku v kabině letadla [25].

Pozemní část se skládá z pozemní přijímací/vysílací stanice, které vysílají ADS-B data několika různým konzumentům těchto dat: službám řízení letového provozu, letadlům, satelitům (satelity distribuují informace leteckým společnostem, nebo službám řízení letového provozu) a ostatním pozemním stanicím [25, 26].

Na obrázku 12 můžeme vidět zjednodušené zobrazení komunikace mezi jednotlivými částmi ADS-B systému. Letadlo získá ADS-B data ze GPS satelitů (GPS Constellation), která distribuují okolnímu provozu (Aircraft-to-Aircraft) a pozemním stanicím (Ground

Communication ke Ground Station), které tato data distribuují službám řízení letového provozu (Air Traffic Control), satelitním systémům (Comm Satellite) a okolnímu letovému provozu (opět přes Ground Communication link) [25, 26].



Obr. 12: Předávání dat pomocí ADS-B vysílačů/přijímačů [43]

4.2.1 Povinnost ADS-B vybavení v Evropě

Majitelé letadel, zapsaných v EU a majících povinnost řídit se požadavky evropské agentury pro bezpečnost letectví (EASA, European Aviation Safety Agency) nejsou, podle výzkumu agentury SESAR, připraveni na zavedení povinného vybavení pro ADS-B v evropském vzdušném prostoru [26].

Podle výše uvedeného průzkumu bude k 7. červnu 2020 připraveno, podle nařízení EU č. 1207/2011 ze dne 22. listopadu 2011, pouze přibližně 20 % letadel, na která se toto nařízení vztahuje. Letadla, těžší než 5700 kg, mající maximální cestovní rychlost vyšší než 250 kt a operující podle pravidel IFR, musí být vybavena systémy pro vysílání ADS-B OUT dat. Tato skutečnost je ovšem pro mnohé majitele a provozovatele letadel dosti finančně, technicky a personálně náročná, proto vznikají různá zdržení a u některých aerolinek hrozí, že část jejich letadel nebude moci po tomto datu vyletět s cestujícími legálně. Náklady na kompletní vybavení komerčního letadla se podle EASA pohybují mezi \$60 000 a \$530 000 a EASA pořádá od roku 2014 školení pro majitele a provozovatele letadel, kde o tématice zavedení

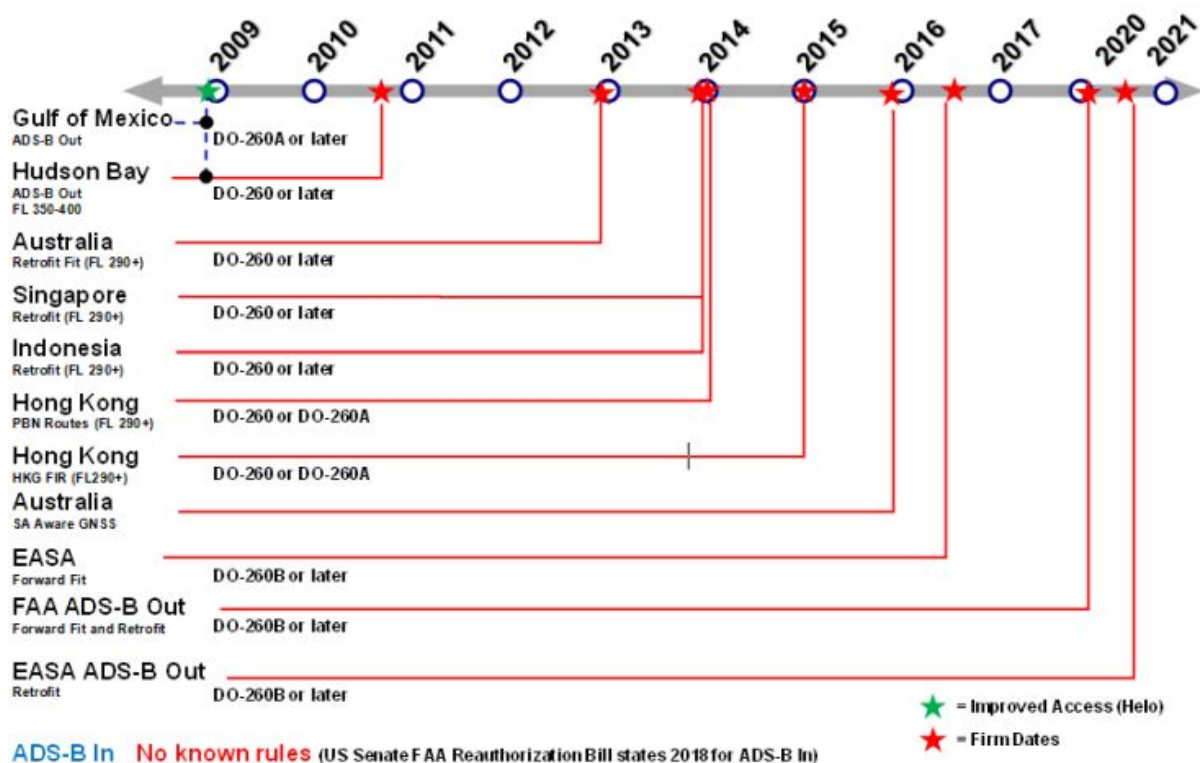
a užívání ADS-B školí. Pro nová letadla, vyrobená po 8. červnu 2016, patřící do výše uvedené kategorie (IFR, <5700 kg, <250 kt), platí povinnost být ADS-B vysílači vybavena už od výroby. Tato povinnost se vztahuje i na letadla vojenská, policejní, celní a podobné, nicméně u těchto letadel je možnost výjimky, kterou může vydat ÚCL [26].

V průzkumu se zkoumalo vybavení více než tři tisíc letadel a bylo zjištěno, že pouze 20 % z tohoto zkoumaného vzorku je vybaveno DO-260B ADS-B odpovídači (určitá část letadel byla vybavena také staršími odpovídači DO-260 a DO-260A, které se od DO-260B liší například způsobem kódování zpráv), které budou od 7. června povinné pro všechna výše zmíněná letadla. [26]

4.2.2 Povinnost ADS-B vybavení v USA a ve světě

Podobně jako EASA, FAA plánuje zavedení povinnosti mít letadlo vybaveno systémem ADS-B OUT. Dle federálních nařízení 14 CFR § 91.225 a 14 CFR § 91.227 jsou všechna letadla, operující v prostorech třídy A, povinna splňovat nároky na provoz systému ADS-B OUT. To znamená, že musí být vybavena příslušnými odpovídači a GNSS vybavením, aby jim bylo umožněno v prostorech třídy A létat. Nařízení nabývá účinnosti od 1. ledna 2020, tedy necelých šest měsíců před povinností mít toto vybavení i při letech v EU [26, 44].

V ostatních světových státech jsou podobná nařízení na vybavení letadel systémy ADS-B také postupně zaváděna, na obrázku 13 níže jsou vyobrazena plánovaná data zavádění v jednotlivých státech/prostorech.



Obr. 13: Plánovaná data zavádění povinnosti osazení letadel systémy ADS-B [44]

4.3 Amatérské využití ADS-B

ADS-B data nejsou nikterak složitě šifrována, ani tajena. Tato skutečnost dala vzniknout velké fanouškovské základně amatérských konzumentů ADS-B vysílání, díky které vzniklo mnoho velice zajímavých projektů a služeb, které jsou popisovány v následujících odstavcích.

Pojem „amatérské využití“ je nutno definovat – amatérské využití ADS-B je takové využití, kdy lidé, nemající přístup k certifikovanému a profesionálnímu pozemnímu vybavení služeb řízení letového provozu, konstruují z volně dostupných součástek vlastní ADS-B přijímače a data dále zpracovávají. Nejedná se tedy o lidi, kteří by nebyli spojeni s letectvím (ostatně – mnoho z nich pracuje pro řízení letového provozu, případně létá samo), jde o fakt, že pro příjem používají povětšinou podomácku vyrobené přijímače. Výsledná data zpracovávají do mnoha různých podob, nejčastěji pro služby on-line sledování světového leteckého provozu, jako je např. Flightradar24 nebo FlightAware. Tyto služby poskytují široké veřejnosti možnost sledování letového provozu téměř v živém přenosu, spolu s informacemi o letadle, trase, počasí atp. Některé z těchto služeb jsou placené, případně omezené pro nezaregistrované uživatele [28].

Chceme-li přijímat data z ADS-B vysílačů letadel, musíme pro to splnit tři hlavní podmínky, a to:

- Hardwarové vybavení,
- Softwarové vybavení,
- Správná poloha přijímače.

4.3.1 Hardwarové vybavení pro amatérský příjem ADS-B dat

1. Anténa
2. Filtr
3. Kabeláž
4. Zesilovač
5. Přijímač vysokofrekvenčního signálu
6. Výpočetní systém

Anténa

Anténa je pasivní prvek pro příjem signálu. Nad územím České republiky je pro přenos ADS-B dat využíváno téměř výhradně kmitočtové pásmo 1090 MHz a vertikální polarizaci pro vysílání dat ADS-B OUT, anténa tedy musí být schopná přijímat kmitočet 1090 MHz. Směrnost antény je na našem uvážení a předpokládaném využití. Chceme-li přijímat signál ze všech směrů, použijeme anténu všesměrovou, chceme-li přijímat data pouze z jednoho směru, použijeme anténu směrovou. Impedance by měla být 50 Ω nebo 75 Ω , přičemž bychom měli pro napojení použít koaxiální kabel o shodné impedanci, aby nedošlo ke ztrátě kvality přijímaného signálu [28].

Velice důležité je také umístění antény. Všesměrová anténa by měla být umístěna co nejvýše a pokud možno nad okolní terén. Jsou však rušivé elementy, které se nedají umístěním antény odstranit – zejména se jedná o okolní silné vysílače, které vysílají v okolí kmitočtového pásma 1090 MHz. Jedná se hlavně o vysílače GNS, DVB-T a LTE signálu, které mohou příjem ADS-B signálu rušit. Částečným řešením je kvalitní filtr s pásmovou propustí 1090 MHz, nicméně kompletně toto rušení odfiltrovat je v blízkosti vysílačů téměř nemožné a provozovatel amatérské stanice se tedy musí spokojit s méně kvalitním příjmem [28].

Filtr

V případě externě rušeného příjmu antény (tj. přijímáme rušený signál a rušení nevzniká v anténovém/výpočetním okruhu jako takovém) je nutno použít filtr, který nám určitý druh rušení odfiltruje. Příkladem použití by bylo výše zmíněné užití antény v blízkosti vysílače LTE,

který pracuje s podobnými kmitočty a zvolením vhodného filtru můžeme toto rušení odfiltrovat [28].

Kabeláž

Pro kabeláž používáme koaxiálního kabelu, který se standardně používá pro vedení televizního/rádiového signálu od antény do přijímače. Existuje několik druhů koaxiálního kabelu, pro naše využití je však vhodné použít kvalitnějšího, vysokofrekvenčního koaxiálního kabelu pro vedení signálu z antény do filtru a dále do přijímače. Impedance kabelu by měla odpovídat impedanci antény a vzhledem k tomu, že kabel se chová de facto jako anténa, je vhodné ho co nejvíce zkrátit [28].

Zesilovač

Zesilovač je aktivním prvkem přijímacího obvodu ADS-B, nicméně není vždy prvkem nutným. Zesilující prvek v obvodu je schopen zesílit přijatý signál a umožnit tak jeho lepší zpracování přijímačem vysokofrekvenčního signálu. Při stavbě vlastní ADS-B stanice můžeme použít buď širokopásmového vysokofrekvenčního zesilovače, nebo zesilovače s předintegrovaným filtrem na 1090 MHz [28].

Jelikož jde o prvek aktivní, musí být pro svou činnost napájen. O napájení se stará buď koaxiální kabel (který je napájen z rozvodové sítě), rozvodová síť nebo teoreticky i napájení po ethernet kabelu (tzv. Power Over Ethernet, POE) a impedance zesilovacího prvku ADS-B přijímacího obvodu by měla být shodná s impedancí kabeláže a antény [28].

Přijímač vysokofrekvenčního signálu

Slouží k samotnému přijetí a zpracování signálu pro vstup do softwarové části amatérského řešení pro přijímání dat o ADS-B provozu. Přijímačů existuje nepřeberné množství druhů, ale jedním z nejrozšířenějších druhů je použití „obyčejného“ USB DVB-T přijímače spolu se správným softwarem pro příjem přijímačem zpracovaného signálu, DVB-T přijímače totiž přijímají (krom televizního a rádiového signálu) také kmitočty 24 MHz – 1766 MHz. Tyto USB přijímače jsou jednoduché, levné a dostupné (obvyklá cena se pohybuje v rozmezí 100–300 Kč a dají se sehnat v každém větším obchodě s elektronikou) a jsou velice dobře softwarově podporované [28].

Připojení k výpočetnímu systému je v případě použití DVB-T přijímače řešeno pomocí USB.

Výpočetní systém

Výpočetním systémem je v celé soustavě obvykle PC, případně chytrý telefon. Musí splňovat nějaké základní podmínky – mít možnost připojení přijímače (nejčastěji to znamená USB port), mít podporovaný operační systém (tedy OS, pro který jsou dostupné ovladače, kompatibilní se zvoleným přijímačem) a podporovat softwarovou část ADS-B soustavy. Používaný operační systém bývá povětšinou Windows, případně větší Linux distribuce. Často se využívá různých IoT soustav – zejména počítačů od výrobce Raspberry, které jsou výhodné zejména z důvodu malých rozměrů, nízké ceny a nízké spotřeby elektrické energie [28].

4.3.2 Softwarové vybavení pro amatérský příjem ADS-B dat

Součástí softwarového vybavení pro příjem ADS-B dat jsou čtyři hlavní části:

- Operační systém počítače,
- Ovladače pro přijímač vysokofrekvenčního signálu,
- Program pro zpracování dat z přijímače,
- Nastavení programu pro zpracování dat.

Existuje několik variant **operačních systémů**, pod kterými je možno domácí přijímací ADS-B stanici zprovoznit. Nejčastěji se jedná o operační systém Windows v některé ze svých variant, případně o operační systémy na bázi linuxového jádra (což zahrnuje i operační systémy výše zmíněných počítačů Raspberry) [28].

Ovladače jsou systémovou komponentou, která propojuje hardwarovou a softwarovou část všech moderních komponent počítačových systémů. Běžný uživatel si může ovladač(e) představit jako soubor instrukcí pro operační systém počítače, které mu říkají, jak má s daty z hardwarové komponenty pracovat [28].

Pro příklad – máme výše zmíněný USB DVB-T ADS-B přijímač, poprvé připojený k počítači. Uživatel musí stáhnout a nainstalovat vhodné ovladače, které obsahují instrukce k zařízení (USB přijímači) pro zpracování dat, které přicházejí do přijímače z antény. Ve chvíli, kdy uživatel ovladače nainstaluje, začne počítač správně pracovat s připojeným USB přijímačem a přijímač může být použit v programu pro přijetí a zpracování dat [28].

Jako **program pro přijetí a zpracování dat** z přijímače máme opět na výběr z několika variant. U linuxových i windowsových distribucí je jedním z neoblíbenějších programů program „dump1090“. Jedná se o open-source distribuci, jejíž zdrojové kódy jsou dostupné pomocí distribuovaného systému správy verzí Git [28].

Nastavení programu pro zpracování dat se nedá jednoznačně specifikovat, jelikož vždy záleží na použitém programu, ovladačích, zařízení pro příjem a operačním systému. Obecně můžeme nastavením (povětšinou se funkce programu nastavuje pomocí konzolových příkazů nebo konfiguračním souborem, který program načte) nastavit například limitaci dosahu zobrazovaných dat (data budou zobrazována pouze do určitého radiusu od přijímače), nebo třeba nastavení složky pro export raw dat, které z přijímače dostáváme [28].

4.3.3 Sdílení ADS-B dat

Po softwarovém dekódování dat pomocí například „dump1090“ je možno datové balíčky sdílet lidem a službám pomocí internetu. Pro správné sdílení a odesílání dat je většinou nutná instalace specializovaného softwaru, který je vydáván serverem, na který chcete data sdílet. Data jsou nejčastěji sdílána na servery FlightRadar24 a FlightAware a za sdílení ADS-B dat získá uživatel prémiový přístup ke službám těchto serverů, které jsou jinak zpoplatněné [28].

5 DEFINICE, POPIS A STRUKTURA PŘIJÍMANÝCH DAT

Data můžeme popsat jako soubor údajů, které používáme pro popis chování, vlastnosti, stavu, nebo jevu vybraného objektu. ČVUT dostává data o letovém provozu ze svých ADS-B přijímačů, umístěných v Praze.

Data jsou několika druhů a jedná se o informace z palubního systému ADS-B, doplněné o některé meteorologické informace (nejsou dostupné z většiny letadel).

V tabulce, která je po průletu letadla v dosahu ADS-B přijímačů vytvořena, jsou o letounu tato data:

- Unix time,
- Message,
- Time,
- DF – Downlink Format,
- Type code/BDS number,
- Latitude,
- Longitude,
- Altitude,
- Ground speed,
- Baro difference,
- Heading,
- TAS,
- Mach speed,
- FOM/Source,
- Wind speed,
- Wind direction,
- Temperature,
- Pressure,
- Turbulence,
- Humidity.

V následujících podkapitolách je detailně popsáno, co konkrétní typy dat znamenají a jaké informace obsahují. Při popisu je také referováno k tabulce/tabulkám – toto jsou tabulky, vygenerované ČVUT z přijatých ADS-B dat a tyto tabulky jsou nahrávány do databáze a je s nimi ve webové aplikaci nadále pracováno.

5.1 Unix time

Unix time, někdy taky označován jako „unixový čas“ nebo „unix timestamp“, je způsob zápisu času, často používaný v počítačových systémech pro svoji jednoduchost a univerzálnost. Hodnota unixového času je vyjádřena celým číslem a vyjadřuje počet sekund, které uběhly od 1. ledna 1970 00:00:00. Například hodnota 1550419361 vyjadřuje datum 17. února 2019 16:02:41. Velkou výhodou je jednoduchost práce s tímto časovým formátem v programování a vývoji softwaru, jelikož jednotně vyjádřený čas lze např. jednoduše odečíst od jiné hodnoty unixového času, abychom dostali časový rozdíl. Také získáváme výhodu při práci s daty z jednotlivých zemí, kdy je formát času často zapisován jinak – například „mm-dd-yyyy“ oproti českému stylu zápisu „dd-mm-yyyy“ [48].

Příklad dat o unixovém čase, které jsou ve vygenerované tabulce k dispozici jsou vyobrazeny na obrázku 14.

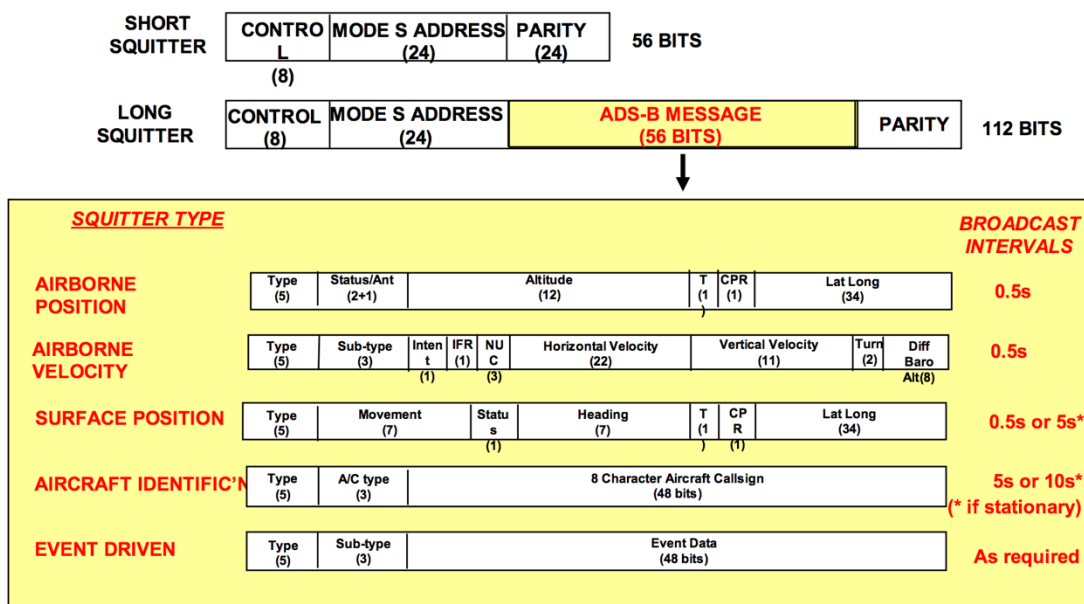
Unix_time
1535777068756
1535777069117
1535777069311
1535777069553
1535777069561
1535777069802
1535777069807

Obr. 14: Příklad unixových dat

5.2 Message

ADS-B message („zpráva ADS-B“) je soubor dat od palubního systému ADS-B OUT, ve kterém jsou zakódovány různé informace o letu a letadlu samotném. Zpráva může být několika druhů, lišících se podle datového obsahu zpráv. Určité typy jsou také vysílány v určité časy a v posloupnosti, jak je ukázáno na obrázku 15 níže [29].

MODE S EXTENDED SQUITTER



connecting australian aviation

Obr. 15: Druhy informací, obsažených v ADS-B zprávě a jejich intervaly vysílání [29]

Výše zobrazené typy ADS-B zpráv jsou zprávy o:

- Airborne position – zpráva o pozici letadla – obsahuje hlavně základní přehledové informace o výšce a GNSS souřadnicích vysílajícího letounu (tedy jeho 3D pozici v prostoru) a čas vysílání. Je vysílána každé 0.5 sekundy,
- Airborne velocity – zpráva o rychlostech letadla – obsahuje informace o horizontální a vertikální rychlosti letadla a rozdílu tlaku vzhledem k předešle vysílanému barometrickému tlaku, vysílána každé 0.5 sekundy,
- Surface position – vektorová pozice – obsahuje informace o polohovém vektoru letounu a GNSS souřadnice, vysílána po 0.5 sekundy nebo po 5 sekundách, pokud není zaznamenán pohyb stroje,
- Aircraft identification – identifikace letadla – obsahuje informaci pro identifikaci letadla – jeho volací znak. Vysílána každých 5 sekund (10 sekund, pokud se letadlo nehýbe),
- Event driven – zpráva, vyvolaná situací – obsahuje další potřebné informace o letu a je vysílána nepravidelně, pokud je třeba.

Příklad zpráv typu „message“, které jsou v tabulkách k dispozici:

Message
8D4CACAB60BBF548C8DEED3E70D6
8D4CACAB994071B3002021619F71
8D4CACAB60BD01D902F35D1B357B
8D4CACAB994071B3002021619F71
8D4CACAB994071B3002021619F71
8D4CACAB60BD01D8E2F362EBC9DE
8D4CACAB60BD01D8E2F362EBC9DE
8D4CACAB994071B300242159A971
8D4CACAB60BD01D8C2F366AE8DC1
8D4CACAB994071B300242159A971

Obr. 16: Ukázka dat typu „message“

5.3 Time

Data o aktuálním čase při odesílání ADS-B zprávy, dešifrované z unix timestampu. Jsou ve formátu „yyyymmdd_hh:mm:ss“. Z pohledu výpočetních systémů se s tímto formátem časových dat pracuje velmi špatně [48].

Time
20180901_04:44:28
20180901_04:44:29
20180901_04:44:29
20180901_04:44:29
20180901_04:44:29
20180901_04:44:29

Obr. 17: Ukázka dat typu „Time“

5.4 DF - downlink format

Tento datový typ udává formát vysílání, ve kterém letadlová stanice vysílá a jakého je tento formát je typu. Například, je-li ADS-B zpráva odeslána z odpovídače, pracujícím v módu S, je DF = 17. Pokud byla zpráva odeslána zpráva ze samostatného systému, který není založen na palubním odpovídači nebo se jedná o systém TIS-B, je DF = 18 [1].

Podle této hodnoty je pozemní systém a systém okolních letadel schopen zjistit, jaké vybavení se nachází na letadle, se kterým komunikují a upravit podle toho formát zprávy pro tuto stanici tak, aby jí byla rozuměla [1].

5.5 Type code/BDS number

Datový typ, udávající druh ADS-B zprávy, se kterou pracujeme. Některé z druhů zpráv, které pomocí této informace rozeznáváme jsou zobrazeny na obrázku 18 [41].

BDS 2,0	Aircraft identification
BDS 2,1	Aircraft and airline registration markings
BDS 4,0	Selected vertical intention
BDS 4,4	Meteorological routine air report
BDS 5,0	Track and turn report
BDS 6,0	Heading and speed report

Obr. 18: Některé číselné označení typu ADS-B zprávy [41]

5.6 Latitude

Data o zeměpisné šířce. ADS-B získává tyto informace z GNSS systémů, nejčastěji z družicového systému GPS. Zeměpisná délka určuje polohu na povrchu země na západ, nebo východ od nultého poledníku [30].

5.7 Longitude

Data o zeměpisné délce. Stejně jako u zeměpisné šířky (latitude) získává ADS-B tyto informace ze satelitních GNSS systémů. Zeměpisnou délkou určujeme polohu na sever, nebo jih od zemského rovníku [30].

Při dekódování zpráv se setkáváme s tím, že v některých zprávách latitude/longitude chybí – je to způsobeno tím, že ADS-B používá pro komunikaci velké množství typů zpráv a ne všechny zprávy tyto informace obsahují. O typech zpráv a jejich rozdílech je blíže pojednáváno v kapitolách „5.2 Message“ a „5.5 Type code/BDS number“. Na obrázku 19 níže je tento jev (střídání typů zpráv, které neobsahují všechny informace najednou) jasně viditelný.

5.8 Altitude

Údaj o aktuální výšce letadla. V letectví může znamenat barometrickou výšku letadla nad střední hladinou moře, výšku nad povrchem země, zjištěnou například radiovým výškometrem nebo jiný typ barometrické výšky. V datech systému ADS-B je zásadně

používána barometrická výška, vztahující se ke střední hladině moře a to hlavně kvůli své univerzálnosti.

Data o aktuální výšce letu jsou nejčastěji posílána ve zprávě „*Airborne position*“ (viz kapitola „5.2 Message“) spolu s informacemi o zeměpisných souřadnicích (latitude/longituds) [30, 31].

Latitude	Longitude	Altitude
50.7727	14.3540	36575
50.7715	14.3545	36600
50.7708	14.3548	36600
50.7708	14.3548	36600
50.7701	14.3551	36600
50.7689	14.3556	36600

Obr. 19: Příklad dat o zeměpisné šířce, délce a výšce

5.9 Ground speed

Ground speed (GS) je údaj o rychlosti letadla relativně vůči zemskému povrchu. Udává nám tedy, jak rychle se letadlo vůči zemi pohybuje. Mění se s mnoha dynamickými parametry letu – tahem motorů, silou větru, úhlem stoupání stroje (stoupá-li čistě kolmo, je GS nulová) a dalšími [17].

Ground speed pro ADS-B systém určujeme z použitého GNSS systému a informace o této rychlosti je opět dostupná jen v určitých typech zpráv [26].

5.10 Baro difference

Jedná se o rozdíl barometrické výšky a výšky nad geoidem, kterou poskytuje GPS. Vysílána je ve zprávě „*Airborne velocity*“ (viz. „5.1.2 Message“).

5.11 Heading

Heading nám udává, jakým směrem je na stupnici 360 stupňů směřován nos letounu. Nemusí se ovšem vždy jednat o stejný směr, kterým letadlo reálně letí (opět záleží na mnoha dalších proměnných – například na větru), ten se nazývá „kurz“ [17].

Je odesílán nejčastěji ve zprávě „*Surface position*“, spolu s informacemi o pohybu letadla a jeho aktuální poloze (latitude/longitude).

5.12 TAS

TAS – True Air Speed, česky „pravá vzdušná rychlost“, je rychlost letadla vůči okolnímu nerozrušenému proudu vzduchu. Může být stejná jako Ground Speed (GS), ale obvykle se od ní liší, kvůli pohybu vzduchové masy po směru nebo proti směru letu (pokud letíme po směru větru, bude $TAS > GS$, pokud proti, tak $TAS < GS$) [17, 21].

Systém ADS-B získává tuto informaci z aerometrických přístrojů a systémů letounu, jelikož se pro určení TAS používá aktuální tlaková informace a je vysílán se zprávou typu 50 – track and turn report („*type code*“, viz kapitola 5.1.5 této práce) [1, 21].

5.13 Mach

Část zprávy, obsahující informace o rychlosti letu v machově čísle má type code 60 – heading and speed report.

Machovo číslo udává poměr rychlosti pohybu pevného tělesa v určitém prostředí k rychlosti šíření zvukových vln v tomto prostředí [21].

5.14 FOM/Source

Pole FOM/Source ve zprávě udává zdroj informace, která je obsažená ve zbytku zprávy. Obsahuje například číselný kód, který nám říká, že pozice letadla byla získána z palubního GNSS systému [1].

5.15 Meteorologická data - wind speed, wind direction, temperature, pressure, turbulence, humidity

Meteorologická data z palubních systému jsou velice důležitá pro řízení letového provozu a pro bezpečnost provozu jako takovou. Tato data jsou poskytována z palubních počítačů, které je dostávají z jednotlivých subsystémů, senzorů a čidel, případně ze systému FIS-B, který využívá pozemních stanic pro přenos a zaznamenávání meteorologických informací pro konkrétní lety [18].

Pokud jsou všechna meteorologická data dostupná, máme z ADS-B dostupné následující informace:

- Wind speed – rychlost větru,
- Wind direction – naměřený směr větru,
- Temperature – teplota,
- Pressure – aerodynamický tlak,
- Turbulence – informace o turbulencích,
- Humidity – vzdušná vlhkost.

5.16 Použití dat v praktické části práce

V kapitole 5 práce je popsáno, jaká data byla pro praktickou část této práce k dispozici, je jednoduše popisován obsah jednotlivých druhů dat, v jakém typu zprávy jsou vysílány a jak palubní ADS-B tato data získává. Nyní je třeba zběžně popsat, jak je pracováno s daty v praktické části diplomové práce – při tvorbě webové aplikace.

Nejdříve je potřeba si definovat, jaké nástroje jsou v praktické části práce využívány pro databázi samotnou a pro uživatelské rozhraní.

Celý systém se skládá z několika vrstev:

- Datová vrstva,
- Aplikační vrstva,
- Prezentační vrstva.

Posléze je nutno si definovat:

- co jsou to proměnné,
- typy proměnných,
- jak vypadá relační databáze,
- jak přes aplikační vrstvu komunikuje web s databází.

Podrobněji je rozdělení a úkoly jednotlivých vrstev popisováno v kapitole 6.

6 DATABÁZOVÉ SYSTÉMY A JEJICH VYUŽITÍ VE ZPRACOVÁNÍ ADS-B DAT

Databáze je systémem souborů (informací, dat, balíčku...), které jsou navzájem propojeny v jenom systému. Představují technologický postup počítačových systémů a na rozdíl od svých předchůdců – papírových kartoték a psaných databází technických údajů – i značně flexibilní způsob práce s mnoha typy a objemy dat. Jejich využití je značné, téměř každá webová stránka, počítačový systém a výpočetní komponenta mají vlastní, nebo jsou připojeny k nějakému druhu databáze [33, 34].

Příkladů je mnoho:

- webový e-shop s databází uživatelů a jejich objednávek,
- softwarové řešení pro řízení dat, uložených na počítačovém harddisku,
- router, vysílající WiFi, držící v databázi údaje o připojených uživateli, jejich komunikaci a přidělených IP adresách.

Je tedy jasné, že jsme obklopeni různými databázemi, potřebnými pro naši každodenní činnost a ani si to nemusíme uvědomovat. Samotné masové rozšíření databázových řešení napovídá o jejich propracovanosti a nenahraditelnosti v běžném životě [33].

V praktické části práce se řeší možnosti využití databázových systémů pro analýzu vzniku kondenzačních stop a uskladnění dat, která ČVUT získává ze svých ADS-B přijímačů. Samotný název a obsah praktické části diplomové práce může být však mírně zavádějící a matoucí, databáze jako taková je pouhá tabulka, ve které jsou určité sloupce, ve kterých jsou uloženy naměřené nebo získané hodnoty [34].

Jak by tedy mohlo dojít k nějaké analýze vzniku kondenzačních čar pomocí pouhé tabulky? Nedochozí. Samotná tabulka nemůže vyprodukovat a exportovat analýzy dat a nějaké závěry, jako například proč a kdy ke vzniku kondenzační čar dochází. Abychom kýženého výsledku dosáhli, musíme přidat k databázi i další, výše zmíněné, vrstvy – aplikační a prezentační. Ve výsledku musíme pro systém, který by měl zpracovávat data z přijímačů ADS-B, použít tři na sebe navázané vrstvy – databázovou, aplikační a prezentační, abychom byli schopni využít funkční systém, který nebude pouhou tabulkou s daty, ale bude schopen analýzy uložených dat a bude mít i uživatelské rozhraní, ve kterém bude možno nová data nahrávat a pracovat s nimi (například exportovat výsledné analýzy dat uživateli tohoto rozhraní) [34, 35, 36, 37].

6.1 Vícevrstvá a třívrstvá architektura

Vícevrstvá architektura je v softwarovém vývoji označení aplikace, jejíž funkčnost není dána pouze jednou vrstvou – celou uživatelskou aplikací na jednom místě, bez externích závislostí a bez komunikace s například servery – ale spojením více vrstev, které obecně běží na různých výpočetních systémech a serverech [36, 37].

Při vývoji databáze bylo použito **řešení třívrstvé**, tedy „třívrstvou architekturu. Jedná se o jednoduchý koncept poskytnutí uživatelského rozhraní uživateli, přičemž datová a aplikační vrstva zůstane vesměs skryta a uživatel do ní nemůže nijak zasahovat, nebo pouze minimálně [36, 37].

Mějme tedy tři samostatné, navzájem propojené vrstvy:

- **Datovou vrstvu – „Data tier“**,
- **Aplikační vrstvu – „Application tier“**,
- **Prezentační vrstva – „Presentation tier“**.

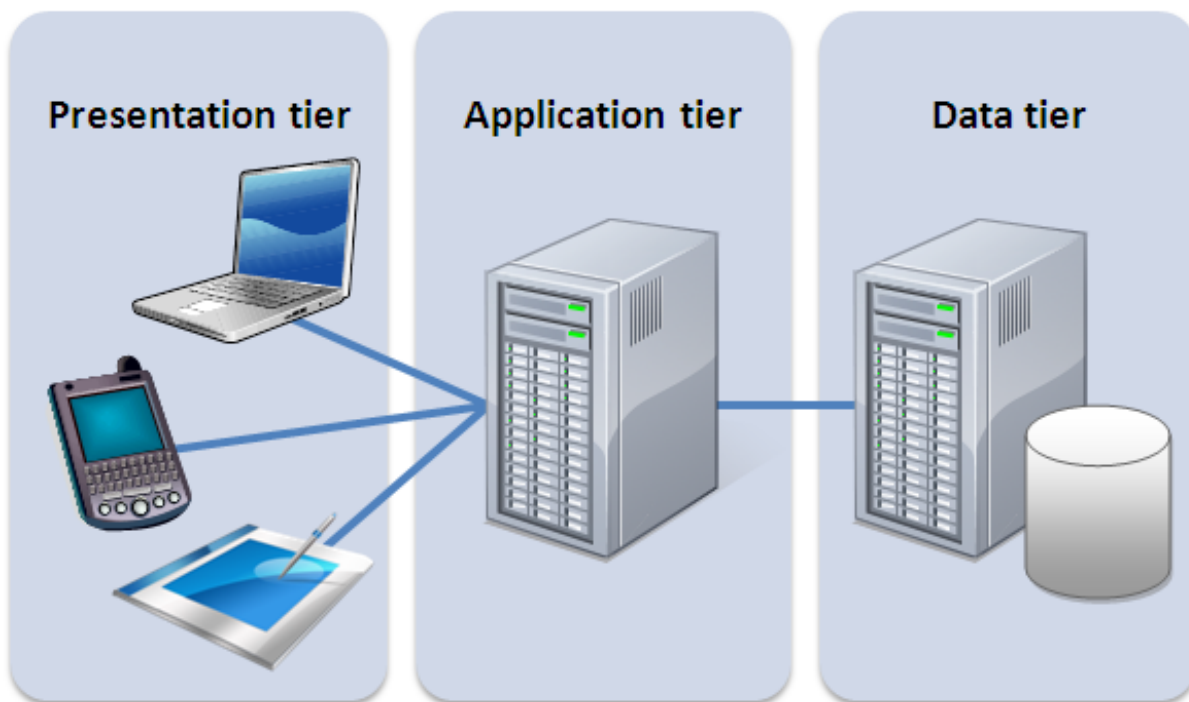
Každá z vrstev plní unikátní a nenahraditelnou roli ve výsledné aplikaci – **datová vrstva** zajišťuje uskladnění a práci s daty pro ostatní vrstvy, **aplikační vrstva** obsahuje funkční business logiku, tedy systém, který propojuje ostatní dvě vrstvy a zpracovává dotazy a požadavky obou těchto vrstev. **Prezentační vrstva** je povětšinou jediná uživateli viditelná část aplikace. Tato vrstva obsahuje uživatelské prostředí a rozhraní pro práci s dalšími vrstvami [36, 37].

Koncept praktické části práce je relativně jednoduchý a pochopitelný – máme databázi, která je součástí datové vrstvy. V databázi máme uložena všechna data, která potřebujeme (data z ADS-B systémů, případně další data, která jsou používána v aplikační vrstvě). S touto vrstvou komunikuje vrstva aplikační, ve které je vytvořen, uchovávan a spouštěn veškerý kód pro obsluhu prezentační a databázové vrstvy. Pokud tedy uživatel klikne na tlačítko pro zobrazení dat, aplikační vrstva dostane od prezentační vrstvy informaci o stisknutí tlačítka, vyžádá si příslušná data od databázové vrstvy a posléze je odešle prezentační vrstvě pro zpracování pro uživatele. Prezentační vrstva slouží jako prostředí pro zobrazování dat uživateli a pro případnou uživatelskou interakci s těmito daty – klikání na tlačítka, filtrování dat, nastavování parametrů a podobně. Nemusí se jednat čistě o webové rozhraní, může jít i o rozhraní počítačové, mobilní a jiné, ovšem v případě vytvořeného systému pro zpracování dat se jedná o webovou stránku [36, 37].

V ideálním případě by neměl mít obyčejný uživatel přístup z webového rozhraní ke vstupu do datové vrstvy, tedy neměl by být schopen jakkoli upravit data, mimo úroveň,

kteřou mu povolují prvky prezentační a aplikační vrstvy, které jsou k tomuto záměru vytvořeny [36, 37].

Na obrázku č. 20 je graficky zobrazena typická třívrstvá architektura a na druhém obrázku (číslo 21) jsou v jednotlivých vrstvách zaznačeny příklady programovacích jazyků, které se pro jednotlivé vrstvy v třívrstvé architektuře často používají:

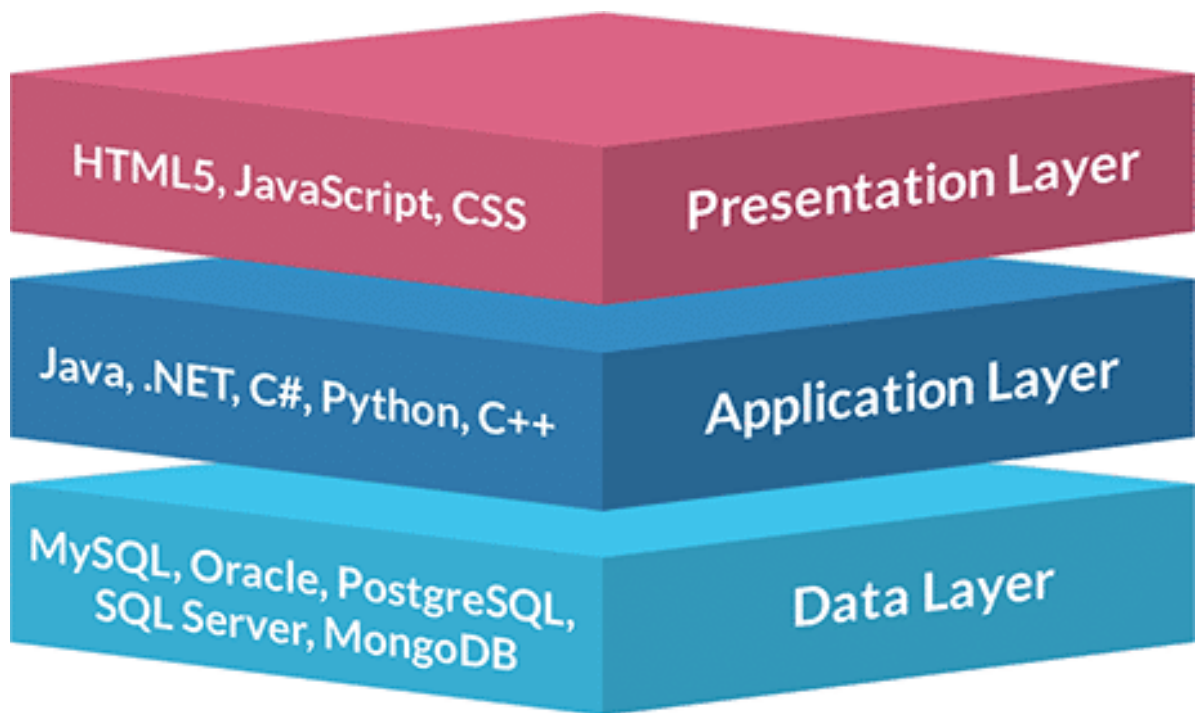


Obr. 20: Grafické vyobrazení propojení jednotlivých vrstev, při použití třívrstvé architektury aplikace [45]

6.2 Datová vrstva v třívrstvé architektuře

Datová vrstva („data tier“) obsahuje databázový systém neboli tabulky uložených dat v určitém pořádku. Jejím účelem je uschování, správa a komunikace s ostatními vrstvami (standardně s vrstvou aplikační, a přes ni i s vrstvou prezentační). Pro svou činnost používá databázový server, na kterém je nasazena („deployed“) a vlastní překladač příkazů z jazyka, ve kterém je databáze napsána a obsluhována [40].

Standardem pro relační databáze (tedy databáze řádků s daty a sloupců s definicemi skupin dat) se víceméně stal jazyk SQL („Structured Query Language“) ve svých mnoha variacích, který funguje na principu strukturovaných dotazů k databázi, která dotazovanému následně například vrací data, dle obsahu obdrženoho dotazu. Nemusí se samozřejmě jednat jen o dotaz na vrácení určitých dat, dotazem SQL se můžeme zeptat na mnoho různých věcí, případně databázi do jisté míry i řídit a nastavovat [40].



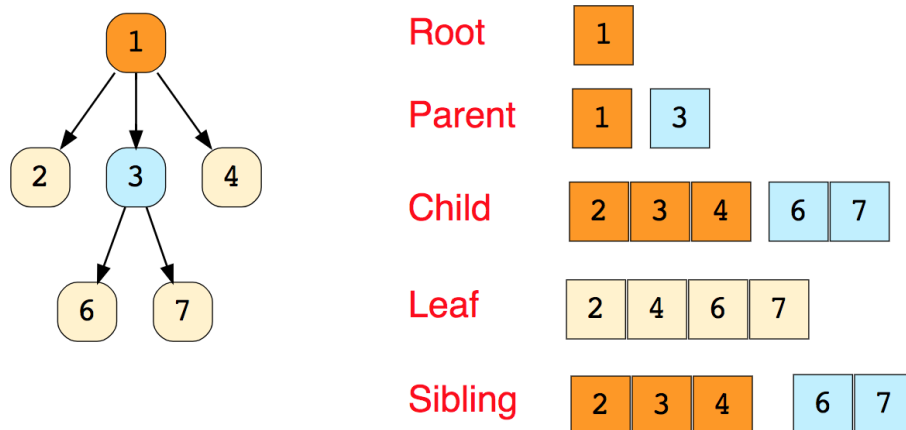
Obr. 21: Grafické vyobrazení jednotlivých vrstev třívrstvé architektury a nejčastěji používaných programovacích jazyků pro jednotlivé vrstvy [37]

6.3 Datová struktura databáze

Datová struktura, je systém uchování dat v určitém uspořádání. Toto uspořádání se mění v závislosti na užití databáze – některá rozložení jsou pro zadaný úkol nevhodná, některá jsou naopak vhodná, záleží tedy na tom, pro co chceme databázi využívat [40, 33].

Častými datovými strukturami, používanými pro různé druhy databází jsou:

- Arrays,
- Stacks,
- Queues,
- Trees (zobrazena na obrázku 22),
- Trie (Prefix trees) a mnoho dalších.



Obr. 22: Struktura dat typu „tree“ s vyobrazením jednotlivých úrovní [33]

6.3.1 Struktura vytvořené relační databáze

V praktické části diplomové práce bylo pro uchovávání dat rozhodnuto využít relační databázi, která je založena na relačním modelu, který popisuje strukturu databáze, způsob manipulace s daty a vlastnosti těchto dat [34, 35].

Mějme tedy databázi, která v sobě drží určité množství tabulek. Tabulka (také označována jako relace) obsahuje v relačním modelu entity a atributy a bývá nejčastěji zobrazena ve dvourozměrném zobrazení. Entitou v takovémto zobrazení pak rozumíme řádek se samotnými daty a atributem sloupec tabulky. Jména atributů jsou v rámci relace různá, dvě stejná jména jsou nepřipustná, data musí taktéž splňovat určité podmínky – zvláště musí být stejného datového typu, jako je zbytek dat a jako je daný sloupec. O datových typech a jejich rozdílech je pojednáváno v kapitole 7.5.2 [34, 35, 40].

Při vytváření takovéto databáze musíme ovšem dodržet určitá pravidla a vyhnout se častým chybám a to zejména:

- Redundance hodnot – opakují-li se určité hodnoty ze sloupce pro více řádků, respektive jsou-li stejné, není nutno tato data udržovat pro každý řádek tabulky,
- Nekonzistence dat z důvodu modifikace v důsledku redundance,
- Anomálie při vkládání záznamů do tabulky,
- Anomálie při odstraňování dat z tabulek – odstranění klíče zaviní zmizení dat v jiné tabulce, jelikož jsou provázané.

Tab. 1: Relace, obsahující některé druhy chyb

Student	Skupina	Předmět	Vyučující	Čas výuky
Novák	C25	3ST	T. K.	15:00
Dvořák	C25	3ST	T. K.	15:00
Petera	C25	3ST		18:00
PetrNovák	C25	3ST	J. N.	17:00
Novotný	C25	3ST	J. N.	17:00
Vrána	C25	3ST	J. N.	17:00

Tabulka 1 výše obsahuje ukázková data – žáky určité studijní skupiny, kteří mají výuku stejného předmětu s různými vyučujícími v různé časy – tato data však obsahují některé výše probírané chyby, které vznikly při práci s daty databáze. Jedná se o **redundantní data** a dva druhy **anomálií dat** – způsobené vkládáním a odstraňováním dat z tabulky [34, 35, 40].

Redundantní data můžeme vidět ve sloupci „**Skupina**“ a „**Předmět**“, kdy jsou data v těchto sloupcích stejná pro všechny studenty a jejich opakování pro každého studenta je tedy bezpředmětné. **Anomálie dat** pozorujeme na dvou místech – anomálii, způsobenou vkládáním dat do tabulky a nekonzistenci z důvodu odstraňováním dat z tabulky. První druh anomálie se nachází ve sloupci „**Student**“ u záznamu „**PetrNovák**“ – zde je evidentní, že nastala chyba při vkládání dat do relační databáze, kdy jméno a příjmení bylo třídícím a zapisujícím skriptem spojeno v jeden záznam. Druhý druh anomálie se nachází ve sloupci „**Vyučující**“, ve kterém jsou uvedeny zkratky jmen vyučujících pro dané žáky. Zde pravděpodobně došlo k smazání záznamů o učitelích v jiné relaci a v důsledku tohoto smazání byl záznam odstraněn i z této příkladové relace [34, 35, 40].

Všechny tyto chyby jsou ošetřitelné a kontrolovatelné pomocí unit testů napsaného kódu, kdy nám test otestuje například vkládání dat do databáze, ale může otestovat i krajní případy, se kterými by se náš software pro vkládání dat mohl setkat – různé speciální znaky, kontrolu podmínek při zadávání dat a další [35].

6.4 Aplikační vrstva v třívrstvé architektuře

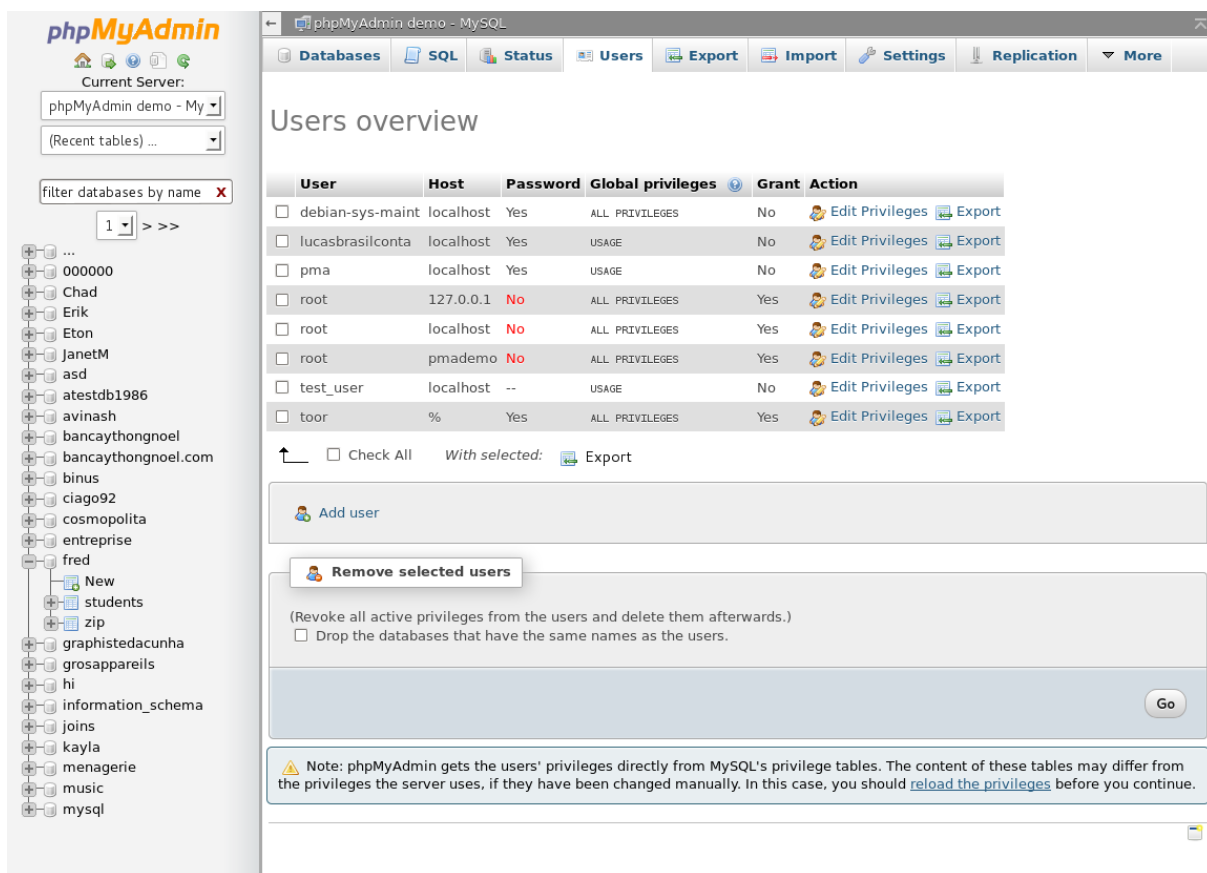
Aplikační vrstva plní v třívrstvé architektuře softwaru funkční roli. To znamená, že obsahuje business logiku dané aplikace, tedy v podstatě definici chování, funkce a funkčnost celé aplikace. Bývá umístěna mezi datovou a prezenční vrstvou tak, že v podstatě zprostředkovává

provádění příkazů a požadavků jednotlivých vrstev, které zpracovává, vyhodnocuje a vykonává [36, 37].

Zjednodušeně – aplikační vrstva obsahuje určitou sadu funkcí, které se v ní vykonávají, přičemž jsou uloženy spolu se zbytkem ovládacích prvků této vrstvy na serveru. I přes to, že na obrázcích 20 a 21, které znázorňují spojení jednotlivých vrstev jako samostatné zařízení, tak často bývají jak datová, aplikační, tak i prezenční vrstva nasazeny pouze jednom fyzickém zařízení (serveru) a rozdělení je provedeno pouze softwarově [36, 37].

Rozhodneme-li se začít programovat aplikační vrstvu, musíme si zvolit vhodný programovací jazyk, ve kterém ji budeme psát. Na výběr máme z desítek jazyků – pro vytvoření aplikační části webové aplikace pro tuto diplomovou práci byl zvolen jazyk PHP (jeho specifika a funkčnost naleznete v kapitole 7.5), pro datovou část aplikace byla použita komponenta MariaDB, používající jazyka SQL a prezentační byla vytvořena v jazycích HTML, CSS s prvky Javascriptu [36, 37].

Aplikační vrstva se často označuje jako **back-end**, zkráceně **BE**. Příklad softwaru, zajišťujícího komunikaci mezi datovou a aplikační vrstvou je zobrazen na obrázku 23.



Obr. 23: Uživatelské rozhraní populárního nástroje phpMyAdmin, které slouží pro správu aplikační a datové vrstvy. Na obrázku zobrazena tabulka dat uživatelů této administrace [46]

6.5 Prezentační vrstva v třívrstvé architektuře

Prezentační vrstva je uživateli dostupné prostředí v třívrstvé aplikační architektuře. Tato vrstva je také často označována jako **front-end** (zkráceně **FE**) a zajišťuje spojení mezi uživatelskými akcemi a aplikační vrstvou [36, 37].

Udělat kvalitně uživatelské prostředí je ovšem velice náročný úkol, na kterém často pro firmu pracují desítky lidí, protože mimo samotné naprogramování elementů, které se budou uživateli zobrazovat, je nutno například udělat UX (uživatelské rozhraní), grafický vzhled stránky, marketing a analýzu trhu, aby firma dosáhla kýženého výsledku – webových stránek, které jim budou generovat zisk. Pokud nejsou webové stránky zamýšlené pro generování zisku, mohou být použity jako místo pro propagaci firmy či události, případně jako nástroj pro zákazníka pro správu jeho objednávek, či jinou administrativu. Příkladem webových stránek, které nejsou zaměřené na generování zisku, ale na poskytování služeb uživateli je například informační systém pro studenty KOS ČVUT (<https://www.kos.cvut.cz>), který není zaměřen na prodej (ani na „akvizici“ nových studentů) a není tedy potřeba jej udržovat ve skvělém designovém stavu, nicméně musí být perfektně funkční [38].

Pro prezenční vrstvu, která je plně napojena na (pro práci vytvořené) back-end řešení a následně i na databázi, byla zvolena kombinace několika programovacích jazyků a frameworků, jmenovitě:

- HTML,
- CSS,
- Javascript,
- PHP,
- Systém pro správu obsahu – Wordpress.

6.5.1 Jazyk HTML

Jazyk HTML (zkratka anglického názvu „*Hypertext Markup Language*“) je značkovací jazyk, který používáme pro tvorbu webových stránek. Jedná se o nejrozšířenější jazyk pro tvorbu webových stránek, v určité podobě se dá najít na většině internetových stránek. Jeho hlavní úlohou je tvorba „kostry“ webové stránky, stylizace textu, tvorba bloků, oddílů, seznamů, tabulek a dalších elementů, přičemž k jejich tvorbě využívá přesně definované tagy a jejich atributy, které si potom prohlížeč překládá a tvoří z těchto značek výslednou podobu stránky [38, 39].

Většina značek je párová, což znamená, že začátek i konec platnosti je ohraničen danou značkou. Koncová značka je doplněna lomítkem. Jako příklad nám poslouží značka `<p>`, která signalizuje začátek a konec odstavce (nejčastěji odstavce textu):

`< p > Příklad textu do odstavce </p >`

V levé části příkladu vidíme počáteční značku `<p>`, která značí prohlížeči počátek odstavce, uprostřed se nachází text, který chceme v odstavci mít a na konci je značka `</p>`, která prohlížeči naopak značí konec odstavce [38, 39].

HTML obsahuje i značky nepovinně párové a nepárové, což znamená, že nemusí být uzavřeny, případně se takovéto značky neuzavírají vůbec. Příkladem nepárové značky je `<hr>`, což je značka, udávající prohlížeči, že má vykreslit vodorovnou čáru. Jazyk HTML tedy obsahuje několik desítek takovýchto značek, každá má svou unikátní funkci a jejich (zejména grafická) stránka funkčnosti je výrazně doplněna kaskádovými styly CSS [38].

```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta http-equiv="x-ua-compatible" content="ie=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8
9   <title>Database</title>
10
11   <link rel="stylesheet" href="CSS/style.css">
12   <link rel="stylesheet" href="CSS/top_nav_bar.css">
13   <link rel="stylesheet" href="CSS/search.css">
14 </head>
15
16 <body>
17
18 <div id="main">
19
20   <html>
21   <body>
22     <div class="top_nav_bar">
23       <a href="/index.php">Hlavní záložka</a>
24       <a href="/search.php">Vyhledávání</a>
25       <a href="/zal2.php">Záložka 2</a>
26       <a href="/import.php">Import dat</a>
27     </div>
28   </body>
29 </html>
30   <div id="body">
31   </div>
32
33 </div>
34
35 </body>
36 </html>
```

Obr. 24: Příklad zdrojového HTML kódu webové stránky.

7 JAZYK SQL a PHP

Jak již bylo napsáno výše, jazyk SQL je jazyk používaný pro obsluhu databáze. Hlavním prvkem komunikace je takzvaný příkaz („statement“), který je v jazyce SQL vyslán databázi, jejíž řídicí systém jej zpracuje a pokud je to možné, tak jej vykoná. Tyto příkazy pak můžeme orientačně klasifikovat podle zamýšleného užití a podle způsobu, jakým s databází komunikují a jaké akce vykonávají. Řadíme je do takzvaných dílčích jazyků („sublanguages“), podle jejich zamýšleného využití, nejčastěji jsou to [38]:

- data query language (DQL),
- data definition language (DDL),
- data control language (DCL),
- data manipulation language (DML).

7.1 Data query language (DQL)

DQL je dílčím souborem příkazů jazyka SQL, který je využíván pro práci s daty, uloženými v databázi, přesněji pro jejich získávání (extrakci) z databáze. Hlavním příkazem, používaným pro dotazy („queries“) tohoto typu je SQL příkaz **SELECT** („vyber“) [40].

Typickým dotazem DQL je dotaz typu „*vyber data ze *jméno tabulky*, kde platí *podmínka**“. Máme tedy například tabulku s výrobky (obrázek 25), které prodává náš imaginární e-shop, tato tabulka se jmenuje „*vyrobky*“. Tabulka výrobky obsahuje sloupce „*jmeno_vyrobku*“, „*id_vyrobku*“, „*mnozstvi_na_sklade*“, přičemž sloupec „*id_vyrobku*“ obsahuje data o identifikačním čísle výrobku a sloupec „*mnozstvi_na_sklade*“ obsahuje data o počtu výrobků příslušného id na skladě [40].

My, jako majitel tohoto e-shopu, si chceme z databáze vybrat data o počtu určitých výrobků na skladě, uděláme to tedy následovně:

```
SELECT *  
FROM vyrobky  
WHERE id_vyrobku IN (1,2,3)
```

Tato query vybere z tabulky „*vyrobky*“ data o výrobcích ze sloupce, kde id výrobku ze sloupce „*id_vyrobku*“ je rovno „1,2,3“.

jmeno_vyrobku	id_vyrobku	mnozstvi_na_sklade
Hrnec, 23 cm	1	12
Poklice, 23 cm	2	11
Lžíce, kávová	3	15
Sklenice, 350 ml	4	3

Obr. 25: Tabulka „vyrobky“

Po použití výše uvedené query na tuto tabulku dostaneme následující výsledky (obrázek 26):

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT * FROM `vyrobky` WHERE `id_vyrobku` IN (1,2,3)
```

Show all | Number of rows: 25 | Filter row

+ Options

jmeno_vyrobku	id_vyrobku	mnozstvi_na_sklade
Hrnec, 23 cm	1	12
Poklice, 23 cm	2	11
Lžíce, kávová	3	15

Obr. 26: Hodnoty, vybrané z databáze pomocí query

7.2 Data definition language (DDL)

„Data definition language“ je dalším ze souborů příkazů jazyka SQL, které jsou používány pro práci s objekty databáze, konkrétně pro jejich vytváření, úpravu a mazání [40].

Pro tuto činnost je vybaven těmito SQL příkazy:

- příkaz CREATE,
- příkaz DROP,
- příkaz ALTER,
- příkaz TRUNCATE,

Příkaz CREATE je možno využít pro vytvoření nové databáze, tabulky, indexu a dalších. **Příkaz DROP** odstraňuje prvek z relační databáze. Může se jednat o databázi, tabulku, index nebo view. V závislosti na implementaci v nástavbě konvenčního SQL může být po provedení (tedy po odstranění prvku z relační databáze) vzata zpět. Tento rollback je možný například ve frameworku PostgreSQL. **Příkaz ALTER** používáme pro modifikaci stávajících databázových objektů, například modifikaci typu proměnné pro daný parametr (převod proměnné z integeru na double). **Příkaz TRUNCATE** je používán pro mazání dat z tabulky, přičemž je o mnoho rychlejší, než příkaz **DELETE** [40].

7.3 Data control language (DCL)

SQL komponenta DCL obsahuje sadu příkazů pro manipulaci s právy k datům databáze. V téměř všech počítačových systémech existuje hierarchie práv – ne všichni uživatelé mohou dělat s daným systémem všechno, například ne všichni uživatelé mohou přistoupit k tabulce, která obsahuje přístupové údaje ostatních uživatelů. K takovýmto datům mohou přistupovat například pouze uživatelé s plnými administrátorskými právy [40].

Dva nejpoužívanější DCL příkazy jsou **GRANT** a **REVOKE**. **Příkaz GRANT** přidělí uživateli práva pro například read-write pro určitý databázový objekt (databázi, tabulku, sloupec...), aby takovýto uživatel s těmito právy mohl do databázového objektu zapisovat/číst z něj. **Příkaz REVOKE** takto získaná práva zase ruší [40].

7.4 Data manipulation language (DML)

Tento soubor SQL příkazů slouží pro manipulaci s daty v SQL databázi. Manipulací je myšleno:

- přidávání/vkládání dat do databázových objektů,
- mazání dat z databázových objektů,
- modifikace dat, uložených v databázových objektech.

SQL příkazy této databázové komponenty jsou **SELECT**, **INSERT**, **UPDATE**, **DELETE**. **Příkaz SELECT** má stejnou funkčnost, jako u DQL, tedy vybírání dat z databázového objektu pro následnou práci s nimi. **Příkaz INSERT** slouží pro vkládání dat do databázového objektu (databáze, tabulky, sloupce...), přičemž může jít o jeden nebo více záznamů najednou. **Příkaz UPDATE** mění zvolená data a nahrazuje/obměňuje je daty, poskytnutými obsluhou databáze (uživatelé). **Příkaz DELETE** maže určitá data, podle poskytnuté podmínky (například – smaž data z tabulky „produkty“, kde je množství produktu na skladě rovno 0). Data nemohou být

obnovena – příkaz DELETE je tedy nutno používat obezřetně, podobně jako příkaz DROP [40].

7.5 Jazyk PHP

Jazyk PHP (celým názvem „Personal Home Page“ nebo také „Hypertext Preprocessor“) je jazyk, používaný pro tvorbu dynamických webových stránek a webových i desktop aplikací. PHP syntaxe jsou psány přímo do kódu webové stránky (volání funkcí, práce s proměnnými atd.) ale nejsou pro běžného uživatele, po zobrazení kódu webové stránky, viditelné. Tyto příkazy jsou prováděny na straně serveru a k uživateli je tak přenášen jen samotný výsledek těchto akcí (například jiný design webové stránky pro uživatele z určitého státu a podobně) [42, 49].

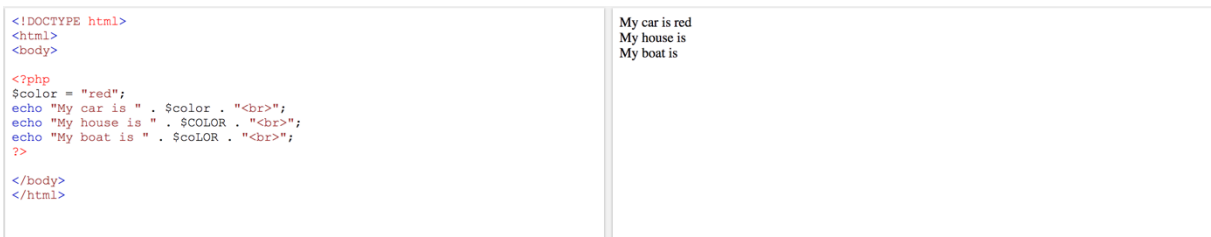
PHP má i velmi dobře řešené propojení různých databázových systémů a webových protokolů. Z databázových systémů podporuje prakticky všechny nejoblíbenější a nejpoužívanější (MySQL, PostgreSQL...) a totéž platí u internetových protokolů, jako jsou HTTP, SMTP, POP3, FTP a další, které jsou používány pro různé účely – od přenosu dat po příjem a odesílání elektronické pošty. Taktéž se řadí mezi nejpoužívanější scriptovací jazyky, přičemž jeho podíl na tomto trhu byl v roce 2017 přes 82 % (následován jazykem ASP.NET), což mimo jiné znamená, že různých vlastních modifikací a doplňků je více, chyby v kódu jsou jednodušeji opravitelné a že existuje velká komunita lidí, která se o údržbu a vývoj tohoto jazyka stará [42, 49].

7.5.1 Technická specifika jazyka PHP

Jedná se o jazyk dynamicky typovaný, což znamená, že datové typy proměnných jsou určovány hodnotou dané proměnné, nikoli jejím předurčeným typem. Začátek a konec scriptu PHP provádíme v kódu webové stránky, jinak psané v HTML, pomocí značek „<?php“ a „?>“ následovně:

```
<?php echo "Text, který chceme vypsát."; ?>
```

Na obrázku 27 jsou zobrazeny jednoduché syntaxe jazyka PHP (na obrázku vlevo), které jsou použity na HTML stránce a jejich překlad na stránku po spuštění (na obrázku vpravo). Příkaz „echo“ slouží pro vypsání určitého textu do uživatelského prostředí a značka „\$“ značí počátek definice proměnné, následována jménem proměnné, které přiřazujeme (za rovnítkem) hodnotou [40, 49].



Obr. 27: Příklad příkazů jazyka PHP na HTML stránce (vlevo) a jejich překlad uživateli (vpravo) [47]

Složitější (či opakující se) skripty není výhodné uzavírat do kódu webové stránky, ale je vhodnější použít samostatné soubory - *.php* skripty, které obsahují celý takovýto skript, včetně deklarací proměnných a definic opakujících se funkcí. Toto řešení nám poskytne velkou výhodu možného znovu-použití takového kódu bez nutnosti jej pokaždé vkládat do každého souboru webové stránky, na kterém chceme takovýto skript použít. V kódu PHP můžeme používat většinu příkazů pro jazyk HTML, například příkaz „
“ (na obrázku 27 výše), který slouží mezerování textu – v podstatě funguje jako klávesa „enter“ na počítačové klávesnici [42, 49].

7.5.2 Proměnné a jejich datové typy

Proměnná je vyhrazený kus výpočetní paměti, který je povětšinou jasně pojmenován a může obsahovat určitou hodnotu nebo také být prázdný. Rozlišujeme určité typy proměnných, povětšinou podle obsahu, který se v nich nachází, případně podle obsahu, který do proměnné tohoto datového typu můžeme uložit [42, 49].

Příklady datových typů proměnných jsou:

- String – datový typ proměnné pro ukládání textu,
- Integer – datový typ proměnné pro ukládání celých čísel,
- Float – datový typ proměnné pro ukládání celých i desetinných čísel,
- Boolean – datový typ proměnné pro ukládání pravdivostních hodnot true/false (1/0).

Mimo tyto čtyři příklady existuje velké množství dalších typů proměnných a také názvy těchto datových typů se mohou v různých programovacích jazycích mírně lišit (například v programovacím jazyce PHP je typ textová proměnná nazýván „String“ a v jazyce SQL je tento samý typ nazýván „Varchar“) [42, 49].

8 MANUÁL K OBSLUZE WEBOVÉ APLIKACE

Webová aplikace se tedy skládá z celkem tří vrstev – prezenční, aplikační a datové. Uživatele ovšem zajímá povětšinou vrstva prezenční, která je mu do prohlížeče prezentována k použití. Takto prezentovaná aplikace musí být přehledná, intuitivní a snadno použitelná. Vytvořená webová aplikace má tedy většinu své funkčnosti přesunutu do vrstvy aplikační, aby bylo uživateli co nejvíce usnadněno používání samotné webové stránky a nemusel řešit příliš mnoho různých nastavení.

Samotná aplikace se z pohledu uživatele skládá ze tří na sobě nezávislých obrazovek:

- Filter results by date,
- Export ADS-B data,
- Upload tables to database.

8.1 Přístup k aplikaci

Aplikace je dočasně spuštěna na webové stránce <https://3219046688.adsb-projekt-vv.cz/>.

Po předání zdrojových kódů aplikace ČVUT bude aplikace nasazena pouze interně na serverech FD ČVUT a to pro pracovníky a studenty ústavu, kteří s ní budou pracovat. Dočasně je však umístěna na výše uvedené adrese, kde je možno si tuto aplikaci otestovat a zhodnotit, přístup není ošetřen žádným heslem a pro zajištění bezpečné komunikace je používáno certifikátu.

8.2 Filter results by date

„Filter results by date“ je úvodní a hlavní obrazovka webové aplikace. Tvoří ji dva formuláře, tlačítko pro odeslání časových dat, nastavených ve formuláři a záhlaví zobrazovaných výsledků. Záhlaví zobrazovaných výsledků obsahuje všechny důležité informace, které máme z tabulky pro dané letadlo dostupné – ICAO adresu letadla, vstupní a výstupní čas, vstupní a výstupní zeměpisnou šířku a délku, první a poslední heading.

Dva formuláře slouží pro nastavení časových dat from - to (od - do) podle kterých se potom zobrazí výsledky, uložené v databázi. Na obrázku 28 je tato stránka zobrazena před vyhledáváním záznamů.

Pomocí „from“ a „to“ formulářů nastaví uživatel rozsah dat, pro které chce v databázi vyhledávat. Po správném nastavení (pokud některé z dat chybí nebo je nastaveno neplatně, je uživatel varován) a kliknutí na tlačítko pro odesílání („Submit“) jsou data předána PHP

scriptu pro zpracování a ten se dotazuje databáze a (z databáze) přijatá data zpracovává do formy tabulek, které jsou uživateli následně zobrazeny.

Filter results by date Export ADS-B data Upload files to database

Filter results by date

Date selection:

From:

To:

SUBMIT

ICAO address	Initial time	Final time	Initial lat	Final lat	Initial lon	Final lon	Initial hdg	Final hdg
--------------	--------------	------------	-------------	-----------	-------------	-----------	-------------	-----------

Obr. 28: Obrazovka „Filter results by date“ bez výsledků vyhledávání

Na obrázku 29 jsou zobrazeny výsledky vyhledávání pro datum „6. září 2018“. Vidíme tedy, že v databázi je pro datum „6.září 2018“ uložena pouze dvojice tabulek a to pro let letounu s ICAO adresou „3C0AC9“ a letounu „4BAB26“. Po získání těchto výsledků má uživatel možnost v levém formuláři pod tabulkou vybrat, co se stalo s kondenzační stopou (předvolené možnosti jsou: „Rozplynula se“, „Byla odváta“, „Splynula s oblačností“ a „Nevzniknula“), v prostředním jak dlouho byla stopa na obloze viditelná (ve formátu hh:mm:ss; defaultně je tento údaj vyplněn 00:00:00 a v případě zvolení možnosti „Nevzniknula“ je i tak uložen) a v pravém jaké letadlo pozorovanou kondenzační stopu vytvořilo (je zde opět možnost výběru z předvolených letadel, např. A380). Uživatel má tedy možnost několika předvolených možností pro levý a pravý formulář, případně může vložit vlastní text. Pro odeslání těchto dat, kliknutím na tlačítko „Submit“ (napravo od formulářů) jsou tato data uložena pro danou tabulku do databáze a jsou pro další hledání v těchto formulářích viditelná – u „3C0AC9“ (na obrázku 29) byla data již vyplněna a odeslána, u „4BAB26“ vyplněna a odeslána ještě nejsou. Oprava již odeslaných dat je vždy možná a to tak, že uživatel změní hodnotu v jednom z formulářů (levý, prostřední, pravý) a takto opravené hodnoty odešle. Původní hodnoty jsou smazány a nahrazeny hodnotami novými. Některé nově nahrané tabulky taky mohou mít předvyplněný letoun (pravý formulář) a to v případě, že je dostupná informace o typu letounu, stejné ICAO adresy z již uložených dat pro jinou tabulku.

ICAO address	Initial time	Final time	Initial lat	Final lat	Initial lon	Final lon	Initial hdg	Final hdg
3C0AC9	20180906_ 11:23:07	20180906_ 11:24:23	50.8624	50.7725	13.9425	14.1392	122	125

Rozplynula_se 02:00:00 A380 **SUBMIT**

4BAB26	20180906_ 11:31:59	20180906_ 11:33:12	50.8547	50.7719	13.9329	14.1411	111	125
--------	-----------------------	-----------------------	---------	---------	---------	---------	-----	-----

00:00:00 **SUBMIT**

Obr. 29: Výsledky vyhledávání pro datum „6.září 2018“

8.3 Export ADS-B data

„Export ADS-B data“ je stránka, určená pro export dat do datových tabulek (ve formátu .csv) dle uživatelem zvolených kritérií. Skládá se ze tří částí:

- Formulář pro výběr intervalu dat,
- Menu pro výběr formátu exportu,
- Tlačítka pro odeslání.

Stejně jako u stránky pro filtrování dat z databáze (kapitola 8.2) je uživatelem na formuláři pro výběr dat nastaveno rozmezí dat pro export tabulek. Následně si v menu vybere, jaký způsob exportu mu vyhovuje a kliknutím na tlačítko „Submit“ odešle. Možnosti exportu jsou tři:

- Links to merged contrail data,
- Links to whole tables in interval,
- Link to contrail information data in interval.

Možnost první (a výchozí) je export hlavních dat z tabulky spolu s daty o kondenzačních stopách (tedy s daty, které uživatel vyplňoval a odesílal na stránce filtrování), spojených v jednu tabulku. Po zvolení této možnosti jsou na serveru vygenerovány dvě tabulky, které jsou uživateli nabídnuty ke stažení. První z těchto tabulek obsahuje veškeré hlavní informace pro všechny tabulky v zadaném datovém intervalu a druhá obsahuje hlavní

informace z tabulek v zadaném intervalu, které mají vyplněné informace o tom, co se stalo s kondenzační stopou.

Možnost druhá („Links to whole tables in interval“) vyexportuje seznam všech tabulek v zadaném intervalu a poskytne odkazy k jednotlivému nebo hromadnému stažení (v jedné tabulce nebo tabulky v „.zip“ archivu).

Možnost třetí („Link to contrail information data in interval“) poskytne uživateli možnost stáhnout pouze data o kondenzačních stopách pro tabulky v zadaném intervalu.

Export selection:

From:

31.12.2015

To:

31.12.2019

- Links to merged contrail data
- Links to whole tables in interval
- Link to contrail information data in interval

SUBMIT

IMPORTANT TABLES INFO (EMPTY CONTRAIL INFO ENABLED):

[merged_tables_with_possibly_empty_contrail_info.csv](#)

IMPORTANT TABLES INFO (EMPTY CONTRAIL INFO DISABLED):

[merged_tables_with_non_empty_contrail_info.csv](#)

Obr. 30: Export .csv tabulek pro zadaný časový interval a pro možnost „Links to merged contrail data“

Na obrázku 30 je zobrazen export tabulek pro možnost první („Links to merged contrail data“) z období 12/2015 do období 12/2019 spolu s odkazy na stažení těchto tabulek. Na obrázku 31 je potom zobrazena část tabulky s hlavními daty o letech v zadaném období spolu s informacemi o kondenzačních stopách.

merged_with_non_empty_con_info1585561234

Table_name	Initial_time	Final_time	Initial_latitude	Final_latitude	Initial_longitude	Final_longitude	Initial_heading	Final_heading	Initial_mach	Final_mach	Initial_TAS	Final_TAS	Contrail_value	Contrail_lasting_time	Aircraft_model	
20180901_0442_3C6DC7	20180901_04:42:09	20180901_04:43:19	50.7684	50.6324	14.4281	14.4785	167	166	776	772	448	448	Byla_odvíta	02:59:58	A380	
20180901_0444_4CACAB	20180901_04:44:28	20180901_04:45:45	50.7727	50.6267	14.3540	14.4169		166	788	796	454	458	Byla_odvíta	00:01:01	A380	
20180901_0446_4891B2	20180901_04:46:26	20180901_04:47:04	50.7797	50.7360	14.4028	14.3014		300	301	768	764	442	440	Nevzniknuta	00:00	B787
20180901_0447_3C5CB2	20180901_04:47:52	20180901_04:48:26	50.9337	50.9182	14.0207	14.1045		105	105	564	572	358	362	Nevzniknuta	00:00	B787
20180901_0447_471F4C	20180901_04:47:25	20180901_04:49:14	50.7643	50.6736	14.5398	14.2523		301	301	772	772	444	444	Byla_odvíta	08:44:59	B777
20180903_1531_123456	20180903_15:31:32	20180903_15:32:14	50.8423	50.7773	13.9191	14.0107		131	139	804	796	470	464	Nevzniknuta	00:00:00	
20180906_1123_3C0AC9	20180906_11:23:07	20180906_11:24:23	50.8624	50.7725	13.9425	14.1392		122	125	696	716	426	436	Rozplynula_se	02:00	A380
20181001_0629_44081E	20181001_06:29:05	20181001_06:30:19	50.8553	50.7712	13.9321	14.0906		118	145	688	704	408	418	Nevzniknuta	00:00	C172
20181001_0633_45CEEF	20181001_06:33:50	20181001_06:34:38	50.7676	50.6798	14.4627	14.5136		162	162	800	800	450	450	Nevzniknuta	00:00:00	

Obr. 31: Výsledná vyexportovaná tabulka, obsahující hlavní informace z tabulek ze zvoleného časového intervalu

8.4 Upload files to database

Stránka „Upload files to database“ umožňuje uživateli nahrání nových tabulek do databáze a to dvěma možnými způsoby. Prvním způsobem je nahrání tabulek (ve formátu „.csv“) jednotlivě. Druhým je potom možnost nahrát tabulky v „.zip“ archívu.

Toto nahrávání má ale pravidla:

1. nahrávaná tabulka musí být v standartním, očekávaném formátu (viz kapitola 5). Pokud by byla nahrána tabulka v jiném než očekávaném formátu, do databáze se neuloží a je zahozena
2. při nahrávání „.zip“ archívu musí být všechny tabulky ve formátu „.csv“ přímo v hlavním adresáři archívu, nikoli v podsložkách. Pokud by byly v podsložkách, není k nim přihlíženo a nejsou do databáze nahrány
3. V případě opakovaného nahrávání stejné tabulky, je uložena pouze jedna kopie a zůstávají pro ni uloženy informace o kondenzační stopě i po opětovném nahrání tabulky.

Upload files to database

CHOOSE YOUR CSV FILE

Soubor nevybrán

CHOOSE YOUR ZIP FILE

Soubor nevybrán

Obr. 32: Stránka „Upload files to database“

Na obrázku 32 je zobrazena stránka pro nahrání nové tabulky/tabulek do databáze a na obrázku 33 je vyfocena vzorová tabulka, která může být nahrána do databáze. Pokud by například počet sloupců neodpovídal, tabulka se do databáze nenahráje.

20180903_1531_123456

Unix time	Message	Time	DF	Type_code/BDS_number	Latitude	Longitude	Altitude	Ground speed	Baro_diff	Heading	TAS	Mach	FOM_SOURCE	Wind_speed	Wind_direction	Temperature	Pressure	Turbulence	Humidity
1535986692269	BD3C097B58AB155478C8A902A996	20180903_15:31:32	17		11	50.8423	13.9191	33025											
1535986692335	BD3C097B58AB155478C8A902A996	20180903_15:31:32	17		11	50.8423	13.9191	33025											
1535986692328	BD3C097B994148A748082A7D3112	20180903_15:31:32	17		19			452.6566		1025									
1535986692335	BD3C097B994148A748082A7D3112	20180903_15:31:32	17		19			452.6566		1025									
15359866922950	A0001531AE9A3D327FC7FE4968F7	20180903_15:31:32	20		60						131	0.8040							
15359866922951	A800069283FF338A03CEB7C9F38	20180903_15:31:32	21		50						470								
15359866922963	A0001531AE9A3D327FC7FE4968F7	20180903_15:31:32	20		60						131	0.8040							
15359866922963	A800069283FF338A03CEB7C9F38	20180903_15:31:32	21		50						470								
15359866932256	BD3C097B58AB11E520DC700144B9	20180903_15:31:33	17		11	50.8425	13.9188	33025											
15359866932268	BD3C097B58AB11E520DC700144B9	20180903_15:31:33	17		11	50.8425	13.9188	33025											
1535986693401	A000153183FF338A03CEB6F62BC	20180903_15:31:33	20		50						470								
1535986693401	A0001531EAA3D326017FEB75AAF	20180903_15:31:33	20		60						131	0.8040							
1535986693402	A000153183FF338A03CEB6F62BC	20180903_15:31:33	20		50						470								
1535986693402	A0001531EAA3D326017FEB75AAF	20180903_15:31:33	20		60						131	0.8040							
1535986693487	BD3C097B994148A748082A7D3112	20180903_15:31:33	17		19			452.6566		1025									
1535986693498	BD3C097B994148A748082A7D3112	20180903_15:31:33	17		19			452.6566		1025									
1535986693567	A0001530EAA3D327FC7FE3E0B64	20180903_15:31:33	20		60						131	0.8040							

Obr. 33: Fotka vzorové tabulky, která může být nahrána do databáze bez problémů

9 ZDROJOVÝ KÓD WEBOVÉ APLIKACE

Zdrojový kód je hlavním prvkem celé aplikace a byl vytvořen speciálně pro použití ve webové aplikaci. Jak již bylo popsáno v kapitolách výše, dělíme jej na kód prezenční, aplikační a datové vrstvy. Zdrojový kód prezenční vrstvy je vytvořen v jazycích HTML, Javascript a CSS, aplikační PHP a pro komunikaci s datovou vrstvou se používají dotazy v jazyce SQL, avšak tato komunikace probíhá na vrstvě aplikační.

9.1 Zdrojový kód aplikační vrstvy

Na obrázku 34 je vytisknut zdrojový kód aplikační vrstvy, který je používán pro filtrování tabulek z databáze dle uživatelem zadaných časových údajů. Z těchto tabulek se v tomto kódu následně získávají klíčové informace (tedy například ICAO kód, vstupní a výstupní souřadnice a další) a tyto data jsou následně zobrazeny uživateli v tabulce s výsledky (zobrazeno na obrázku 29).

Na obrázku 35 je zobrazen kód pro export tabulek, dle uživatelem zadaného časového intervalu, přičemž uživatel si volí druh exportu, což je blíže popsáno v kapitole 8.3.

Na obrázcích 36 a 37 je zdrojový kód pro import nových tabulek uživatelem a to buď nahráním jednoho „.csv“ souboru (jedné tabulky), nebo nahráním celého „.zip“ archívu tabulek.

```

1  if(array_key_exists('save_date',$_POST)) {
2      global $from_date, $to_date;
3
4      $from_date = htmlspecialchars($_POST['dateFrom']);
5      $to_date = htmlspecialchars($_POST['dateTo']);
6
7      do_filtering($from_date, $to_date);
8  }
9
10 if(array_key_exists('submit_contrail_info',$_POST)) {
11     global $wordpress,$wpdb;
12
13     $from_date = htmlspecialchars($_POST['dateFrom']);
14     $to_date = htmlspecialchars($_POST['dateTo']);
15     $con_life_value = htmlspecialchars($_POST['contrail_list_life']);
16     $plane_model_value = htmlspecialchars($_POST['plane_model']);
17     $table = htmlspecialchars($_POST['tableName']);
18     $con_value = htmlspecialchars($_POST['contrail_list']);
19
20     $select_value_query = "SELECT value FROM
    • contrail_info_about_tables WHERE table_name='$table' LIMIT 1";
21     $result_value = $wpdb->get_results($select_value_query);
22     $final_result_value = array_column($result_value, "value");
23     $query = $wpdb->query("DELETE FROM contrail_info_about_tables
    • WHERE table_name='$table'");
24     $icao_code = substr($table, -6);
25
26     $wpdb->insert('contrail_info_about_tables', array(
27         'table_name' => $table,
28         'value' => $con_value,
29         'contrail_lasting_time' => $con_life_value,
30         'model' => $plane_model_value,
31         'icao24' => $icao_code,
32     ));
33
34     do_filtering($from_date, $to_date);
35 }
36
37 function do_filtering($from_date, $to_date) {
38     $date_interval_array = save_date_interval($from_date, $to_date);
39     $fetched_tables_array = fetch_wordpress_tables();
40     $cropped_tables = crop_fetched_tables($fetched_tables_array);
41     $filtered_array = select_tables_in_interval($cropped_tables,
    • $date_interval_array);
42     $filtered_tables_array =
    • select_tables_by_filter($fetched_tables_array, $filtered_array);
43     $array_of_icao_codes = get_icao_codes($filtered_tables_array);
44     get_aircraft_initial_and_final_values($filtered_tables_array,
    • $array_of_icao_codes, $from_date, $to_date);

```

```

45 }
46
47 function fetch_wordpress_tables() {
48     global $wordpress,$wpdb;
49
50     $sql = "SHOW TABLES LIKE '20%'";
51
52     $results = $wpdb->get_results($sql);
53
54     foreach($results as $index => $value) {
55         foreach($value as $tableName) {
56             }
57         $array_of_tables[$index] = $tableName;
58     }
59     return $array_of_tables;
60 }
61
62 function crop_fetched_tables($tables_array) {
63     foreach($tables_array as $index => $value) {
64         $array_of_cropped_tables[$index] =
        • intval(substr($tables_array[$index],0,8));
65     }
66     return $array_of_cropped_tables;
67 }
68
69 function save_date_interval($date_from, $date_to) {
70     $from_date = date('Ymd', strtotime($date_from));
71     $to_date = date('Ymd', strtotime($date_to));
72
73     if ($to_date != "19700101" && $from_date != "19700101") {
74         $date_array_int[0] = intval($from_date);
75         $date_array_int[1] = intval($to_date);
76         return $date_array_int;
77     } else {
78         echo "<script type='text/javascript'>alert('One of the dates not
        • selected!');</script>";
79     }
80 }
81
82 function select_tables_in_interval($array_of_tables,
    • $interval_array) {
83     $from_int = $interval_array[0];
84     $to_int = $interval_array[1];
85
86     foreach($array_of_tables as $index => $value) {
87         $tmp_table = $array_of_tables[$index];
88
89         if ($interval_array[0] <= $tmp_table && $tmp_table <=
        • $interval_array[1]) {

```

```

90     $filtered_array[$index] = $tmp_table;
91 }
92 }
93 return $filtered_array;
94 }
95
96 function select_tables_by_filter($array_of_tables, $filtered_array) {
97     $final_array = array();
98
99     foreach($array_of_tables as $i => $value_i) {
100         foreach($filtered_array as $j => $value_j) {
101             if (intval(substr($array_of_tables[$i],0,8)) ==
    • $filtered_array[$j]) {
102                 $final_array[$i] = $array_of_tables[$i];
103             }
104         }
105     }
106     $final_array = array_unique($final_array);
107     return $final_array;
108 }
109
110 function get_aircraft_initial_and_final_values($array_of_tables,
    • $array_of_icao_codes, $date_from, $date_to) {
111     global $wordpress,$wpdb;
112
113     foreach ($array_of_tables as $index => $tables) {
114         $initial_vector = array();
115         $final_vector = array();
116         $initial = "
117             SELECT Time,Latitude,Longitude,Heading
118             FROM `\$array_of_tables[$index]`
119             WHERE Latitude !=0
120             ORDER BY Time ASC
121             LIMIT 1";
122         $final = "
123             SELECT Time,Latitude,Longitude,Heading
124             FROM `\$array_of_tables[$index]`
125             WHERE Latitude != 0
126             ORDER BY Time
127             DESC LIMIT 1";
128
129         $init_heading = "
130             SELECT Heading
131             FROM `\$array_of_tables[$index]`
132             WHERE Heading != 0
133             ORDER BY Time
134             LIMIT 1";
135
136         $final_heading = "

```



```

137     SELECT Heading
138     FROM ` $array_of_tables[$index] `
139     WHERE Heading != 0
140     ORDER BY Time
141     DESC LIMIT 1";
142
143     $results_initial = $wpdb -> get_results($initial);
144     $results_final = $wpdb -> get_results($final);
145     $result_heading_init = $wpdb -> get_results($init_heading);
146     $result_heading_final = $wpdb -> get_results($final_heading);
147
148     $time_init = array_column($results_initial, 'Time');
149     $time_final = array_column($results_final, 'Time');
150     $lat_init = array_column($results_initial, 'Latitude');
151     $lat_final = array_column($results_final, 'Latitude');
152     $lon_init = array_column($results_initial, 'Longitude');
153     $lon_final = array_column($results_final, 'Longitude');
154     $heading_init = array_column($result_heading_init, 'Heading');
155     $heading_final = array_column($result_heading_final, 'Heading');
156
157     echo "
158     <table>
159     <tr>
160         <td><small>". $array_of_icao_codes[$index] . "</small></td>
161         <td><small>". $time_init[0] . "</small></td>
162         <td><small>". $time_final[0] . "</small></td>
163         <td><small>". $lat_init[0] . "</small></td>
164         <td><small>". $lat_final[0] . "</small></td>
165         <td><small>". $lon_init[0] . "</small></td>
166         <td><small>". $lon_final[0] . "</small></td>
167         <td><small>". $heading_init[0] . "</td>
168         <td><small>". $heading_final[0] . "</td>
169     </tr>
170     </table>
171     ";
172
173     choose_what_happened_with_contrail($array_of_icao_codes[$index],
174     • $date_from, $date_to, $array_of_tables[$index]);
175 }
176
177 function get_icao_codes($array_of_tables) {
178     foreach ($array_of_tables as $index => $table) {
179         $array_of_icao_codes[$index] = substr($array_of_tables[$index], -
180         • 6);
181     }
182     return $array_of_icao_codes;
183 }

```

```

184 function choose_what_happened_with_contrail($icao_code, $date_from,
    • $date_to, $table) {
185     global $wordpress,$wpdb;
186
187     $icao_code = substr($table, -6);
188
189     $model_icao24 = "SELECT model FROM contrail_info_about_tables
    • WHERE icao24='$icao_code' LIMIT 1";
190     $select_value_query = "SELECT value FROM
    • contrail_info_about_tables WHERE table_name='$table' LIMIT 1";
191     $select_lifetime_query = "SELECT contrail_lasting_time FROM
    • contrail_info_about_tables WHERE table_name='$table' LIMIT 1";
192     $select_model_query = "SELECT model FROM
    • contrail_info_about_tables WHERE table_name='$table' LIMIT 1";
193
194     $result_value = $wpdb->get_results($select_value_query);
195     $result_lifetime = $wpdb->get_results($select_lifetime_query);
196     $result_model = $wpdb->get_results($select_model_query);
197     $result_icao_model = $wpdb->get_results($model_icao24);
198
199     $final_result_value = array_column($result_value, "value");
200     $final_result_lifetime = array_column($result_lifetime,
    • "contrail_lasting_time");
201     $final_result_model_icao = array_column($result_icao_model,
    • "model");
202
203     if (empty($final_result_model_icao[0])) {
204         $final_result_model = array_column($result_model, "model");
205     } else if (!empty($final_result_model_icao[0])) {
206         $final_result_model[0] = $final_result_model_icao[0];
207     }
208
209     if (empty($final_result_lifetime[0])) {
210         $final_result_lifetime[0] = "00:00:00";
211     }
212
213     echo "
214     <form name='list' method='post'>
215         <input type='hidden' name='dateFrom' value='$date_from' />
216         <input type='hidden' name='dateTo' value='$date_to' />
217         <input type='hidden' name='tableName' value='$table' />
218         <div>
219         <input type='text' list='contrail' name='contrail_list'
    • value=".$final_result_value[0].">
220         <datalist id='contrail'>
221             <option value='Rozplynula_se'>
222             <option value='Byla_odvata'>
223             <option value='Splynuma_s_oblacností'>
224             <option value='Nevzniknula'>
    ---

```

```

225     </datalist>
226
227     <input type='time' step='1' id=appt' list='contrail_life'
    • name='contrail_list_life' value=".$final_result_lifetime[0].">
228     <datalist id='contrail_life'>
229         <option value='1h'>
230         <option value='2h'>
231         <option value='3h'>
232         <option value='4h'>
233     </datalist>
234
235     <input type='text' list='plane_model' name='plane_model'
    • value=".$final_result_model[0].">
236     <datalist id='plane_model'>
237         <option value='B777'>
238         <option value='A380'>
239         <option value='B787'>
240         <option value='C172'>
241     </datalist>
242
243     <input type='submit' name='submit_contrail_info' value='Submit' />
244 </div>
245 </form>
246 <hr>;
247 }
248

```

Obr. 34: Zdrojový kód aplikační vrstvy, používaný pro filtrování tabulek a hlavních údajů z těchto tabulek

```

1  if(array_key_exists('save_date_export',$_POST)){
2      global $from_date, $to_date;
3      global $wordpress,$wpdb;
4
5      $from_date = htmlspecialchars($_POST['dateFromExport']);
6      $to_date = htmlspecialchars($_POST['dateToExport']);
7      $export_choice = htmlspecialchars($_POST['tables_in_range']);
8
9      $array_of_filtred_tables = do_filtering($from_date, $to_date);
10
11     if (sizeof($array_of_filtred_tables) > 0) {
12         if ($export_choice == "contrail_data_in_range") {
13             export_contrail_info_in_interval($array_of_filtred_tables);
14         } else if ($export_choice == "whole_tables_in_range") {
15             export_full_tables_in_interval($array_of_filtred_tables);
16         } else if ($export_choice == "merged_contrail_data") {
17             get_main_contrail_table_info($array_of_filtred_tables);
18         }
19     } else {
20         echo "<script type='text/javascript'>alert('No tables in this
    • interval.');

```

```

46
47 function fetch_wordpress_tables() {
48     global $wordpress,$wpdb;
49
50     $sql = "SHOW TABLES LIKE '20%'";
51     $results = $wpdb->get_results($sql);
52
53     foreach($results as $index => $value) {
54         foreach($value as $tableName) {
55             }
56         $array_of_tables[$index] = $tableName;
57     }
58     return $array_of_tables;
59 }
60
61 function crop_fetched_tables($tables_array) {
62     foreach($tables_array as $index => $value) {
63         $array_of_cropped_tables[$index] =
64         • intval(substr($tables_array[$index],0,8));
65     }
66     return $array_of_cropped_tables;
67 }
68 function select_tables_in_interval($array_of_tables,
69 • $interval_array) {
70     $from_int = $interval_array[0];
71     $to_int = $interval_array[1];
72
73     foreach($array_of_tables as $index => $value) {
74         $tmp_table = $array_of_tables[$index];
75
76         if ($interval_array[0] <= $tmp_table && $tmp_table <=
77         • $interval_array[1]) {
78             $filtered_array[$index] = $tmp_table;
79         }
80     }
81     return $filtered_array;
82 }
83 function select_tables_by_filter($array_of_tables, $filtered_array) {
84     $final_array = array();
85
86     foreach($array_of_tables as $i => $value_i) {
87         foreach($filtered_array as $j => $value_j) {
88             if (intval(substr($array_of_tables[$i],0,8)) ==
89             • $filtered_array[$j]) {
90                 $final_array[$i] = $array_of_tables[$i];
91             }
92         }
93     }
94 }

```

```

91     }
92     $final_array = array_unique($final_array);
93     return $final_array;
94 }
95
96 function export_full_tables_in_interval($array_of_tables) {
97     global $wordpress,$wpdb;
98     clean_buffer_folder();
99     $random_number = strval(rand());
100
101     $keys_array = array(
102         "Unix_time",
103         "Message",
104         "Time",
105         "DF",
106         "Type_code/BDS_number",
107         "Latitude",
108         "Longitude",
109         "Altitude",
110         "Ground_speed",
111         "Baro_diff",
112         "Heading",
113         "TAS",
114         "Mach",
115         "FOM_SOURCE",
116         "Wind_speed",
117         "Wind_direction",
118         "Temperture",
119         "Pressure",
120         "Turbulence",
121         "Humidity",
122     );
123
124     $buffer = fopen("export_data/buffer".$random_number.".csv", "w+");
125     fputcsv($buffer, $keys_array);
126     foreach ($array_of_tables as $index => $tables) {
127         $sql = "SELECT * FROM `{$array_of_tables[$index]}`";
128         $results[$index] = $wpdb->get_results($sql, 'ARRAY_A');
129         $table_file = fopen("export_data/
    • {$array_of_tables[$index]}.csv", "w+");
130         fputcsv($table_file, $keys_array);
131         foreach ($results[$index] as $row) {
132             fputcsv($table_file, $row);
133         }
134         fclose($table_file);
135     }
136     foreach ($results as $result) {
137         foreach ($result as $row) {
138             fputcsv($buffer, $row);
139         }
140     }

```



```

139     }
140     fclose($buffer);
141     }
142
143     create_zip_archive($array_of_tables);
144     print_out_download_links($array_of_tables, $random_number);
145 }
146
147 function export_merged_tables_in_interval($array_of_tables) {
148     global $wordpress,$wpdb;
149
150     clean_buffer_folder();
151     $empty_con = fopen("export_data/
152     • merged_with_possible_empty_con_info.csv", "w+");
153     $not_empty_con = fopen("export_data/
154     • merged_with_non_empty_con_info.csv", "w+");
155
156     foreach ($array_of_tables as $index => $tables) {
157         $sql = "SELECT * FROM `{$array_of_tables[$index]}`";
158         $results[$index] = $wpdb->get_results($sql, 'ARRAY_A');
159         $table_file = fopen("export_data/
160         • {$array_of_tables[$index]}.csv", "w+");
161         foreach ($results[$index] as $row) {
162             fputcsv($table_file, $row);
163         }
164         fclose($table_file);
165         foreach ($results as $result) {
166             foreach ($result as $row) {
167                 fputcsv($buffer, $row);
168             }
169         }
170         fclose($buffer);
171     }
172
173     create_zip_archive($array_of_tables);
174     print_out_download_links($array_of_tables);
175 }
176
177 function get_main_contrail_table_info($array_of_tables) {
178     global $wordpress,$wpdb;
179
180     clean_buffer_folder();
181     $random_number = strval(rand());
182     $empty_con = fopen("export_data/
183     • merged_with_possible_empty_con_info".$random_number.".csv", "w+");
184     $not_empty_con = fopen("export_data/
185     • merged_with_non_empty_con_info".$random_number.".csv", "w+");
186     $keys_array = array(
187         "Table_name",

```

```

183     "Initial_time",
184     "Final_time",
185     "Initial_latitude",
186     "Final_latitude",
187     "Initial_longitude",
188     "Final_longitude",
189     "Initial_heading",
190     "Final_heading",
191     "Initial_mach",
192     "Final_mach",
193     "Initial_TAS",
194     "Final_TAS",
195     "Contrail_value",
196     "Contrail_lasting_time",
197     "Aircraft_model"
198 );
199
200 fputcsv($empty_con, $keys_array);
201 fputcsv($not_empty_con, $keys_array);
202
203 foreach ($array_of_tables as $index => $tables) {
204     $initial_data = "
205         SELECT Time,Latitude,Longitude,Heading
206         FROM `\$array_of_tables[$index]`
207         WHERE Latitude !=0
208         ORDER BY Time ASC
209         LIMIT 1";
210
211     $final_data = "
212         SELECT Time,Latitude,Longitude,Heading
213         FROM `\$array_of_tables[$index]`
214         WHERE Latitude != 0
215         ORDER BY Time
216         DESC LIMIT 1";
217
218     $init_heading = "
219         SELECT Heading
220         FROM `\$array_of_tables[$index]`
221         WHERE Heading != 0
222         ORDER BY Time
223         LIMIT 1";
224
225     $final_heading = "
226         SELECT Heading,Mach
227         FROM `\$array_of_tables[$index]`
228         WHERE Heading != 0
229         ORDER BY Time
230         DESC LIMIT 1";
231

```



```

232 $init_mach = "
233     SELECT Mach
234     FROM ` $array_of_tables[$index] `
235     WHERE Mach != 0
236     ORDER BY Time
237     LIMIT 1";
238
239 $final_mach = "
240     SELECT Mach
241     FROM ` $array_of_tables[$index] `
242     WHERE Mach != 0
243     ORDER BY Time
244     DESC LIMIT 1";
245
246 $init_tas = "
247     SELECT TAS
248     FROM ` $array_of_tables[$index] `
249     WHERE TAS != 0
250     ORDER BY Time
251     LIMIT 1";
252
253 $final_tas = "
254     SELECT TAS
255     FROM ` $array_of_tables[$index] `
256     WHERE TAS != 0
257     ORDER BY Time
258     DESC LIMIT 1";
259
260 $select_icao_info_allow_empty = "
261     SELECT value, contrail_lasting_time, model
262     FROM contrail_info_about_tables
263     WHERE table_name = '$array_of_tables[$index]'
264     LIMIT 1";
265
266 $results_initial = $wpdb -> get_results($initial_data);
267 $results_final = $wpdb -> get_results($final_data);
268 $result_heading_init = $wpdb -> get_results($init_heading);
269 $result_heading_final = $wpdb -> get_results($final_heading);
270 $result_mach_init = $wpdb -> get_results($init_mach);
271 $results_mach_final = $wpdb -> get_results($final_mach);
272 $result_tas_init = $wpdb -> get_results($init_tas);
273 $result_tas_final = $wpdb -> get_results($final_tas);
274 $con_info = $wpdb -> get_results($select_icao_info_allow_empty);
275
276 $time_init = array_column($results_initial, 'Time');
277 $time_final = array_column($results_final, 'Time');
278 $lat_init = array_column($results_initial, 'Latitude');
279 $lat_final = array_column($results_final, 'Latitude');
280 $lon_init = array_column($results_initial, 'Longitude');

```

```

281 $lon_final = array_column($results_final, 'Longitude');
282 $heading_init = array_column($result_heading_init, 'Heading');
283 $heading_final = array_column($result_heading_final, 'Heading');
284 $mach_init = array_column($result_mach_init, 'Mach');
285 $mach_final = array_column($results_mach_final, 'Mach');
286 $tas_init = array_column($result_tas_init, 'TAS');
287 $tas_final = array_column($result_tas_final, 'TAS');
288 $con_value = array_column($con_info, 'value');
289 $con_lasting_time = array_column($con_info,
  • 'contrail_lasting_time');
290 $con_model = array_column($con_info, 'model');
291
292 if (strlen($lat_init[0]) != 7){
293     for ($i = 1; strlen($lat_init[0]) <= 6; $i++) {
294         $lat_init[0] = $lat_init[0]."0";
295     }
296 }
297
298 if (strlen($lat_final[0]) != 7){
299     for ($i = 1; strlen($lat_final[0]) <= 6; $i++) {
300         $lat_final[0] = $lat_final[0]."0";
301     }
302 }
303
304 if (strlen($lon_init[0]) != 7){
305     for ($i = 1; strlen($lon_init[0]) <= 6; $i++) {
306         $lon_init[0] = $lon_init[0]."0";
307     }
308 }
309
310 if (strlen($lon_final[0]) != 7){
311     for ($i = 1; strlen($lon_final[0]) <= 6; $i++) {
312         $lon_final[0] = $lon_final[0]."0";
313     }
314 }
315
316 if (strlen($mach_init[0]) != 5){
317     for ($i = 1; strlen($mach_init[0]) <= 4; $i++) {
318         $mach_init[0] = $mach_init[0]."0";
319     }
320 }
321
322 if (strlen($mach_final[0]) != 5){
323     for ($i = 1; strlen($mach_final[0]) <= 4; $i++) {
324         $mach_final[0] = $mach_final[0]."0";
325     }
326 }
327
328 $info_array = array(
329     "type" => "contrail",
330     "model" => $con_model,
331     "value" => $con_value,
332     "lasting_time" => $con_lasting_time,
333     "tas_init" => $tas_init,
334     "tas_final" => $tas_final,
335     "mach_init" => $mach_init,
336     "mach_final" => $mach_final,
337     "heading_init" => $heading_init,
338     "heading_final" => $heading_final,
339     "lon_init" => $lon_init,
340     "lon_final" => $lon_final,
341     "lat_init" => $lat_init,
342     "lat_final" => $lat_final
343 );

```

```

329         "table_name" => $array_of_tables[$index],
330         "Initial_time" => $time_init[0],
331         "Final_time" => $time_final[0],
332         "Initial_latitude" => $lat_init[0],
333         "Final_latitude" => $lat_final[0],
334         "Initial_longitude" => $lon_init[0],
335         "Final_longitude" => $lon_final[0],
336         "Initial_heading" => $heading_init[0],
337         "Final_heading" => $heading_final[0],
338         "Initial_mach" => $mach_init[0],
339         "Final_mach" => $mach_final[0],
340         "Initial_TAS" => $tas_init[0],
341         "Final_TAS" => $tas_final[0],
342         "Contrail_value" => $con_value[0],
343         "Contrail_lasting_time" => $con_lasting_time[0],
344         "Aircraft_model" => $con_model[0],
345     );
346
347     fputcsv($empty_con, $info_array);
348
349     if (!empty($con_value[0])) {
350         fputcsv($not_empty_con, $info_array);
351     }
352 }
353
354 fclose($empty_con);
355 fclose($not_empty_con);
356
357 echo "<h4>Important tables info (empty contrail info ENABLED): </
    • h4>";
358 echo "<a href='https://www.3219046688.adsb-projekt-vv.cz/
    • export_data/
    • merged_with_possible_empty_con_info'."$random_number.".csv'
    • download>merged_tables_with_possibly_empty_contrail_info.csv </a>";
359 echo "<h4> Important tables info (empty contrail info DISABLED): </
    • h4>";
360 echo "<a href='https://www.3219046688.adsb-projekt-vv.cz/
    • export_data/merged_with_non_empty_con_info'."$random_number.".csv'
    • download>merged_tables_with_non_empty_contrail_info.csv</a>";
361 }
362
363 function print_out_download_links($array_of_tables, $random_number) {
364     echo "<h4> Link to zip file (contains all .csv tables in
    • interval): </h4>";
365     echo "<a href='https://www.3219046688.adsb-projekt-vv.cz/
    • export_data/tables.zip'>tables.zip</a>";
366     echo "<h4> Links to separate tables from interval: </h4>";
367
368     foreach ($array_of_tables as $index => $tables) {
369         if ($time_init[0] < $time_final[0]) {

```

```

369     IT ($index := 0) 1
370     echo "<br>";
371 }
372
373     echo "<a href='https://www.3219046688.adsb-projekt-vv.cz/
    •   export_data/
    •   ".$array_of_tables[$index].".csv'>$array_of_tables[$index]</a>";
374 }
375
376     echo "<h4> Link to buffer file (all tables from interval in one
    •   .csv file): </h4>";
377     echo "<a href='https://www.3219046688.adsb-projekt-vv.cz/
    •   export_data/buffer".$random_number.".csv'>buffer.csv</a>";
378
379 }
380
381 function create_zip_archive($array_of_tables) {
382     $zip_file = fopen("export_data/tables.zip", "w+");
383     $zip = new ZipArchive();
384     fclose($zip_file);
385     $zip_path = "export_data/tables.zip";
386
387     if($zip->open($zip_path, ZIPARCHIVE::CREATE)!==TRUE) {
388         echo "ZIP creation failed!";
389     }
390     foreach ($array_of_tables as $index => $table) {
391         $zip->addFile("export_data/".$array_of_tables[$index].".csv",
    •     $array_of_tables[$index].".csv");
392     }
393     $zip->close();
394 }
395
396 function clean_buffer_folder() {
397     $files = glob('export_data/*');
398
399     foreach($files as $file) {
400         if(is_file($file))
401             unlink($file);
402     }
403 }
404
405 function export_contrail_info_in_interval($array_of_tables){
406     global $wordpress,$wpdb;
407
408     clean_buffer_folder();
409     $con_info_table = fopen("export_data/contrail_info.csv", "w+");
410     fputcsv($con_info_table, array("table_name", "value",
    •     "contrail_lasting_time", "model", "icao24"));
411     foreach ($array_of_tables as $index => $tables) {
412         $sql = "SELECT * FROM contrail_info about tables WHERE

```

```

412     $sql = SELECT * FROM contrail_info_about_tables WHERE
      • table_name='$array_of_tables[$index]';
413
414     $results = $wpdb->get_results($sql, 'ARRAY_A');
415     foreach ($results as $result) {
416         fputcsv($con_info_table, $result);
417     }
418 }
419 fclose($con_info_table);
420 echo "<h4> Link to contrail info csv of tables in interval: </h4>";
421 echo "<a href='https://www.3219046688.adsb-projekt-vv.cz/
      • export_data/contrail_info.csv'>contrail_info.csv</a>";
422 }
423

```

Obr. 35: Zdrojový kód aplikační vrstvy pro export dat z databáze


```

1  if (isset($_POST['submit_csv'])) {
2      global $wordpress,$wpdb;
3
4      $file = $_FILES['fileToUpload']['tmp_name'];
5      $target_dir = $_SERVER['DOCUMENT_ROOT']."/domains/3219046688.adsb-
•   projekt-vv.cz/data";
6      $file_name = basename($_FILES["fileToUpload"]["name"]);
7      $target_file = $target_dir . '/' . $file_name;
8      $files = glob($target_dir . '/*');
9      move_uploaded_file($file, $target_file);
10     $ext_csv = end(explode(".", $file_name));
11     $trimmed_csv_name = str_replace(".", '', $file_name) ;
12
13     if ($ext_csv == "csv") {
14         $drop_table = "DROP TABLE `{$trimmed_csv_name}`";
15         $create_table = "CREATE TABLE `{$trimmed_csv_name}` (
16             `Unix_time` int(20) NOT NULL,
17             `Message` text CHARACTER SET latin1 NOT NULL,
18             `Time` varchar(17) COLLATE latin1_bin NOT NULL,
19             `DF` int(11) NOT NULL,
20             `Type_code/BDS_number` int(11) NOT NULL,
21             `Latitude` double NOT NULL,
22             `Longitude` double NOT NULL,
23             `Altitude` int(11) NOT NULL,
24             `Ground_speed` double NOT NULL,
25             `Baro_diff` int(11) NOT NULL,
26             `Heading` int(11) NOT NULL,
27             `TAS` int(11) NOT NULL,
28             `Mach` float NOT NULL,
29             `FOM_SOURCE` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
30             `Wind_speed` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
31             `Wind_direction` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
32             `Temperture` text CHARACTER SET latin1,
33             `Pressure` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
34             `Turbulence` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
35             `Humidity` varchar(40) CHARACTER SET latin1 DEFAULT NULL
36         ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_bin;";
37
38         $load_csv_file="
39             LOAD DATA LOCAL INFILE ''.$target_file.'' INTO TABLE
•         ''.$trimmed_csv_name.''
40             FIELDS TERMINATED BY ';' IGNORE 1 ROWS;";
41
42         $query = $wpdb->query($drop_table);
43         $query = $wpdb->query($create_table);
44         $query = $wpdb->query($load_csv_file);
45         echo "<script type='text/javascript'>alert('Upload to database
•         was successful!');</script>";
46

```

```
47     foreach($files as $file){
48         if(is_file($file))
49             unlink($file);
50     }
51 } else {
52     echo "<script type='text/javascript'>alert('Not a .csv
•     file!');</script>";
53 }
54 }
55
```

Obr. 36: Zdrojový kód aplikační vrstvy, používaný pro nahrání nového „.csv“ souboru

```

1  if (isset($_POST['submit_zip'])) {
2      global $wordpress,$wpdb;
3
4      $file = $_FILES['zipFileToUpload']['tmp_name'];
5      $target_dir = $_SERVER['DOCUMENT_ROOT']."/domains/3219046688.adsb-
•   projekt-vv.cz/data";
6      $file_name = basename($_FILES["zipFileToUpload"]["name"]);
7      $target_file = $target_dir . '/' . $file_name;
8      $files = glob($target_dir . '/*');
9      move_uploaded_file($file, $target_file);
10     $ext_zip = end(explode(".", $file_name));
11     $trimmed_zip_name = str_replace(".", $ext_zip, $file_name);
12
13     if ($ext_zip == "zip") {
14         $zip = new ZipArchive;
15         $zip->open($target_file);
16         $zip->extractTo($target_dir);
17         $zip->close();
18
19         foreach (new DirectoryIterator($target_dir) as $fileInfo) {
20             if($fileInfo->isDot()) continue;
21             $tmp_file_name = $fileInfo->getFilename();
22             $tmp_ext_csv = end(explode(".", $tmp_file_name));
23             $target_csv_file = $target_dir . '/' . $tmp_file_name;
24
25             if ($tmp_ext_csv == "csv") {
26                 $trimmed_csv_name = str_replace(".", $tmp_ext_csv, $
•   tmp_file_name);
27                 $drop_table = "DROP TABLE `{$trimmed_csv_name}`";
28                 $create_table = "CREATE TABLE `{$trimmed_csv_name` (
29                     `Unix_time` int(20) NOT NULL,
30                     `Message` text CHARACTER SET latin1 NOT NULL,
31                     `Time` varchar(17) COLLATE latin1_bin NOT NULL,
32                     `DF` int(11) NOT NULL,
33                     `Type_code/BDS_number` int(11) NOT NULL,
34                     `Latitude` double NOT NULL,
35                     `Longitude` double NOT NULL,
36                     `Altitude` int(11) NOT NULL,
37                     `Ground_speed` double NOT NULL,
38                     `Baro_diff` int(11) NOT NULL,
39                     `Heading` int(11) NOT NULL,
40                     `TAS` int(11) NOT NULL,
41                     `Mach` float NOT NULL,
42                     `FOM_SOURCE` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
43                     `Wind_speed` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
44                     `Wind_direction` varchar(40) CHARACTER SET latin1 DEFAULT
•   NULL,
45                     `Temperture` text CHARACTER SET latin1,
46                     `Pressure` varchar(40) CHARACTER SET latin1 DEFAULT NULL,

```



```

47         `Turbulence` varchar(40) CHARACTER SET latin1 DEFAULT NULL,
48         `Humidity` varchar(40) CHARACTER SET latin1 DEFAULT NULL
49     ) ENGINE=InnoDB DEFAULT CHARSET=latin1;";
50
51     $load_csv_file="
52     LOAD DATA LOCAL INFILE '$target_csv_file.'" INTO TABLE
53     • "$trimmed_csv_name."
54         FIELDS TERMINATED BY ',' IGNORE 1 ROWS;";
55     $query = $wpdb->query($drop_table);
56     $query = $wpdb->query($create_table);
57     $query = $wpdb->query($load_csv_file);
58     }
59
60     foreach($files as $file){
61         if(is_file($file))
62             unlink($file);
63     }
64     echo "<script type='text/javascript'>alert('Upload to database
65     • was successful!');</script>";
66 } else {
67     echo "<script type='text/javascript'>alert('Not a .zip
68     • file!');</script>";
69 }

```

Obr. 37: Zdrojový kód aplikační vrstvy, používaný pro nahrání „.zip“ archivu, který obsahuje „.csv“ soubory

9.2 Zdrojový kód prezenční vrstvy

Kód pro samotnou prezenční vrstvu (tedy UI a UX, které je uživateli k dispozici a ze kterého se spouští PHP kód vrstvy aplikační) je vyobrazen na obrázcích 38, 39, 40 a 41. Na obrázcích 38, 39 a 40 je zdrojový kód v jazyce HTML a na obrázku 41 je vlastní nastavení kaskádových stylů – CSS. CSS zbytku webové stránky je zajišťováno Wordpress šablonou a není nutné jej zde uvádět.

```

1 <script>
2     function detectBrowserAndDenyAccessWithIt() {
3         if(navigator.userAgent.indexOf("Safari") != -1) {
4             for(var i=0;;i++)
5                 alert('Safari does not support some of HTML5 elements. The
6                 • application cannot be used in Safari – please use another
7                 • browser. For more info, contact vaclavvidon@gmail.com.');
```

```

6         } else if (navigator.appName == 'Microsoft Internet Explorer'
7         • || !(navigator.userAgent.match(/Trident/) ||
8         • navigator.userAgent.match(/rv:11/)) || (typeof $.browser !=
9         • "undefined" && $.browser.msie == 1)) {
10            for(var i=0;;i++)
11                alert('Internet Explorer does not support some of HTML5
12                • elements. The application cannot be used in Internet
13                • Explorer – please use another browser. For more info,
14                • contact vaclavvidon@gmail.com.');
```

```

9        }
10    }
11
12    detectBrowserAndDenyAccessWithIt();
13 </script>
14
15 <h2>Date selection:</h2>
16
17 <form method="post">
18     <div id="search_option_container">
19         <span id="search_subheaders">From: </span>
20         <input type="date" name="dateFrom" value="<?php echo date('Y-m-
21         • d');?>"> </input>
22     </div>
23     <div id="search_option_container">
24         <span id="search_subheaders">To: </span>
25         <input type="date" name="dateTo" value="<?php echo date('Y-m-d'); ?>">
26     </div>
27 <div>
28     <input type="submit" name="save_date" value="Submit">
29 </div>
30 </form>
31
32 <table><tbody><tr><th>ICAO address</th><th>Initial time</th><th>Final
33     • time</th><th>Initial lat </th><th>Final lat</th><th>Initial lon</
34     • th><th>Final lon</th><th>Initial hdg</th><th>Final hdg</th></tr></
35     • tbody></table>
36
37 [wbc_r_php_snippet id="71"] //this imports and runs the filter data PHP

```

Obr. 38: Zdrojový HTML kód stránky „Filter results by date“

```

1 <script>
2     function detectBrowserAndDenyAccessWithIt() {
3         if(navigator.userAgent.indexOf("Safari") != -1) {
4             for(var i=0;;i++)
5                 alert('Safari does not support some of HTML5 elements. The
6                 • application cannot be used in Safari – please use another
7                 • browser. For more info, contact vaclavvidon@gmail.com.');
```

```

6         } else if (navigator.appName == 'Microsoft Internet Explorer'
7         • || !(navigator.userAgent.match(/Trident/) ||
8         • navigator.userAgent.match(/rv:11/)) || (typeof $.browser !==
9         • "undefined" && $.browser.msie == 1)) {
10            for(var i=0;;i++)
11                alert('Internet Explorer does not support some of HTML5
12                • elements. The application cannot be used in Internet
13                • Explorer – please use another browser. For more info,
14                • contact vaclavvidon@gmail.com.');
```

```

9        }
10    }
11
12    detectBrowserAndDenyAccessWithIt();
13 </script>
14
15 <h2>Export selection:</h2>
16 <form method="post">
17     <div id="search_option_container">
18         <span id="search_subheaders">From: </span><input type="date"
19         • name="dateFromExport" value="<?php echo date('Y-m-d'); ?>">
20     </div>
21     <div id="search_option_container">
22         <span id="search_subheaders">To: </span><input type="date"
23         • name="dateToExport" value="<?php echo date('Y-m-d'); ?>">
24     </div>
25
26     <input type="submit" name="save_date_export" value="Vypsat">
27     <br> <br>
28
29     <div>
30         <input type="radio" name="tables_in_range"
31         • value="merged_contrail_data" checked="">
32         <label for="tables_in_range">Links to merged contrail data</label>
33     </div>
34
35     <div>
36         <input type="radio" name="tables_in_range"
37         • value="whole_tables_in_range">
38         <label for="tables_in_range">Links to whole tables in interval
39     </label>
40
41     </div>

```

```

38
39 <div>
40 <input type="radio" name="tables_in_range"
  • value="contrail_data_in_range">
41 <label for="tables_in_range">Link to contrail information data in
  • interval</label>
42 </div>
43 <br>
44
45 <input type="submit" name="save_date_export" value="Submit">
46 <br><br>
47 </form>
48
49 [wbcr_php_snippet id="8"] //this imports and runs the export data PHP
50

```

Obr. 39: Zdrojový HTML kód stránky „Export ADS-B data“

```

1 <script>
2     function detectBrowserAndDenyAccessWithIt() {
3         if(navigator.userAgent.indexOf("Safari") != -1) {
4             for(var i=0;;i++)
5                 alert('Safari does not support some of HTML5 elements. The
        • application cannot be used in Safari – please use another
        • browser. For more info, contact vaclavvidon@gmail.com.');
```

```

6         } else if (navigator.appName == 'Microsoft Internet Explorer'
        • || !(navigator.userAgent.match(/Trident/) ||
        • navigator.userAgent.match(/rv:11/)) || (typeof $.browser !=
        • "undefined" && $.browser.msie == 1)) {
7             for(var i=0;;i++)
8                 alert('Internet Explorer does not support some of HTML5
        • elements. The application cannot be used in Internet
        • Explorer – please use another browser. For more info,
        • contact vaclavvidon@gmail.com.');
```

```

9         }
10    }
11
12    detectBrowserAndDenyAccessWithIt();
13 </script>
14
15 <form action="" method="post" enctype="multipart/form-data">
16     <h4><p>Choose your CSV file</p></h4><br>
17     <input type="file" name="fileToUpload" id="fileToUpload">
18     <input type="submit" value="Upload CSV" name="submit_csv">
19 </form>
20 <br>
21 <form action="" method="post" enctype="multipart/form-data">
22     <h4><p>Choose your ZIP file</p></h4><br>
23     <input type="file" name="zipFileToUpload" id="zipFileToUpload">
24     <input type="submit" value="Upload ZIP" name="submit_zip">
25 </form>
26
27 [wbc_r_php_snippet id="11"] //this imports and runs the import single
    • ".csv" PHP
28 [wbc_r_php_snippet id="12"] //this imports and runs the import archive
    • of ".csv" files PHP
29
```

Obr. 40: Zdrojový HTML kód stránky „Upload files to database“

```

1  input[type=time] {
2    width: 20%;
3    padding-top: 9.25px;
4  }
5
6  input[type=text] {
7    width: 32%;
8    box-sizing: border-box;
9    padding-top: 9.5px;
10 }
11
12 input[type=submit] {
13   margin-top: 10px;
14   margin-bottom: 10px;
15   box-sizing: border-box;
16   height: 40px;
17 }
18
19 input[type=date] {
20   margin-bottom: 0px;
21   height: 40px;
22   box-sizing: border-box;
23 }
24
25 table {
26   margin-top: 5px;
27   margin-bottom: 0px;
28 }
29
30 hr {
31   margin-top: 0px;
32   margin-bottom: 0px;
33 }
34

```

Obr. 41: Vlastní CSS, používané pro všechny stránky webové aplikace

9.3 Zdrojový kód datové vrstvy

K samotnému kódu datové vrstvy (tedy například k metodikám ukládání dat, funkčnosti SQL příkazů a dalším) nemá běžný uživatel přístup ani programátor a není nutno jej zde prezentovat. Pro práci s daty a komunikaci s datovou vrstvou je používáno aplikační vrstvy, jak je například zobrazeno na obrázku 36, řádek 42 (podle čísel levého sloupce na obrázku) - na tomto řádku je do databáze vyslán příklad pro smazání tabulky.

10 ZÁVĚR

V rámci diplomové práce byla vytvořena webová aplikace s databází pro ukládání, zpracovávání, správu, doplňování a export informací, které jsou generovány z přijímaných dat o sledovaných letadlech a jsou zpracovávány do formy tabulek. Dále byl vytvořen manuál k této webové aplikaci, který popisuje, jak má uživatel s aplikací pracovat. Vypracované zdrojové kódy aplikace jsou přiloženy v kapitole 9.

Vstupní datové tabulky pro webovou aplikaci jsou generovány systémem Ústavu letecké dopravy, Fakulty Dopravní ČVUT v Praze, a jsou tvořeny spojením BDS registrů a komunikace, přijímané na frekvenci 1090 MHz, ze které jsou filtrovány ADS-B zprávy. Takto vytvořená tabulka je posléze aplikací zpracovávána. Uživatel má k dispozici rozhraní, ve kterém může vybrat a do databáze uložit, zda kondenzační stopa vznikla, jak se chovala, jak dlouho byla viditelná, o jaký letoun se jednalo a má k dispozici důležité informace – časové, polohové, identifikační a další pro daný let. Mezi další funkcionality patří nahrávání nových tabulek a jejich balíčků do databáze a export zvolených tabulek s uživatelem uloženými informacemi podle požadavků Fakulty Dopravní ČVUT v Praze pro jejich další zpracování.

Pro vytvoření bylo použito třívrstvého modelu aplikací, který dělí aplikaci na tři samostatné vrstvy – prezenční, datovou a aplikační. Uživateli je viditelná pouze vrstva prezenční, se kterou může pracovat dle své vůle. Samotné výpočty a práce s daty od uživatele a s daty z databáze se potom dějí na vrstvě aplikační, která není uživateli viditelná ani přístupná a může s ní pracovat pouze pomocí vrstvy prezenční. Pro ukládání dat tabulek a záznamů o tom, jak se stopa chovala, slouží vrstva datová, která je uživateli přístupná pouze skrz výstup dat do vrstvy prezenční z vrstvy aplikační. Pro vytvoření byly použity programovací jazyky HTML, PHP a SQL (HTML pro vytvoření základů webové stránky, PHP pro vykonávání požadavků uživatele a pro práci s daty a SQL pro vytváření a úpravu dat v databázi).

Při tvorbě aplikace bylo nutno v práci popsat samotný vznik kondenzačních stop, jejich chování, meteorologické a jiné vlastnosti, jakožto i jednotlivé druhy leteckých radarů, odpovídačů, ADS-B systémů a programovacích jazyků, jejich funkce a použití, aby bylo dostatečně vysvětleno, jak jsou informace o prolétajícím letadle získávány a kde všude se s nimi pracuje.

Diplomová práce může být využita pro zpracovávání dat o zaznamenaných letadlech a kondenzačních stopách, které tato letadla vytvořila. Užití práce není limitované jen pro interní účely Ústavu letecké dopravy, může být využita i pro studijní účely odborné veřejnosti, jelikož vysvětluje princip vzniku kondenzačních stop, činnosti leteckých radarů, odpovídačů, systému ADS-B a programovacích jazyků, ve kterých aplikace vznikla.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] PALEČKOVÁ, Lucie. ANALÝZA KONDENZAČNÍCH STOP PROSTŘEDNICTVÍM DAT VYSÍLANÝCH LETADLY. Praha, 2018. Magisterská práce. ČVUT v Praze. Vedoucí práce Jakub Kraus.
- [2] SOBOTA, J. and KRÁČMAR, J. Meteorologie. 1. ed. Brno: Akademické nakladatelství CERM. 2002, ISBN 80-7204-238-6.
- [3] Portál ČHMÚ : Aktuální situace : Aktuální stav počasí : Informace a zajímavosti : Kondenzační stopy. Portál ČHMÚ : Home [online]. Dostupné z: <http://portal.chmi.cz/aktualni-situace/aktualni-stav-pocasi/informace-a-zajimovosti/kondenzacni-stopy>
- [4] Izobarické změny vlhkého vzduchu [online]. 1s [cit. 2019-03-13]. Dostupné z: https://kke.zcu.cz/export/sites/kke/old_web/_files/projekty/enazp/04/IUT/027_Izobaricke_zmeny_vlhkeho_vzduchu_-_P1.pdf
- [5] VLHKÝ VZDUCH – STAVOVÉ VELIČINY: Aplikovaná termomechanika - Vlhký vzduch, Izobarické změny stavů vlhkého vzduchu [online]. 1-14s [cit. 2019-03-13]. Dostupné z: <http://tzb.fsv.cvut.cz/vyucujici/14/vlhky-vzduch.pdf>
- [6] D. PUDA, David a Patrick MINNIS. Basic Diagnosis and Prediction of Persistent Contrail Occurrence Using High-Resolution Numerical Weather Analyses/Forecasts and Logistic Regression.: Part II: Evaluation of Sample Models [online]. 13.02.2009, 1-13s [cit. 2019-03-13]. Dostupné z: <https://journals.ametsoc.org/doi/pdf/10.1175/2009JAMC2057.1>
- [7] WORLD METEOROLOGICAL ORGANIZATION. Guide to meteorological instruments and methods of observation [online]. 6th ed. Geneva, Switzerland: Secretariat of the World Meteorological Organization, 1996, 716 p. [cit. 2019-03-13]. WMO (Series), no. 8. ISBN 92- 631-6008-2. Available from: https://library.wmo.int/pmb_ged/wmo_8_en-2012.pdf
- [8] BEDNÁŘ, Jan. Meteorologie: Úvod do studia dějů v zemské atmosféře. Praha: Portál, 2003. ISBN 80-717-8653-5
- [9] DVOŘÁK, Petr. Pozorování a předpovědi počasí. Cheb: Svět křídel, 2012. ISBN 978-80-87567-20-3.
- [10] BEZOUSEK, Pavel, SCHEJBAL, Vladimír: Radar Technology in the Czech Republic [online dokument formátu PDF], IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE, August 2004 [cit. 2019-03-13]. Dostupné z: <https://www.semanticscholar.org/paper/Radar-technology-in-the-Czech-Republic-Bezou%C5%A1ek-Schejbal/15b123ac0053aa2f4ff749014234157969fdb4e8>

- [11] HATZIPANAGOS, Georgie: Kamara and Merenda Hill Terminal Area Radar [online], c1997-2005, [cit. 2019-03-13]. Dostupné z:
http://www.hcaa-eleng.eu/en/systems/star2000_en.html
- [12] HAVELKA, Radek: CZK - RL-4AS [online], c1997-2008, [cit. 2019-03-13].
 Dostupné z: <http://forum.valka.cz/viewtopic.php/title/CZK-RL-4AS/t/15784>
- [13] KULČÁK, Ludvík a kol.: Air Traffic Management, Brno, Akademické nakladatelství CERM, 2002, 314s. ISBN 80-7204-229-7
- [14] KULČÁK, Ludvík: Klasické prostriedky zabezpečovacej leteckej techniky, Žilina, EDIS – vydavateľství ŽU, 1998, 358s. ISBN 80-7100-477-4
- [15] RENDALL, David: Jane's Letadla: Příručka pro rozpoznávání letadel, přeložil Tomáš Moravec, české první vydání, Praha, Jan Vašut, 1999, 512s. ISBN 80-7236-078-7
- [16] Statistics. Document Moved [online]. Copyright © International Civil Aviation Organization [cit. 17.05.2019].
 Dostupné z: <https://www.icao.int/sustainability/Pages/Statistics.aspx>
- [17] VOSECKÝ, Slavomír a kol.: Základy leteckých navigačních zařízení, Brno, Vojenská akademie, 1988, 750s.
- [18] FREI, J. Basic Evaluation of Meteorological Observation from Aircraft. In: Zvyšovanie bezpečnosti a kvality v civilnom letectve 2016. Zvyšovanie bezpečnosti a kvality v civilnom letectve 2016. Zuberec, 27.01.2016 - 29.01.2016. Žilina: Žilinská univerzita. 2016, pp. 16-21. ISBN 978-80-554-1143-9.
- [19] U.S. ENVIRONMENTAL PROTECTION AGENCY. Aircraft Contrails Factsheet. Washington, D.C., 2000, 6 s[online]. Copyright © [cit. 08.05.2019]. Dostupné z:
https://www.faa.gov/regulations_policies/policy_guidance/envir_policy/media/contrails.pdf
- [20] LÁN, Sébastien. VZNIK A ANALÝZA KONDENZAČNÍCH STOP LETADEL [online]. Praha, 2017 [cit. 2019-03-13]. Dostupné z:
<https://dspace.cvut.cz/bitstream/handle/10467/70467/F6-DP-2017-Lan-Sebastien-dp.pdf?sequence=1&isAllowed=y>. Diplomová práce. České vysoké učení technické v Praze.
- [21] KOHUTEK, JAKUB. PŘEHLEDOVÉ RADARY V ČR: CZECH SURVEILLANCE RADARS [online]. Brno, 2009 [cit. 2019-03-13]. Dostupné z:
https://dspace.vutbr.cz/bitstream/handle/11012/13138/BP_2009_Prehledove_radary_v_CR_Kohutek.pdf?sequence=1&isAllowed=y. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. SLAVOMÍR VOSECKÝ, CSc.

- [22] *Předpis L10* [online]. In: [cit. 2019-03-13]. Dostupné z:
<https://lis.rlp.cz/predpisy/predpisy/index.htm>
- [23] LAZARIDIS, Giorgos. Pulse Position Modulation and Differential PPM.
[Http://www.pcbheaven.com](http://www.pcbheaven.com) [online]. 2011, 22.6.2011 [cit. 2019-03-03]. Dostupné z:
http://www.pcbheaven.com/wikipages/Pulse_Position_Modulation
- [24] ADS-B Technology basics [online]. Copyright © [cit. 2019-03-13]. Dostupné z:
<https://www.icao.int/APAC/Meetings/2010/adsb/sp14.pdf>
- [25] European Airlines Not on Track to Meet 2020 ADS-B Mandate - Avionics. Avionics - The Pulse of Aviation Electronics [online]. Copyright © 2019, Access Intelligence, LLC. All rights reserved. Learn more about licensing or reproducing content on this site. [cit. 13.03.2019]. Dostupné z: <https://www.aviationtoday.com/2018/07/06/european-airlines-not-track-meet-2020-ads-b-mandate/>
- [26] *ADS-B and other means of surveillance implementation status* [online]. 15.05.2018, 1-56s [cit. 2019-03-13]. Dostupné z:
<https://ec.europa.eu/transport/sites/transport/files/20180515-sesar-ads-b-report.pdf>
- [27] *SEASONAL TECHNICAL COMMUNICATION* [online]. 01.06.2018, 1-16s [cit. 2019-03-13]. Dostupné z:
https://www.easa.europa.eu/sites/default/files/dfu/EASA_STC_NEWS_JUNE_2018.pdf
- [28] Jak se loví letadla aneb Přijímáme ADS-B | brichacek.net | [online]. Dostupné z:
<https://blog.brichacek.net/jak-se-lovi-letadla-aneb-prijimame-ads-b/>
- [29] DUNSTONE, Greg. *ADS-B Introduction: Surveillance Program Lead, Airservices Australia* [online]. 2012, 1-35s [cit. 2019-03-13]. Dostupné z:
https://www.icao.int/APAC/Meetings/2012_SEA_BOB_ADSB_WG8/SP01_AUS%20-%20ADS-B%20Basics.pdf
- [30] Mervart L. Globální polohový systém. Skriptum FSTA ČVUT Praha. Vydavatelství ČVUT Praha 1994.
- [31] *RADIOLOKACE: Radiolokace a radionavigace* [online]. 2004, 1-131s [cit. 2019-03-13]. Dostupné z:
http://www.urel.feec.vutbr.cz/~sebestaj/RAR/literatura/Radiolokace_a_radionavigace_2004.pdf
- [32] Automatic Dependent Surveillance Broadcast (ADS-B) - SKYbrary Aviation Safety. [online]. Dostupné z:
[https://www.skybrary.aero/index.php/Automatic_Dependent_Surveillance_Broadcast_\(ADS-B\)](https://www.skybrary.aero/index.php/Automatic_Dependent_Surveillance_Broadcast_(ADS-B))

- [33] UL HAQ, Fahim. *The top data structures you should know for your next coding interview* [online]. 30.07.2018, , 1s [cit. 2019-03-13]. Dostupné z: <https://medium.freecodecamp.org/the-top-data-structures-you-should-know-for-your-next-coding-interview-36af0831f5e3?gi=340567ddd9e9>
- [34] ZINDULKA, J. *Databázové systémy: Relační model dat* [online]. 30.07.2018, 1s [cit. 2019-03-13]. Dostupné z: https://www.fit.vutbr.cz/study/courses/DSI/public/pdf/nove/4_relmod.pdf
- [35] *RELAČNÍ DATOVÝ MODEL* [online]. 1-45s [cit. 2019-03-13]. Dostupné z: <http://wiki.cs.vsb.cz/images/2/2a/lnz2.pdf>
- [36] What is a multi layered software architecture? | Packt Hub. Technology News, Insights and Tutorials | Packt Hub [online]. Copyright © Packt Publishing Ltd [cit. 13.03.2019]. Dostupné z: <https://hub.packtpub.com/what-is-multi-layered-software-architecture/>
- [37] 3-Tier Architecture: A Complete Overview - JReport. Embedded Analytics and Business Intelligence Software [online]. Copyright © 2019 Jinfonet Software, Inc. All rights reserved. [cit. 13.03.2019]. Dostupné z: <https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/>
- [38] DUCKETT, Jon. *HTML and CSS: Design and Build Websites*. Indianapolis, IN: Wiley, 2011. ISBN 11-180-0818-9.
- [39] MACDONALD, Matthew. *HTML5: The Missing Manual*. Second edition. Beijing: O'Reilly, 2013. Missing manual. ISBN 14-493-6326-1.
- [40] KRAUS, Vladimír a Jaroslav HAJNÝ. *SQL pro blbce*. 2. rozšířené vydání. Praha: Palmknihy, 2014. ISBN 9788074860034.
- [41] The 1090MHz Riddle. Home for Mode-S and ADS-B [online]. Dostupné z: <https://mode-s.org/decode/ehs/introduction.html#bds-comm-b-data-selector>
- [42] Historical trends in the usage of server-side programming languages, March 2019. W3Techs - extensive and reliable web technology surveys [online]. Copyright © 2009 [cit. 17.03.2019]. Dostupné z: https://w3techs.com/technologies/history_overview/programming_language
- [43] FAA reopens \$500 ADS-B rebate program – key takeaways. Home - iPad Pilot News [online]. Copyright © Copyright 2013 [cit. 16.03.2019]. Dostupné z: <https://ipadpilotnews.com/2018/10/faa-reopens-ADS-B-rebate/>

- [44] The Ins and Outs of the ADS-B DO-260B Compliance. Honeywell Aerospace [online]. Dostupné z: <https://aerospace.honeywell.com/en/news-listing/2015/october/the-ins-and-outs-of-the-ads-b-do-260b-compliance>
- [45] Three-tier architecture - ManagementMania.com. [online]. Copyright © 2011 [cit. 16.03.2019]. Dostupné z: <https://managementmania.com/en/three-tier-architecture>
- [46] phpMyAdmin - Try. phpMyAdmin [online]. Copyright © 2003 [cit. 16.03.2019]. Dostupné z: <https://www.phpmyadmin.net/try/>
- [47] Tryit Editor v3.6 - Show PHP. W3Schools Online Web Tutorials [online]. Dostupné z: https://www.w3schools.com/pHp/showphp.asp?filename=demo_syntax_case2
- [48] php - What is a Unix timestamp and why use it? - Stack Overflow. Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. Dostupné z: <https://stackoverflow.com/questions/20822821/what-is-a-unix-timestamp-and-why-use-it>
- [49] LERDORF, Rasmus a Kevin TATROE. Programming PHP. Sebastopol, CA: O'Reilly, c2002. ISBN 978-1565926103.

SEZNAM OBRÁZKŮ

- Obrázek 1. Čerstvá kondenzační stopa za letadlem
- Obrázek 2. Rozfoukaná kondenzační stopa
- Obrázek 3. „Černé“ kondenzační stopy na obloze, způsobené západem Slunce
- Obrázek 4. Mollerův h-x diagram vlhkého vzduchu pro tlak 100 kPa
- Obrázek 5. Obálky vysokofrekvenčních signálů vysílače PSR
- Obrázek 6. Komunikace dotazovač-odpovídač
- Obrázek 7. Dotaz radaru SSR
- Obrázek 8. Odpověď na dotaz SSR
- Obrázek 9. Vyzařovací diagram monopulzního sekundárního radaru
- Obrázek 10. Zobrazení intenzity přijímaného signálu monopulzního sekundárního radaru
- Obrázek 11. Modulace dat do schématu PPM pro přenos
- Obrázek 12. Předávání dat pomocí ADS-B vysílačů/přijímačů
- Obrázek 13. Plánovaná data zavádění povinnosti osazení letadel systémy ADS-B
- Obrázek 14. Příklad unixových dat
- Obrázek 15. Druhy informací, obsažených v ADS-B zprávě a jejich intervaly vysílání
- Obrázek 16. Ukázka dat typu „message“
- Obrázek 17. Ukázka dat typu „Time“
- Obrázek 18. Některé číselné označení typu ADS-B zprávy
- Obrázek 19. Příklad dat o zeměpisné šířce, délce a výšce
- Obrázek 20. Grafické vyobrazení propojení jednotlivých vrstev, při použití třívrstvé architektury aplikace
- Obrázek 21. Grafické vyobrazení jednotlivých vrstev třívrstvé architektury a nejčastěji používaných programovacích jazyků pro jednotlivé vrstvy
- Obrázek 22. Struktura dat typu „tree“ s vyobrazením jednotlivých úrovní
- Obrázek 23. Uživatelské rozhraní populárního nástroje phpMyAdmin, které slouží pro správu aplikační a datové vrstvy. Na obrázku zobrazena tabulka dat uživatelů této administrace

- Obrázek 24. Příklad zdrojového HTML kódu webové stránky.
- Obrázek 25. Tabulka „vyrobky“
- Obrázek 26. Hodnoty, vybrané z databáze pomocí query
- Obrázek 27. Příklad příkazů jazyka PHP na HTML stránce (vlevo) a jejich překlad uživateli (vpravo)
- Obrázek 28. Obrazovka „Filter results by date“ bez výsledků vyhledávání
- Obrázek 29. Výsledky vyhledávání pro datum „6.září 2018“
- Obrázek 30. Export .csv tabulek pro zadaný časový interval a pro možnost „Links to merged contrail data“
- Obrázek 31. Výsledná vyexportovaná tabulka, obsahující hlavní informace z tabulek ze zvoleného časového intervalu
- Obrázek 32. Stránka „Upload files to database“
- Obrázek 33. Fotka vzorové tabulky, která může být nahrána do databáze bez problémů
- Obrázek 34. Zdrojový kód aplikační vrstvy, používaný pro filtrování tabulek a hlavních údajů z těchto tabulek
- Obrázek 35. Zdrojový kód aplikační vrstvy pro export dat z databáze
- Obrázek 36. Zdrojový kód aplikační vrstvy, používaný pro nahrání nového „.csv“ souboru
- Obrázek 37. Zdrojový kód aplikační vrstvy, používaný pro nahrání „.zip“ archivu, který obsahuje „.csv“ soubory
- Obrázek 38. Zdrojový HTML kód stránky „Filter results by date“
- Obrázek 39. Zdrojový HTML kód stránky „Export ADS-B data“
- Obrázek 40. Zdrojový HTML kód stránky „Upload files to database“
- Obrázek 41. Vlastní CSS, používané pro všechny stránky webové aplikace

SEZNAM TABULEK

Tabulka 1. Relace, obsahující některé druhy chyb