

UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I RAČUNARSTVO

**Simulacija rada Flight Management Computer-a
aviona tipa Boeing 737NG**

-Diplomski rad-

Mentor:

prof. dr Mladen Veinović

Student:

Kristian Nikolić

Br. indeksa: 2011200415

Beograd, 2015

FAKULTET ZA INFORMATIKU I RAČUNARSTVO

UNIVERZITET SINGIDUNUM

FAKULTET ZA INFORMATIKU I RAČUNARSTVO

Beograd, Danijelova 32

Kandidat: Kristian Nikolić

Broj indeksa: 2011/200415

Smer: Informatika i računarstvo

Tema: Simulacija rada Flight Management Computer-a aviona tipa Boeing 737NG

MENTOR

Datum odobrenja rada:

Beograd, ____ . ____ . ____ .

Prof. dr Mladen Veinović

DEKAN

Prof. dr Dragan Cvetković

Sadržaj

1	Uvod	- 5 -
2	Računarska simulacija	- 6 -
2.1	Modeliranje i simulacija	- 6 -
3	Tehnologije korišćene za izradu projekta	- 9 -
3.1	PHP programski jezik	- 9 -
3.2	HTML	- 11 -
3.3	JavaScript	- 12 -
4	Baza podataka	- 13 -
4.1	MySQL	- 13 -
4.2	Flight Management System – Princip rada	- 14 -
4.3	Model baze podataka	- 15 -
4.4	Realizacija i objašnjenje tabela	- 17 -
4.4.1	Tabela Airports	- 17 -
4.4.2	Tabela Ilses	- 19 -
4.4.3	Tabela Nav aids	- 21 -
4.4.4	Tabela Runways	- 23 -
4.4.5	Tabela Terminals	- 25 -
4.4.6	Tabela Waypoints	- 27 -
4.4.7	Tabela Airways	- 29 -
5	Programska realizacija	- 32 -
5.1	Strana IDENT	- 32 -
5.1.1	Programska realizacija strane	- 33 -
5.2	INIT/REF INDEX strana	- 36 -
5.2.1	Programska realizacija strane	- 37 -
5.3	POSITION INIT strana	- 38 -
5.3.1	Programska realizacija strane	- 39 -
5.4	ROUTE strana	- 45 -
5.4.1	Programska realizacija strane	- 45 -
5.5	PERF INIT strana	- 49 -
5.5.1	Programska realizacija strane	- 50 -
5.6	N1 LIMIT strana	- 52 -
5.6.1	Programska realizacija strane	- 54 -
5.7	DEP/ARR Index strana	- 54 -
5.7.1	Programska realizacija strane	- 56 -
5.8	Take off reference page	- 57 -
5.8.1	Programska realizacija strane	- 58 -

5.9	Climb page.....	- 59 -
5.9.1	Programska realizacija strane.....	- 60 -
5.10	Cruise page.....	- 61 -
5.10.1	Programska realizacija strane.....	- 61 -
5.11	Descent page.....	- 62 -
5.11.1	Programska realizacija strane.....	- 63 -
6	Zaključak	- 64 -
7	Literatura.....	- 66 -

1 Uvod

Tema diplomskog rada je simulacija i opis principa rada glavnog računara u avionu, koji za cilj ima proračune performansi letelice i navigacije, sa akcentom na izradi simulacije samog računara koji te funkcije i obavlja.

Diplomski rad, pored opisa koji se odnose na radi realizaciju, sadrži i implementaciju programa za simulaciju Flight Management Computer-a (u daljem tekstu FMC). Program je detaljno opisan, svaka stranica (osim onih koje parametre dobijaju od određenih instrumenata) je detaljno objašnjena i simulirana. Izvorni kod svake stranice priložen je u celosti, a pojedini delovi su posebno objašnjeni.

Korišćene tehnologije su:

- HTML5,
- CSS3,
- JavaScript,
- PHP,
- MySQL.

Za frontend su zaduženi HTML5 i CSS3, dok su za backend korišćeni JavaScript i PHP u kombinaciji sa MySQL-om.

Kada su u pitanju razvojna okruženja, korišćeni su Notepad++ za programiranje i Navicat za modelovanje i izradu baze podataka.

Glavni cilj izrade ove aplikacije je mogućnost treninga letачkog osoblja i mogućnost ljudima koji se ne bave vazduhoplovstvom, a imaju želju da nauče kako funkcionišu pojedine stvari u savremenim avionima.

Simulatori su jako korisni, jer omogućavaju aktivno učenje i predstavljaju bolji način da se određene stvari razumeju, jer za razliku od učenja iz knjige sa kojom učenik nema toliko interakciju sa „opipljivim“ stvarima, na simulatoru može odmah videti šta se dešava i lakše upamtiti redosled rada.

Ukoliko je simulator dostupan i na Internetu, omogućava širokom spektru ljudi da koriste njegove mogućnosti i nauče mnoge stvari koje su možda oduvek želeli da znaju. Prednost ovih simulatora je u tome što dolazi do ogromne uštede novca. Ukoliko bi neko u ovom slučaju želeo da nauči da isprogramira FMC, jedini način je da upiše pilotsku akademiju (koja je vrlo skupa) i vremenom nauči kako se on pravilno popunjava. Izradom simulatora i njegovim postavljanjem na Internet, osobe koje možda nemaju dovoljno novca za akademiju, mogu sebi priuštiti potpuno besplatno ili za malu količinu novca obuku na istom tom simulatoru. Pored toga što je jeftiniji, simulator je i dosta fleksibilniji, u smislu dozvoljavanja učenikima da naprave greške i na njima se nauče, što bi ih u pravom avionu moglo koštati života!

2 Računarska simulacija

Samo poreklo reči simulacija potiče od latinske reči „*simulatio*“, što po definiciji znači pretvaranje, zabušavanje, odnosno prikazivanje nekog pravog stanja drugačije nego što je u stvari. Pored tog značenja, simulacija predstavlja i metodu kojom se utvrđuje ponašanje sistema, modela i slično pro promenjenim uslovima.[1]

Američka komanda za obuku i simulacije (STRICOM) dala je jednu od najjednostavnijih definicija simulacije koja predstavlja i njihov moto: „Sve osim rata je simulacija“.

Simulacija predstavlja najbolji način prikaza ponašanja realnog sistema ili načina odvijanja realnog procesa, putem eksperimentisanja na modelu koji je napravljen tako da predstavlja prikaz realnog sistema ili situacije, obuhvatajući one aspekte realnosti (elemente i veze između njih) koji su od interesa za proučavanje. U današnje vreme, simulacioni eksperimenti i modeli se razvijaju i koriste uz pomoć računara.[2]

2.1 Modeliranje i simulacija

Modeliranje i simulacija predstavljaju složenu aktivnost koja se sastoji od tri elementa:

1. *Realni sistem* – Sistem koji predstavlja uređen, međuzavistan skup elemenata koji formiraju jednu celinu i deluju zajednički kako bi ostvarili zadati cilj. Realni sistem je izvor podataka, oblika $x(t)$, za specifikaciju modela.
2. *Model* - Apstraktni prikaz sistema, daje njegovu strukturu, komponente i njihovo uzajamno delovanje. U računarskoj tehnici, model predstavlja skup instrukcija (program) koje služe za generisanje ponašanja simuliranog sistema. Model ima svoje objekte koji su opisani atributima i promenljivama.
3. *Računar* – Uređaj koji služi za izvršavanje instrukcija modela koje generišu razvoj modela u vremenu na osnovu ulaznih podataka.

Modeliranje je proces kojim se uspostavlja veza između realnog sistema i modela. Odnosi se na validnost modela koja opisuje koliko verno model predstavlja simulacioni sistem (validacija).

Simulacija je proces koji uspostavlja vezu između modela i računara. Odnosi se na proveru da li simulacioni program verno prenosi model na računar i na tačnost kojom računar vrši instrukcije. Procena korektnosti simulatora zove se *verifikacija*. [3]

Simulacioni proces (dijagram toka simulacionog procesa)

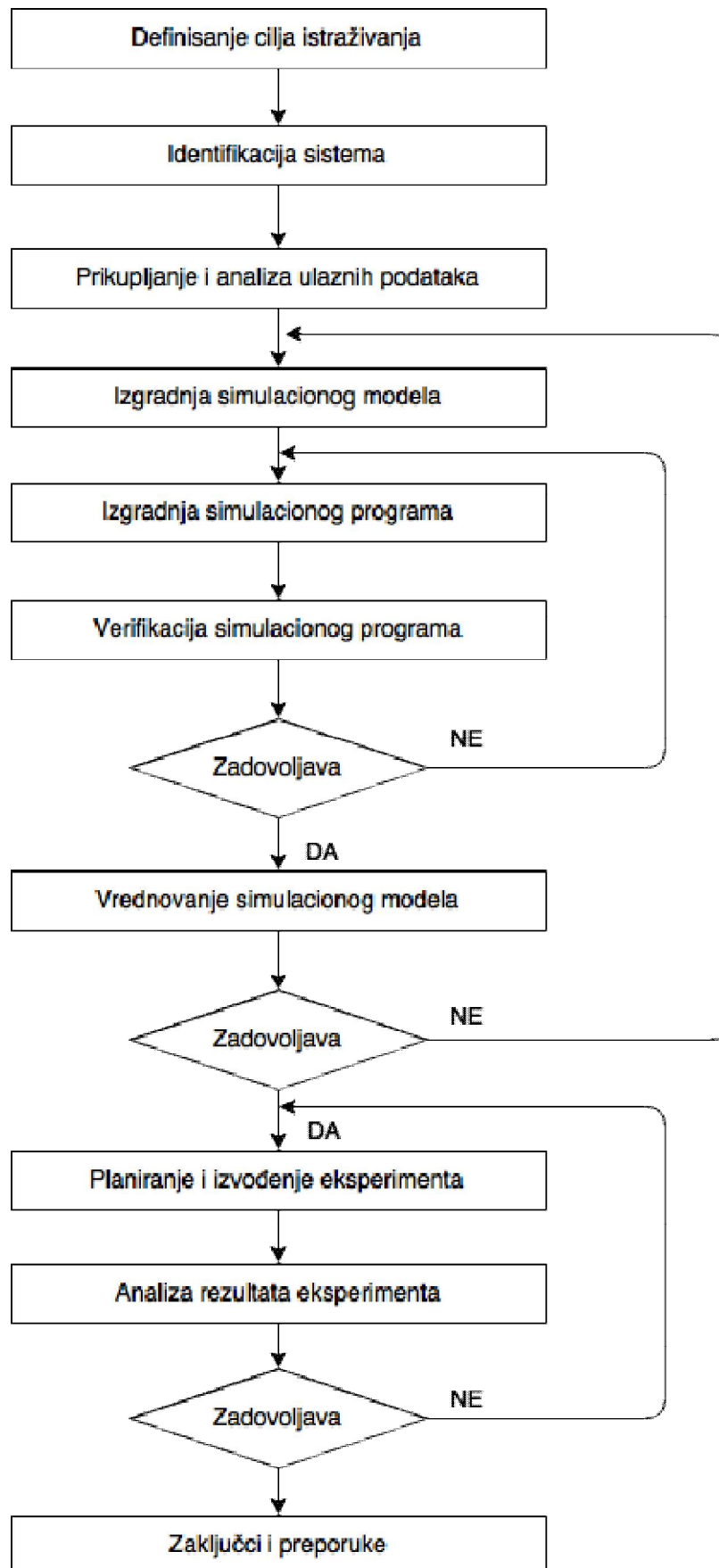
Simulacioni proces predstavlja strukturu rešavanja stvarnih problema pomoću simulacionog modeliranja. Ovakav tip procesa sastoji se iz nekoliko koraka i omogućava povratak na prethodni korak (nije sekvencijalan).

Simulacioni proces, sadrži ili podrazumeva nekoliko koraka. Sam izgled dijagrama toka simulacionog procesa je predstavljen na slici 1.

Koraci simulacionih procesa:

1. Definicija cilja istraživanja (definicija cilja simulacione studije);
2. Identifikacija sistema (opis komponenti sistema, način na koji sistem radi, kakva je veza sistema sa okolinom, prikaz sistema);
3. Prikupljanje i analiza ulaznih podataka (orikupljanje podataka o sistemu i njihova analiza);
4. Izgradnja simulacionog modela (konceptualni model koji opisuje sistem);
5. Izgradnja simulacionog programa (odabir adekvatnog programskog jezika i izrada simulacionog programa);
6. Verifikacija simulacionog programa (ovde se radi neophodna da li nam program verno prenosi model koji smo prethodno napravili i u koliko ga ne zadovoljava, vraća se na ponovnu izgradnju simulacionog programa, tačka 4.);
7. Vrednovanje simulacionog modela (vrednovanje da li model adekvatno predstavlja realni sistem);
8. Planiranje i izvođenje eksperimenta
9. Analiza rezultata eksperimenta (u većini slučajeva statička analiza)
10. Zaključci i preporuke

Tokom izrade simulacije FMC-a, praćeni su svi gore navedeni koraci kako bi se ispoštovala procedura i simulacija dovela na najviši nivo. U daljem tekstu, svaki korak će biti detaljno objašnjen: Od ideje do realizacije.



Slika 1 Dijagram toka simulacionog procesa

3 Tehnologije korišćene za izradu projekta

Pri izradi ovog diplomskog rada, korišćene su razne web tehnologije. Od programskih jezika korišćen je PHP koji je najviše došao do izražaja pri interakciji sa bazom podataka. Za UI deo zaduženi su HTML5, CSS i JavaScript.

3.1 PHP programski jezik

PHP (*Hypertext Preprocessor*) je programski jezik koji se prvi put pojavio 1995. godine i osmišljen je od strane Rasmus Lerdorfa, programera sa Grenlanda. PHP je osmišljen kao skriptni jezik čija se sintaksa zasnivala na sintaksi jezika C, Java i Perl sa nekoliko specifičnih mogućnosti koje su dodate pored već postojećih. Cilj ovog jezika je da omogući programerima brzo razvijanje dinamički generisanih stranica.

Ono što ovaj jezik u prvi mah izdvojilo od ostalih jeste njegova integracija sa HTML-om. Po prvi put je omogućeno nekom programskom jeziku da se izvršava direktno unutar HTML koda ili da se HTML izvršava unutar njega. Još jedna bitna stavka koja ga izdvaja od drugih klijentskih skriptnih jezika poput JavaScript-a je ta što kod izvršava na strani servera gde se generiše HTML koji se zatim šalje na klijentsku stranu. Klijent dobija rezultate izvršene skripte bez znanja kako je izgledao kod koji je to odradio. Moguće je čak i konfigurisati web server tako da procesuirava sav HTML preko PHP-a.

Lakoća učenja ovog programskog jezika je još jedna od njegovih prednosti. Uprkos tome što deluje lako, ima jako veliki broj naprednih opcija koje koriste profesionalni programeri.

Da je ovaj programski jezik popularan, svedoči činjenica da je do januara 2013. godine čak 240 miliona (39%) web sajtova koristilo PHP programski jezik kao glavni.

PHP se lako prepoznaje unutar svakog HTML dokumenta. Kod koji se nalazi unutar posebnih instrukcija za početak i kraj skripte je PHP kod. Instrukcija za početak je `<?php` a za kraj je `?>`. Sav kod koji se nalazi unutar ovih znakova je PHP kod.

```
<!DOCTYPE html>
<html>
<head>
  <title>Primer</title>
</head>
<body>

  <?php
    echo "Zdravo, ja sam PHP!";
  ?>

</body>
</html>
```

Trenutno aktuelna verzija PHP-a je 5.6. Od prve verzije (1.0) do sada, prešao je put dug 20 godina i za to vreme pretrpeo razne promene. Svaka verzija php-a izlazi sa podrškom u trajanju od nekoliko godina. Trenutna verzija se pojavila krajem avgusta 2014. godine i imaće podršku do avgusta 2017. godine.

Tokom verzija 3 i 4, PHP je dobio osnovnu funkcionalnost rada sa objektima koja je zatim u potpunosti prerađena u verziji 5. U ovoj verziji, poboljšana je i podrška za MySQL, dodate nove funkcije, system za prijavu grešaka, itd...

Osnovu objektno orijentisanog programiranja čini deklarisanje klasa i njihovo instanciranje u objekte[4]. Kao i u većini programskih jezika da bi se izvršila deklaracija klase, neophodno je korišćenje službene reči „*class*“. Svaka klasa može imati nekoliko atributa i metoda ukoliko je potrebno, a ti atributi i metode mogu biti privatne (*private*), javne (*public*) i zaštićene (*protected*). Ono što „*public*“ metode izdvaja od privatnih i zaštićenih je to što je njima moguće pristupiti i van objekta. Razlika između „*private*“ i „*protected*“ metoda je u tome što se „*protected*“ metode mogu naslediti.

Primer jedne PHP klase sa funkcijama i promenljivama:

```
<?php

classMyClass
{
    public$prop1= "I'm a class property!";

    publicfunctionsetProperty($newval)
    {
        $this->prop1 = $newval;
    }

    publicfunctiongetProperty()
    {
        return$this->prop1 . "<br />";
    }
}

$obj= newMyClass;

echo$obj->prop1;

?>
```

3.2 HTML

HTML je markup language, odnosno jezik za kreiranje i označavanje dokumenata na Internetu, nastao početkom devedesetih godina prošlog veka. Pored samog označavanja dokumenata, HTML se koristi i za kreiranje samog korisničkog interfejsa svake web aplikacije. Za razliku od PHP-a, HTML ne spada u grupu programskih jezika, već se isključivo može svrstati u grupu jezika za označavanje.

Suštinski gledano, HTML je najobičniji tekstualni fajl za čiju izradu može poslužiti bilo koji program za uređivanje teksta. Da bi se dokument smatrao validnim HTML-om, on mora poštovati određena pravila koja su usklađena standardima. Svaka verzija ima svoje propisane standarde pisanja koji se moraju poštovati. Aktuelna verzija HTML standarda je 5, ali s obzirom da se taj standard još uvek razvija, većina dokumenata na Internetu je rađeno uz korišćenje nekih starijih verzija.

HTML 5 se vrlo lako prepoznaje po svojoj deklaraciji `<!DOCTYPE html>`, dok starije verzije poseduju nekoliko varijanti deklarisanja zbog različitih standarda. Ovo je ujedno i prvi osnovni element svakog HTML dokumenta, dok je drugi element, sam html element

Unutar html elementa se nalazi sadržaj koji je podeljen na dva osnovna dela – zaglavlje i telo (head, body).

Sama struktura izgleda ovako:

```
<html>
  <head>
  ...
</head>
  <body>
  ...
</body>
</html>
```

Podaci o samom dokumentu poput naslova, opisa, ključnih reči i veze sa skriptama i stilovima se po pravilu skladišti u zaglavlju dokumenta (head), dok se sadržaj same stranice, odnosno kod koji se učitava i interpretira čuva u telu (body). Sav sadržaj se definiše odgovarajućim oznakama (tagovima) koje su predefinisane odgovarajućim standardom.

S obzirom na to da je osnovna namena HTML-a kada je inicijalno razvijan, bila kreiranje dokumenata i sadržaja koji je moguće povezati, HTML pruža širok spektar mogućnosti za formatiranje samog sadržaja. Svaki paragraf, tabela, lista ili bilo koji element HTML-a moguće je zasebno formatirati i urediti na način na koji korisnik to želi.

3.3 JavaScript

JavaScript potiče iz porodice skriptnih jezika uz mogućnost objektno orijentisanog pristupa. Iako na prvi pogled sintaksa i programske konstrukcije poput petlji i operatora ovog skriptnog jezika dosta podseća na C, C++ i Javu, ostatak i nije kako nam se čini. Važno je napomenuti da je JavaScript slabo tipiziran jezik, odnosno potrebno je definisati tip promenljive.

Kada je objektno orijentisano programiranje JavaScript-a u pitanju, mehanizam nasleđivanja je zasnovan na prototipovima. Ovakav mehanizam ne postoji u jezicima C, C++ i Java. Objekti mapiraju, odnosno preslikavaju imena svojstava u njihove proizvoljne vrednosti.

Kao i kod HTML-a i u JavaScript-u postoje delom određeni standardi, dok se određene stvari mogu raditi i van propisanih standarda. Važno je napomenuti da je JavaScript jezik koji se izvršava na klijentskoj strani. U većini slučajeva se koristi u samim pretraživačima i omogućava upravljanje istim, interakciju sa korisnikom, izvršavanje „*embedded*” (ugrađenih) prozora unutar pretraživača, i tako dalje.

Još jedna zanimljiva stvar vezana za JavaScript je da je ljudi često mešaju sa programskim jezikom Java. Iako sličan po sintaksi, uz mogućnost oba jezika da se izvršavaju unutar pretraživača i na sve to (delimično) zajedničko ime, Java i JavaScript nemaju nikakvu povezanost. Java je programski jezik koji se prevodi, dok je JavaScript skriptni jezik koji se interpretira.

Primer JavaScript-a koji klikom na dugme pokazuje datum i vreme:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Datum i vreme</title>
  </head>
  <body>
    <h1>JavaScript datum i vreme</h1>
    <button type="button"
      onclick="document.getElementById(datumVreme').innerHTML =
      Date()">
      Kliknite ovde kako biste videli datum i vreme.</button>
    <p id="datumVreme"></p>
  </body>
</html>
```

4 Baza podataka

4.1 MySQL

MySQL predstavlja jedan od najpopularnijih SQL database management Sistema otvorenog koda. Sama skraćenica SQL znači „Structured Query Language” i predstavlja najčešće korišćen standardizovan jezik kada je u pitanju rad sa bazama podataka. U zavisnosti od programskog okruženja, SQL se može koristiti direktno (npr. za generisanje izveštaja), ugrađene SQL naredbe unutar nekog drugog programskog jezika ili preko posebnih API-ja iza kojih se krije SQL sintaksa.

Ako bismo pokušali da sam pojam baze podataka objasnimo na najjednostavniji način, rekli bismo da je to struktuirana grupa podataka, koja može varirati od jednostavne poput galerije slika, do kompleksne poput ogromne baze podataka za neku korporativnu mrežu.

Bitna karakteristika MySQL baza podataka je da su *relacione*. To znači da se podaci unutar baze podataka čuvaju u nekoliko razdvojenih tabela umesto u jednu veliku. Razlog ovog razdvajanja je optimizacija brzine pretrage i pronalaska podataka unutar velike baze podataka. Ukoliko se radi o bazama manjeg obima, tipa 1000 podataka, upiti će se izvršavati dovoljno brzo i bez čekanja. Ukoliko se baza velikog obima, npr. glavna baza podataka berze u kojoj se količina podataka broji milionima, svaka milisekunda može značiti gubitak novca i clijenata. Ovaj način fizičkog razdvajanja podataka (objekti poput baza, tabela, kolona, redova, pogleda, itd.), ne samo da je optimizovan po pitanju brzine, već i omogućava fleksibilnije programiranje. Takođe bitna karakteristika relacionih baza podataka su i takozvane *veze* ili *relacije*. Postoji ukupno 3 tipa relacija:

1. Jedan prema jedan (one-to-one , 1-1);
2. Jedan prema više (one-to-many, 1-m);
3. Više prema više (many-to-many, m-n)

MySQL Database Software je klijent/server sistem koji se sastoji od *multi-threaded* SQL servera koji ima podršku za različite backend-ove, alate za administraciju, nekoliko različitih klijentskih programa i biblioteka i širok spektar API¹-ja

¹API – Application Programming Interface

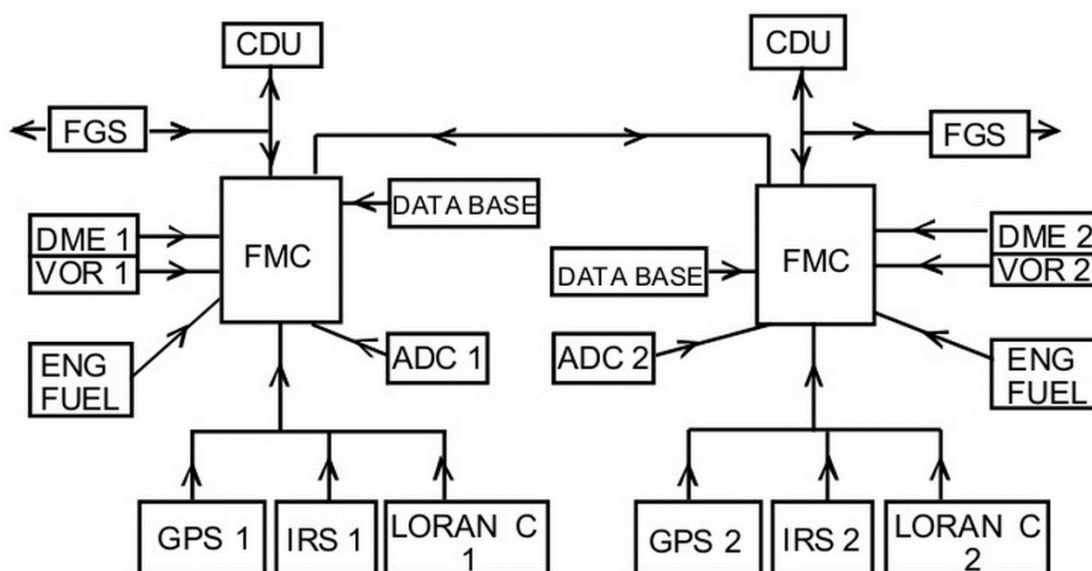
4.2 Flight Management System – Princip rada

Flight Management System (FMS) je osmišljen da bi se unapredila navigacija, povećala efikasnost potrošnje goriva i smanjio obim posla članovima posade. Ovaj sistem omogućava lakše vođenje vazduhoplova kroz kompleksne vazdušne puteve uz pomoć lateralne navigacije (LNAV).

Vertikalna navigacija (VNAV) omogućava da system izračuna optimalnu visinu leta vazduhoplova i izračuna najbolji odnos korišćenja automatske snage (Auto-Throttle) i brzine penjanja i spuštanja kako bi potrošnja goriva bila što manja.

Glavni cilj ovog sistema je prikupljanje podataka sa raznih radio navigacijskih uređaja i baze podataka, a zatim obrada pristiglih parametara radi kalkulacije što boljih performansi vazduhoplova u toku leta. Na ovaj način smanjuje se obim posla posade koja više ne mora da prati karte da bi navigacija bila uspešna i nestala je potreba za praćenjem (papirnih) uputstava za kontrolisano i bezbedno krstarenje (Cruise Control Manual). Na osnovu podataka koji su vezani za performanse motora, minimalne i maksimalne brzine, sve kalkulacije vezane za brzinu i potisak motora vrši FMC umesto posade.

Šematski prikaz rada ovog Sistema se nalazi slici ispod.



Slika 2 Šematski prikaz principa rada Sistema

Objašnjenja značenja skraćenica sa slike:

CDU – Control and Display Unit

FGS – Flight Guidance System

FMC – Flight Management Computer

ADC - Air Data Computer

IRS - Inertial Reference System

GPS – Global Positioning System

DME – Distance Measuring Equipment

VOR – VHF Omni-directional radio Range
LORAN – LOng RANGE Navigation

4.3 Model baze podataka

FMC poseduje tri različite baze podataka i svaka se čuva na posebnim EEPROM memorijskim karticama:

1. Software Options (OP PROGRAM) – Baza koja čuva sve izmene u različitim verzijama softvera.
2. Model/Engine database (MEDB) – Baza koja čuva podatke o modelu aviona i motora koje poseduje, kao i podatke o performansama za V^2 brzine, potrošnju goriva, mogućnost letenja na određenim visinama, i tako dalje.
3. Navigation database (NDB) – Baza koja čuva podatke o navigacionim tačkama, vazдушnim putevima, itd.. i validna je 28 dana.

U ovom radu, biće detaljno odrađena navigaciona baza podataka po uzoru na bazu koja se koristi i na pravom avionu.

Navigaciona baza se sastoji iz nekoliko delova:

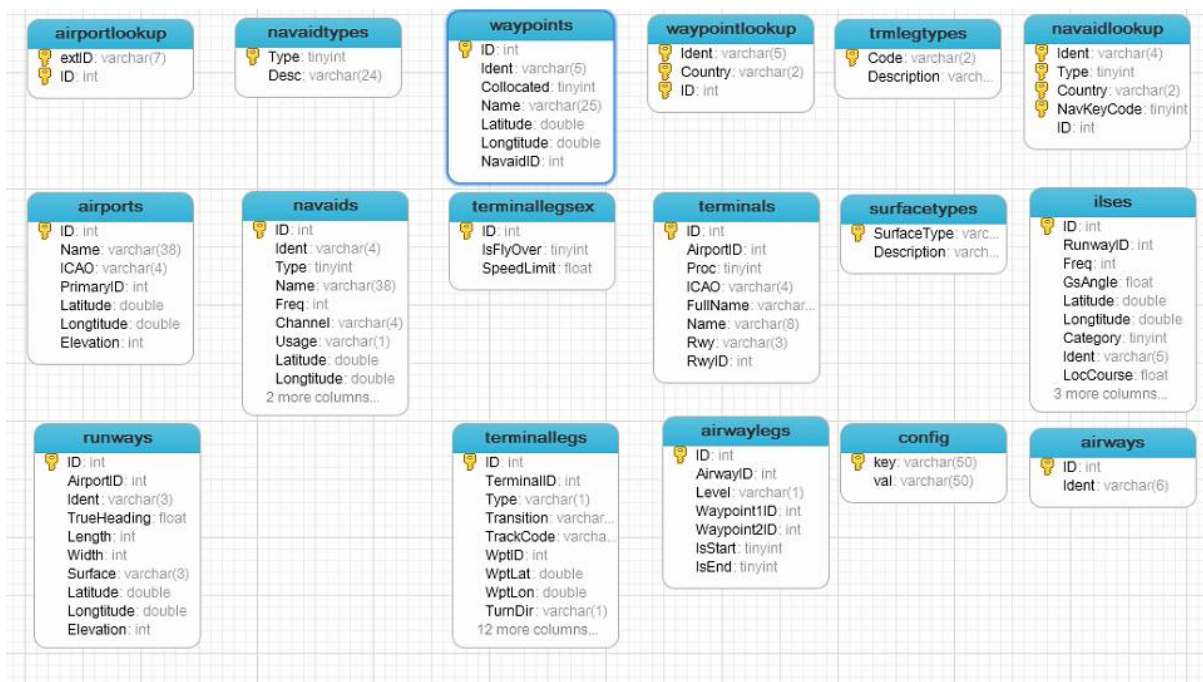
1. *Permanent*
2. *Supplemental (SUPP)*
3. *Temporary (REF)*

*Permanen*tni su oni delovi koje članovi posade (piloti) ne mogu promeniti. Tipičan primer permanentnih podataka unutar ove baze su podaci o poletno-sletim stazama. To su podaci poput identifikatora staze (piste), aerodroma za koji se ta staza vezuje (na kom se nalazi), geografska širina i dužina, elevacija staze u odnosu na nivo mora, dužina i širina staze izražena u metrima i stopama i magnetna varijacija.

SUPP i *REF* tipovi baze koja pruža mogućnost članovima posade da unesu neke privremene podatke koji će se tu skladištiti. Mogu primiti do 40 tačaka u vazдушnom prostoru (*waypoints*), 40 navigacionih uređaja koji se nalaze na zemlji (*nav aids*) i 6 aerodroma (*airports*). *SUPP* podaci se mogu uneti samo tokom faze boravka na zemlji, dok je u toku faze leta ova opcija onemogućena. Svi *REF* podaci se automatski brišu nakon završetka leta.

Na slici 3 predstavljena je šema navigacione baze koja je urađena u grafičkom interfejsu za realizaciju tabela i baza podataka Navicat i koja se sastoji od 17 tabela.

²Postoji nekoliko tipova V brzina. Koriste se za označavanje minimalne i maksimalne brzine kojom vazduhoplov može bezbedno leteti, poletne i sletne brzine, itd...



Slika 3 Model navigacione baze podataka

Svaka od gore navedenih tabela se može predstaviti i tekstualno po njihovim strukturama sa primarnim ključevima označenim podebljanim slovima:

Airportlookup(**extID**, **ID**)

Airports(**ID**, Name, ICAO, PrimaryID, Latitude, Longitude, Elevation)

Airwaylegs(**ID**, AirwayID, Level, Waypoint1ID, Waypoint2ID, IsStart, IsEnd)

Airways(**ID**, ident)

Config(**Key**, val)

Ilses (**ID**, RunwayID, Freq, GsAngle, Latitude, Longitude, Category, Ident, LocCourse, CrossingHeight, HasDme, Elevation)

Navaidlookup (**Ident**, **Type**, **Country**, **NavKeyCode**, **ID**)

Navairs (**ID**, Ident, Type, Name, Freq, Channel, Usage, Latitude, Longitude, Elevation, SlavedVar)

Navaidtypes (**Type**, Desc)

Runways (**ID**, AirportID, Ident, TrueHeading, Length, Width, Surface, Latitude, Longitude, Elevation)

Surfacetypes (**SurfaceType**, Description)

Terminallegs (**ID**, TerminalID, Type, Transition, TrackCode, WptID, WptLat, WptLon, TurnDir, NavID, NavLat, NavLon, NavBear, NavDist, Course, Distance, Alt, Vnav, CenterID, CenterLat, CenterLon)

Terminallegsex (**ID**, IsFlyOver, SpeedLimit)

Terminals (**ID**, AirportID, Proc, ICAO, FullName, Rwy, RwyID)

Trmlegtypes (**Code**, Description)

Waypointlookup (**Ident, Country, ID**)

Waypoints (**ID**, Ident, Collocated, Name, Latitude, Longitude, NavaidID)

4.4 Realizacija i objašnjenje tabela

Svaka tabela korišćena za simulaciju u ovom diplomskom radu, urađena je po uzoru na realne tabele kakvi su one zaista u stvarnosti. U nastavku, biće izdvojeno 5 najvažnijih tabela i biće objašnjena njihova uloga i važnost. Tabele koje će biti objašnjene su:

- Airports
- Ilses
- Nav aids
- Runways
- Terminals
- Waypoints
- Airways

4.4.1 Tabela Airports

Tabela Airports predstavlja prvu u nizu najbitnijih tabela i u njoj su smešteni podaci o aerodromima. Ova tabela ima 7 polja za unost 7 različitih podataka. Polja u koja se unose podaci su:

1. ID tipa int(11)
2. Name tipa varchar(38) u koje se upisuje naziv aerodroma
3. ICAO³ tipa varchar(4) u koje se upisuje jedinstveni kod aerodroma
4. PrimaryID tipa int(11)u koje se upisuje ID ukoliko se nazivi aerodroma poklapaju
5. Latitude tipa double u koje se upisuje geografska širina aerodroma
6. Longitude tipa double u koje se upisuje geografska dužina aerodroma
7. Elevation tipa int(11) u koje se upisuje elevacija aerodroma u odnosu na nivo mora

#	Name	Type	Collation	Attributes	Null	Default
☐ 1	ID	int(11)			No	0
☐ 2	Name	varchar(38)	utf8_general_ci		Yes	NULL
☐ 3	ICAO	varchar(4)	utf8_general_ci		Yes	NULL
☐ 4	PrimaryID	int(11)			Yes	0
☐ 5	Latitude	double			Yes	0
☐ 6	Longitude	double			Yes	0
☐ 7	Elevation	int(11)			Yes	0

Slika 4 Izgled tabele Airports

³ICAO – Definisan od strane International Civil Aviation Organization i predstavlja alfanumerički kod sastavljen od 4 karaktera u cilju izdvajanja i prepoznavanja aerodroma u svetu

ID	Name	ICAO	PrimaryID	Latitude	Longitude	Elevation
1	LAVA HOT SPRINGS	01ID	NULL	42.60825	-112.032461	5268
2	GRANVILLE	01NK	NULL	43.425	-73.261944	421
3	PREHN CRANBERRY COMPANY	01WI	NULL	44.01	-90.389167	930
4	CAWLEYS SOUTH PRAIRIE	02WA	NULL	47.152325	-122.094003	690
5	STILLWATER CREEK	04WN	NULL	46.958359	-119.63882	1510
6	GWINN	06FA	NULL	26.908153	-80.328414	28
7	WESTERN	06WN	NULL	46.924986	-122.553039	394
8	OCEAN REEF CLUB	07FA	NULL	25.325392	-80.274775	8
9	GLASGOW INDUSTRIAL	07MT	NULL	48.421108	-106.527703	2762
10	ALTA SIERRA	09CL	NULL	39.113783	-121.056614	2275
11	LBJ RANCH	0TE7	NULL	30.251814	-98.622531	1515
12	SEVEN CS RANCH	0XA4	NULL	27.997111	-98.882861	336
13	74 RANCH	0XA5	NULL	28.685383	-98.383117	317
14	MACK MESA	10CO	NULL	39.268111	-108.864028	4724
15	THOMPSON CREEK OILFIELD	11WY	NULL	44.958606	-104.868656	3814
16	NEWTON FIELD	12CL	NULL	36.156897	-119.849578	194
17	HAINS	12LS	NULL	30.178611	-92.285	20
18	WESTPORT	12NK	NULL	44.158386	-73.432906	276
19	RING HILL	14ME	NULL	44.79257	-69.070322	340
20	WYATT	14TS	NULL	27.421667	-98.604361	537
21	BREDLOW FARM	17AR	NULL	34.533428	-92.100144	233
22	JUMBOLAIR	17FL	NULL	29.280253	-82.124814	100
23	RE-DUN	17NK	NULL	42.456945	-76.950834	1351
24	QUAHADI RANCH	17XS	NULL	32.177764	-98.43505	1422
25	SKY RANCH AT CAREFREE	18AZ	NULL	33.818095	-111.897925	2568

Slika 5 Tabela popunjena podacima

Ubacivanje podataka u tabelu se vrši SQL komandom INSERT na sledeći način:

```
INSERT INTO `fmc`.`airports` (`ID`, `Name`, `ICAO`, `PrimaryID`, `Latitude`,
`Longitude`, `Elevation`) VALUES ('1', 'Test', 'Test', '0', '42.1777760', '-91.865305', '350');
```

4.4.2 Tabela Ilses

Tabela Ilses je druga tabela koja će biti objašnjena i u njoj su smešteni podaci o posebnoj vrsti navigacije za instrumentalni prilaz aerodromu i automatsko sletanje. Ova tabela ima ukupno 12 polja za unos podataka. Polja u koja se unose podaci su:

1. ID int(11)
2. RunwayID int(11) u koje se upisuje ID poletno sletne staze
3. Freq int(11) u koje se upisuje frekvencija ILS⁴-a
4. GsAngle float u koje se upisuje ugao GS⁵-a
5. Latitude double - u koje se upisuje geografska širina aerodroma
6. Longitude double - u koje se upisuje geografska dužina aerodroma
7. Category – tinyint(4) u koje se upisuje kategorija ILS-a
8. Ident varchar(5) – u koje se upisuje identifikator ILS-a
9. LocCourse float – u koje se upisuje kurs localizer⁶-a
10. CrossingHeight tinyint(4) – u koje se upisuje visina na kojoj se prelazi preko localizer antene izražena u stopama
11. HasDme tinyint(1) – u koje se upisuje da li poseduje DME⁷
12. Elevation int(11) – u koje se upisuje elevacija aerodroma, odnosno visina aerodroma u odnosu na nivo mora, takođe izražena u stopama

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	0
2	RunwayID	int(11)			Yes	0
3	Freq	int(11)			Yes	0
4	GsAngle	float			Yes	0
5	Latitude	double			Yes	0
6	Longitude	double			Yes	0
7	Category	tinyint(4)			Yes	0
8	Ident	varchar(5)	utf8_general_ci		Yes	NULL
9	LocCourse	float			Yes	0
10	CrossingHeight	tinyint(4)			Yes	0
11	HasDme	tinyint(1)			Yes	0
12	Elevation	int(11)			Yes	0

Slika 6 Izgled tabele Ilses

⁴ILS – Instrument Landing System predstavlja sistem za automatsko navođenje vazduhoplova na poletno sletnu stazu uz pomoć radio signala čijim se praćenjem letelica pozicionira tačno na sredinu staze i dobija odgovarajući vertikalni profil za bezbedno spuštanje do praga poletno sletne staze.

⁵GS –Glide Slope je posebna vrsta stanice sa antenom koja se nalazi pored „touchdown zone“ (dela staze predviđenom kao tačka dodira letelice i staze) i emituje signal koji se šalje do letelice sa tačnim uglom spuštanja do tačke sletanja.

⁶Localizer – Slično kao i Glide Slope, s tim što se antena nalazi ispred poletno sletne staze i na sredini i na taj način omogućava precizno navođenje vazduhoplova na sredinu poletno sletne staze.

⁷DME – Distance Measuring Equipment je uređaj koji šalje radio signal do vazduhoplova koji se nalazi u prilazu i meri njegovu udaljenost od praga staze.

ID	RunwayID	Freq	GsAngle	Latitude	Longitude	Category	Ident	LocCourse	CrossingHeight	HasDme	Elevation
1	17	17321984	3	48.410503	-106.500147	1	IGSG	106	56	1	2773
2	18	17920000	3	48.431706	-106.555328	1	IGBE	286	55	1	2764
3	435	17829888	3	-9.458922	147.231225	1	IWVG	142	50	1	126
4	438	17387520	3	-9.427633	147.2119	1	IBB	322	50	1	126
5	475	17305600	3.2	-6.080111	142.880675	1	IXM	162	50	1	5056
6	484	17838080	3	64.188178	-51.678261	0	GN	225	50	1	0
7	499	17388800	3	67.018506	-50.682072	0	ISF	94	50	1	0
8	503	17387520	3	76.532645	-68.632995	1	ITL	85	51	0	212
9	509	17928192	3	65.631445	-18.011194	0	IEY	38	50	1	0
10	509	17338368	5.3	65.669695	-18.07125	1	IAL	13	83	1	7
11	523	17379328	3	65.2925	-14.391139	1	IES	37	53	1	106
12	537	17903616	3	63.993334	-22.605445	1	IKN	15	51	1	200
13	538	17387520	3	63.985056	-22.58375	2	IKF	106	55	1	100
14	539	17838080	3	63.963139	-22.605445	2	IKO	196	50	1	200
15	540	17321984	3	63.985028	-22.661722	1	IKW	285	55	1	200
16	555	17371136	3	64.126944	-21.929556	0	IRE	132	50	1	0
17	556	17403904	3.5	64.120694	-21.935945	1	IRK	191	53	1	26
18	589	17829888	3	42.558611	21.037222	2	PRS	173	49	1	1788
19	1007	17903616	3	64.509789	-110.267228	0	IDK	99	50	1	0
20	1008	17903616	3	64.512906	-110.312208	0	IDV	279	50	1	0
21	1234	17846272	3	57.288297	-110.976667	1	ISR	345	45	1	1760
22	2432	17387520	3	46.48	-84.494167	1	IAM	115	42	0	610
23	2463	17838080	2.5	49.134467	-68.187361	1	IBC	98	46	0	70
24	2471	17831168	3	48.331667	-70.971667	1	ITV	109	52	0	514
25	2473	17911808	3	48.332317	-71.021547	1	IBG	289	51	0	511

Slika 7 Tabela popunjena podacima

Ubacivanje podataka u tabelu se vrši SQL komandom INSERT na sledeći način:

```
INSERT INTO `fmc`.`ilses` (`ID`, `RunwayID`, `Freq`, `GsAngle`, `Latitude`,
`Longitude`, `Category`, `Ident`, `LocCourse`, `CrossingHeight`, `HasDme`, `Elevation`)
VALUES ('1', '1', '12345678', '3', '42.1777760', '-91.865305', '1', 'KNK', '113', '50', '1', '7');
```

Ova tabela je od velikog značaja jer sadrži veoma bitne podatke koji će omogućiti bezbedno sletanje vazduhoplova. Jednostavnim unosom identifikatora, izlistać se bitni podaci poput frekvencije ILS-a, kursa lokalajzera, aktiviraće se DME (ukoliko postoji) i prikazaće se ugao spuštanja.

4.4.3 Tabela Navaid

Tabela Navaid je treća tabela koja će biti objašnjena i u njoj su smešteni podaci o navigacionim tačkama za navođenje vazduhoplova. Ova tabela ima ukupno 11 polja za unos podataka. Polja u koja se unose podaci su:

1. ID int(11)
2. Ident varchar(4) - u koje se upisuje identifikator
3. Type tinyint(4) - u koje se upisuje vrsta navigacione stanice
4. Name varchar(38) - u koje se upisuje ime navigacione stanice
5. Freq int(11) - u koje se upisuje radna frekvencija
6. Channel varchar(4) - u koje se upisuje frekvencija posebnog kanala ukoliko postoji
7. Usage varchar(1) - u koje se upisuje tip korišćenja stanice
8. Latitude double - u koje se upisuje geografska širina aerodroma
9. Longitude double - u koje se upisuje geografska dužina aerodroma
10. Elevation int(11) - u koje se upisuje elevacija navigacione stanice
11. SlavedVar float – u koje se upisuje varijacija ukoliko je ima

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	0
2	Ident	varchar(4)	utf8_general_ci		Yes	NULL
3	Type	tinyint(4)			Yes	0
4	Name	varchar(38)	utf8_general_ci		Yes	NULL
5	Freq	int(11)			Yes	NULL
6	Channel	varchar(4)	utf8_general_ci		Yes	NULL
7	Usage	varchar(1)	utf8_general_ci		Yes	NULL
8	Latitude	double			Yes	0
9	Longitude	double			Yes	0
10	Elevation	int(11)			Yes	0
11	SlavedVar	float			Yes	0

Slika 8 Izgled tabele Navaid

Važnost ove tabele ogleda se u tome što je sadrži podatke o navigacionim stanicama koje se nalaze na zemlji i koje se danas sve manje koriste zbog dolaska novih tehnologija, ali još uvek imaju značajan udeo u navigaciji. Kada govorimo o navigacionim stanicama, mislimo na VOR⁸ i NDB⁹ koji su nekada bili jedini način navigacije kroz vazduh.

⁸VOR - VHF Omni-directional Radio range je navigaciona stanica koja se nalazi na zemlji emituje signal u svim pravcima omogućavajući pritom lakše snalaženje vazduhoplova i određivanje pozicije

⁹NDB – Non-Directional radio Beacon je radio transmitter na zemlji koji šalje signal vazduhoplovu ali na manjoj frekvenciji i signal ima manji domet.

ID	Ident	Type	Name	Freq	Channel	Usage	Latitude	Longitude	Elevation	SlavedVar
1	ITA	3	TERRA NOVA BAY	17387520	NULL	H	-74.691436	164.103947	0	0
2	HN	4	HONIARA	18046976	NULL	H	-9.435111	160.037306	32	0
3	NI	4	NAURU I.	18022400	NULL	H	-0.543889	166.920834	14	0
4	BUK	9	BUKA	18116608	NULL	H	-5.443011	154.667164	20	0
5	GBE	9	GOBE	18046976	NULL	H	-6.890325	143.679103	181	0
6	GNV	4	GURNEY	18178048	NULL	H	-10.315828	150.361739	65	0
7	GUA	9	GIRUA	18239488	NULL	H	-8.744936	148.253678	300	0
8	HKN	9	HOSKINS	18087936	NULL	H	-5.458097	150.414353	66	0
9	KIU	9	KIUNGA	18055168	NULL	H	-6.1216	141.287092	86	0
10	KVG	4	KAVIENG	18288640	NULL	H	-2.585517	150.813967	23	0
11	MD	4	MADANG	18231296	NULL	H	-5.211642	145.776775	11	0
12	MH	9	MT. HAGEN	18153472	NULL	H	-5.837875	144.308531	5350	0
13	MOE	9	MOMOTE	18042880	NULL	H	-2.061425	147.426261	30	0
14	MOR	9	MORO	18112512	NULL	H	-6.362872	143.254336	2800	0
15	NZ	4	NADZAB	18059264	NULL	H	-6.568339	146.708947	230	0
16	OKT	9	OK TEDI	18120704	NULL	H	-5.4099	141.318697	3000	0
17	PY	4	PORT MORESBY	18284544	NULL	H	-9.452961	147.215092	525	0
18	TOK	4	TOKUA	18046976	NULL	H	-4.346525	152.397606	32	0
19	VNO	9	VANIMO	18178048	NULL	H	-2.6789	141.303111	10	0
20	WK	4	WEWAK	18243584	NULL	H	-3.583514	143.675953	19	0
21	YXM	4	KOMO	18051072	NULL	H	-6.055497	142.872792	5158	0
22	AS	9	AASIAAT	17331456	NULL	T	68.71942	-52.792753	74	0
23	JA	9	ILULISSAT	17929472	NULL	T	69.240286	-51.060811	95	0
24	MS	9	MANIITSOQ	17847552	NULL	T	65.411636	-52.930039	91	0
25	NOR	9	STATION NORD	17920000	NULL	B	81.598278	-16.665333	80	0

Slika 9 Tabela popunjena podacima

Ubacivanje podataka u tabelu se vrši SQL komandom INSERT na sledeći način:

```
INSERT INTO `fmc`.`navaids` (`ID`, `Ident`, `Type`, `Name`, `Freq`, `Channel`,
`Usage`, `Latitude`, `Longitude`, `Elevation`, `SlavedVar`) VALUES (1, 'TEST', 2, 'Test',
'12345678', NULL, NULL, '-2.585517', 152.397606, '15', '0');
```

4.4.4 Tabela Runways

Tabela Navaid je četvrta tabela koja će biti objašnjena i u njoj su smešteni podaci o svim poletno sletnim stazama koje postoje u svetu. Ova tabela ima ukupno 10 polja za unos podataka. Polja u koja se unose podaci su:

1. ID int(11)
2. AirportID (int(11) koja predstavlja spoljni ključ ka tabeli Airports
3. Ident varchar(3) - u koje se unosi identifikator poletno sletne staze
4. TrueHeading float - u koje se unosi pravac prostiranja staze
5. Length int(11) - u koje se unosi dužina poletno sletne staze
6. Width int(11) - u koje se unosi širina poletno sletne staze
7. Surface varchar(3) - u koje se unosi tip podloge staze (beton, asfalt, trava)
8. Latitude double - u koje se upisuje geografska širina
9. Longitude double - u koje se upisuje geografska dužina
10. Elevation int(11) – u koje se unosi elevacija staze

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	0
2	AirportID	int(11)			Yes	0
3	Ident	varchar(3)	utf8_general_ci		Yes	NULL
4	TrueHeading	float			Yes	0
5	Length	int(11)			Yes	0
6	Width	int(11)			Yes	0
7	Surface	varchar(3)	utf8_general_ci		Yes	NULL
8	Latitude	double			Yes	0
9	Longitude	double			Yes	0
10	Elevation	int(11)			Yes	0

Slika 10 Izgled tabele Runways

Važnost ove tabele se ogleda u tome što sadrži podatke o poletno sletnim stazama za svaki aerodrom. Podaci poput identifikatora koji se obeležava brojevima ili kombinacija brojeva i slova L za levu stazu, C za centralnu stazu i R za desnu stazu bitni su, jer se koriste pri identifikaciji staze pri poletanju/sletanju (poput oznaka za autoput). Svaka staza ima svoju tačnu dužinu i širinu koja pomaže pri kalkulaciji performansi za poletanje/sletanje vazduhoplova, kao i tip staze koji takođe ima direktan uticaj na performanse. Elevacija je još jedan jako bitan detalj koji takođe utiče na proračun performansi pri poletanju i sletanju, jer se motori različito ponašaju na različitim nadmorskim visinama.

ID	AirportID	Ident	TrueHeading	Length	Width	Surface	Latitude	Longitude	Elevation
1	1	14	156	2894	100	GRE	42.611	-112.034867	5268
2	1	32	336	2894	100	GRE	42.603733	-112.030533	5268
3	2	16	144.9	2500	50	ASP	43.425	-73.261944	421
4	2	34	324.9	2500	50	ASP	43.425	-73.261944	421
5	3	01	7.6	2600	120	GRE	0	0	930
6	3	19	187.6	2600	120	GRE	0	0	930
7	4	16	177.1	2650	50	GRE	0	0	690
8	4	34	357.1	2650	50	GRE	0	0	690
9	5	11	126.6	4100	60	ASP	46.958359	-119.63882	1510
10	5	29	306.6	4100	60	ASP	46.958359	-119.63882	1510
11	6	09	88.3	7003	150	ASP	26.908153	-80.33842	28
12	6	27	268.4	7003	150	ASP	26.908156	-80.318408	28
13	7	09	107.3	2845	30	ASP	46.925831	-122.556397	394
14	7	27	287.3	2845	30	ASP	46.923697	-122.547939	394
15	8	04	33.8	4500	70	ASP	0	0	8
16	8	22	213.8	4500	70	ASP	0	0	8
17	9	10	117	13500	300	ASP	48.430342	-106.551786	2757
18	9	28	297	13500	300	ASP	48.415275	-106.512558	2756
19	10	01	24.6	3000	50	ASP	0	0	2275
20	10	19	204.6	3000	50	ASP	0	0	2275
21	11	17	175.3	6291	60	ASP	30.260439	-98.623375	1515
22	11	35	355.3	6291	60	ASP	30.243203	-98.621703	1433
23	12	15	153.5	5005	75	ASP	28.003258	-98.886356	335
24	12	33	333.5	5005	75	ASP	27.99097	-98.879367	336
25	13	14	144.1	3887	50	ASP	28.689733	-98.386642	317

Slika 11 Tabela popunjena podacima

Ubacivanje podataka u tabelu se vrši SQL komandom INSERT na sledeći način:

```
INSERT INTO `fmc`.`runways` (`ID`, `AirportID`, `Ident`, `TrueHeading`, `Length`,
`Width`, `Surface`, `Latitude`, `Longitude`, `Elevation`) VALUES ('1', '1', '12', '118', '8450',
'250', 'ASP', '27.99097', '-98.879367', '113');
```

4.4.5 Tabela Terminals

Tabela Terminals je peta tabela koja će biti objašnjena i u njoj su smešteni podaci o svim odlaznim i dolaznim vazdušnim putanjama aerodroma. Ova tabela ima ukupno 8 polja za unos podataka. Polja u koja se unose podaci su:

1. ID int(11)
2. AirportID int(11) – koja predstavlja spoljni ključ ka tabeli Airports
3. Proc tinyint(4) – u koje se označava tip procedure
4. ICAO varchar(4) - u koje se upisuje jedinstveni kod aerodroma
5. FullName varchar(28) – u koje se upisuje pun naziv procedure
6. Name varchar(8) – u koje se upisuje skraćeni naziv procedure
7. Rwy varchar(3) – u koje se upisuje identifikator poletno sletne staze za koju se procedura vezuje
8. RwyID int(11) – koje predstavlja spoljni ključ ka tabeli Runways

#	Name	Type	Collation	Attributes	Null	Default
1	<u>ID</u>	int(11)			No	0
2	AirportID	int(11)			Yes	0
3	Proc	tinyint(4)			Yes	0
4	ICAO	varchar(4)	utf8_general_ci		No	None
5	FullName	varchar(28)	utf8_general_ci		No	None
6	Name	varchar(8)	utf8_general_ci		No	None
7	Rwy	varchar(3)	utf8_general_ci		Yes	NULL
8	RwyID	int(11)			Yes	0

Slika 12Izgled tabele Terminals

Značaj ove tabele se ogleda u tome što se u njoj nalaze imena SID¹⁰ i STAR¹¹ procedura, odnosno odlaznih i dolaznih putanja od/do aerodroma. Uvođenjem ove opcije, znatno je olakšano letenje odlaznih i dolaznih procedura i omogućeno je autopilotu da potpuno automatizovano isprati celu proceduru. Ovim je takođe smanjen obim posla posade i letenje je daleko preciznije. Sve dolazne i odlazne procedure se nalaze iscrtane na kartama i nekada su se letele „na ruke“, bez mogućnosti autopilota da sve sam isprati, uključujući ograničenje brzine ili visine na određenim tačkama i slično. Danas je sve to mnogo lakše zahvaljujući ovoj opciji i odlazne i dolazne putanje su detaljno iscrtane na instrumentima, tako da je i potreba za konstantnim praćenjem karata smanjena.

¹⁰SID – Standard Instrumental Departure je standardna navigaciona procedura koju vazduhoplov prati od trenutka poletanja do „en-route“ faze

¹¹ STAR- Standard Terminal Arrival Route je standardna navigaciona procedura koju vazduhoplov prati od kraja „en-route“ faze do inicijalne tačke prilaza aerodromu

ID	AirportID	Proc	ICAO	FullName	Name	Rwy	RwyID
1	131	2	AGGH	AKNA1A	AKNA1A	06	279
2	131	2	AGGH	AKNA1B	AKNA1B	24	280
3	131	2	AGGH	AUKI1A	AUKI1A	06	279
4	131	2	AGGH	AUKI1B	AUKI1B	24	280
5	131	2	AGGH	DUGO1A	DUGO1A	06	279
6	131	2	AGGH	DUGO1B	DUGO1B	24	280
7	131	2	AGGH	ENOU1A	ENOU1A	06	279
8	131	2	AGGH	ENOU1B	ENOU1B	24	280
9	131	2	AGGH	FERA1A	FERA1A	06	279
10	131	2	AGGH	FERA1B	FERA1B	24	280
11	131	2	AGGH	GIZO1A	GIZO1A	06	279
12	131	2	AGGH	GIZO1B	GIZO1B	24	280
13	131	2	AGGH	KAGL1A	KAGL1A	06	279
14	131	2	AGGH	KAGL1B	KAGL1B	24	280
15	131	2	AGGH	KAKO1A	KAKO1A	06	279
16	131	2	AGGH	KAKO1B	KAKO1B	24	280
17	131	2	AGGH	KIRA1A	KIRA1A	06	279
18	131	2	AGGH	KIRA1B	KIRA1B	24	280
19	131	2	AGGH	KOPA1A	KOPA1A	06	279
20	131	2	AGGH	KOPA1B	KOPA1B	24	280
21	131	2	AGGH	LEGO1A	LEGO1A	06	279
22	131	2	AGGH	LEGO1B	LEGO1B	24	280
23	131	2	AGGH	MANO1A	MANO1A	06	279
24	131	2	AGGH	MANO1B	MANO1B	24	280
25	131	2	AGGH	NIPE1A	NIPE1A	06	279

Slika 13 Tabela popunjena podacima

Ubacivanje podataka u tabelu se vrši SQL komandom INSERT na sledeći način:

```
INSERT INTO `fmc`.`terminals` ('ID', 'AirportID', 'Proc', 'ICAO', 'FullName',
`Name`, 'Rwy', 'RwyID') VALUES ('1', '1', '2', 'TEST', 'TestProcedure', 'TEST3A', '12',
'150');
```

4.4.6 Tabela Waypoints

U tabeli waypoints su smešteni podaci o svim navigacionim tačkama u vazдушnom prostoru. Ova tabela ima ukupno 8 polja za unost podataka. Polja u koja se unose podaci su:

1. ID int(11)
2. Ident varchar(5) – u koje se unosi identifikator
3. Collocated tinyint(1) – u koje se upisuje da li tačka učestvuje u kolokaciji sa nekom drugom ili ne
4. Name varchar(25) – u koje se upisuje naziv tačke
5. Latitude double – u koje se upisuje geografska širina
6. Longitude double - u koje se upisuje geografska dužina
7. NavaidID int(11) – spoljni ključ

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	0
2	Ident	varchar(5)	utf8_general_ci		No	None
3	Collocated	tinyint(1)			Yes	0
4	Name	varchar(25)	utf8_general_ci		Yes	NULL
5	Latitude	double			Yes	0
6	Longitude	double			Yes	0
7	NavaidID	int(11)			Yes	0

Slika 14 Izgled tabele Waypoints

Važnost ove tabele je u tome se u tome što se u njoj nalaze podaci ključni za navigaciju vazduhoplova. Navigacija se pored navigacionih stanica koje emituju radio signal, obavlja i uz pomoć imaginarnih tačaka u vazдушnom prostoru. Svaka tačka ima svoje tačne koordinate i svoj naziv i ne postoje dve tačke sa istim imenom¹². Tačke se grupišu i stvaraju vazdušne puteve koji predstavljaju koridore po kojima se kreću vazduhoplovi, nešto poput puteva na zemlji.

Identifikator je standardizovan i može imati najviše pet karaktera, dok ime nema nikakvih ograničenja i često se poklapaju. Jednostavim ukucavanjem identifikatora u FMC, navigacija je daleko jednostavnija i lakša u odnosu na navigaciju pre dvadesetak godina.

¹²Za razliku od tačaka koje su jedinstvene, imena navigacionih se mogu poklapati, ali ne i njihova frekvencija rada i koordinate

ID	Ident	Collocated	Name	Latitude	Longitude	NavaidID
1	05S60	0	05S160E	-5	160	NULL
2	06S60	0	06S160E	-6	160	NULL
3	07S60	0	07S160E	-7	160	NULL
4	08S60	0	08S160E	-8	160	NULL
5	09S60	0	09S160E	-9	160	NULL
6	10S60	0	10S160E	-10	160	NULL
7	11S60	0	11S160E	-11	160	NULL
8	12S60	0	12S160E	-12	160	NULL
9	13S60	0	13S160E	-13	160	NULL
10	14S60	0	14S160E	-14	160	NULL
11	20HNG	0	HN224120	-10.7	158.433333	NULL
12	20HNI	0	HN225120	-9.655556	158	NULL
13	ABVUB	0	ABVUB	-8.650411	160.668664	NULL
14	ADNUG	0	ADNUG	-10.955945	160.744783	NULL
15	AGGA	0	AUKI	-8.702642	160.681697	NULL
16	AGGC	0	CHOISEUL	-6.711986	156.396195	NULL
17	AGGE	0	BALLALAE	-6.990717	155.885772	NULL
18	AGGF	0	FERA	-8.107417	159.577114	NULL
19	AGGK	0	KIRAKIRA	-10.44927	161.898097	NULL
20	AGGM	0	MUNDA	-8.326931	157.263133	NULL
21	AGGR	0	RENNELL	-11.550514	160.063361	NULL
22	AGGS	0	SEGHE	-8.578428	157.87567	NULL
23	AGOXI	0	AGOXI	-10.254086	158.563958	NULL
24	AGRUK	0	AGRUK	-9.9374	160.043333	NULL
25	AKNAK	0	AKNAK	-13.22	164.393333	NULL

Slika 15 Tabela popunjena podacima

Ubacivanje podataka u tabelu se vrši SQL komandom INSERT na sledeći način:

```
INSERT INTO `fmc`.`waypoints` (`ID`, `Ident`, `Collocated`, `Name`, `Latitude`,
`Longitude`, `NavaidID`) VALUES (0, 'TEST', 0, 'Test', 0, 0, 0);
```

4.4.7 Tabela Airways

Podaci o svim vazdušnim putevima, sastavni su deo tabele Airways. Ona ima svega 2 polja za unos podataka. Polja u koja se unose podaci su:

1. ID int(11)
2. Ident varchar(6) – u koje se unosi identifikator vazdušnog puta

#	Name	Type	Collation	Attributes	Null	Default
1	<u>ID</u>	int(11)			No	0
2	Ident	varchar(6)	utf8_general_ci		No	None

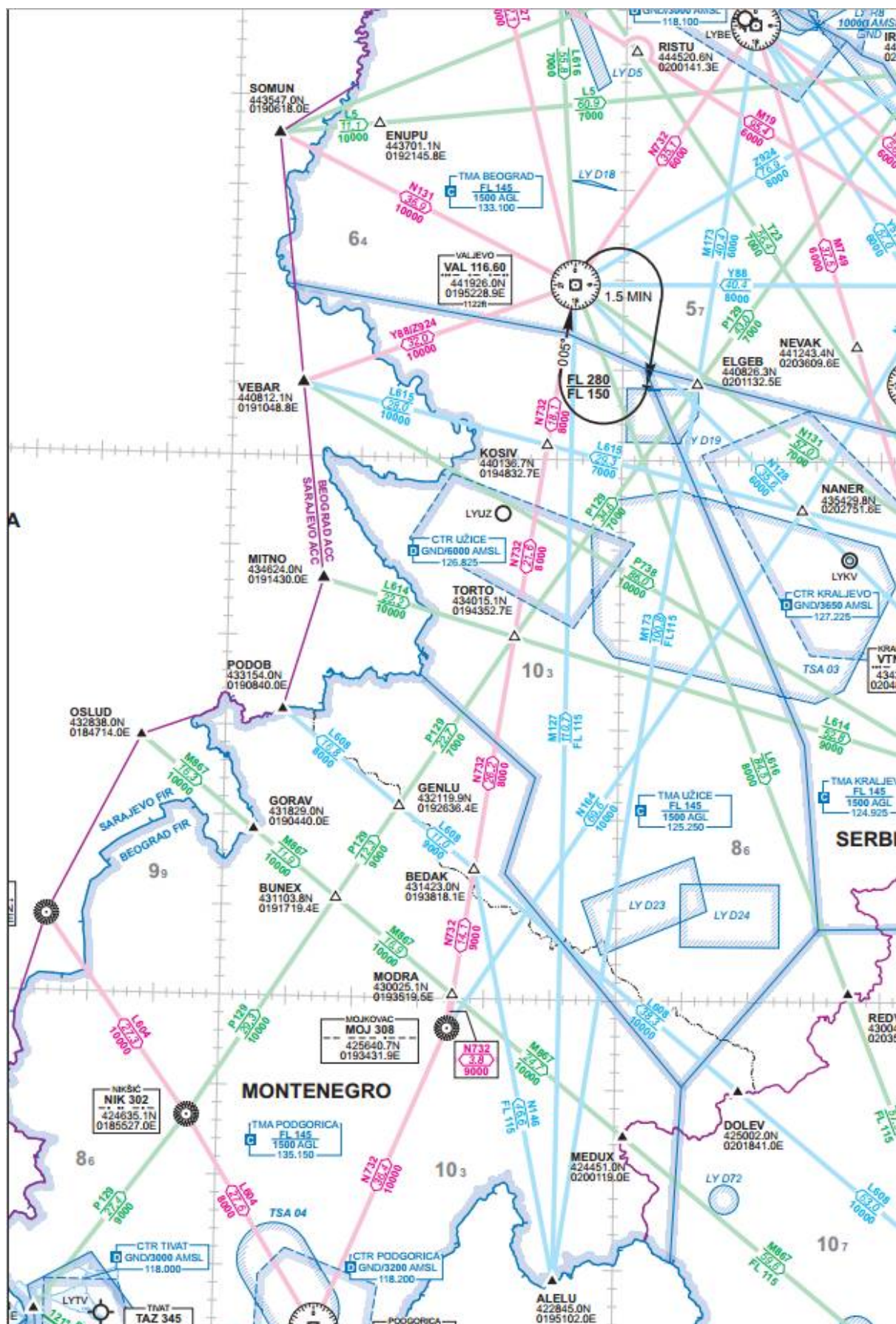
Slika 16 Izgled tabele Airways

Tabela vazdušnih puteva je prilično jednostavna. Sadrži svega dva polja od kojih je jedno ID, a drugo identifikator vazdušnog puta. Kao što je već napomenuto, vazdušni putevi nastaju spajanjem dve ili više tačaka i označavaju se uz pomoć identifikatora. Ideja vazdušnih puteva nastala je u isto vreme kada i ideja o navigacionim tačkama (waypoints). Svaka zemlja, odnosno njen vazdušni prostor, ima svoje vazdušne puteve. U poslednje vreme, određene zemlje praktikuju takozvani „*free route airspace*“, odnosno u određenom delu dana (uglavnom u periodu između 22h i 4h) ne koriste vazdušne puteve kao koridore, već posada vazduhoplova dobija prečice (uglavnom izlazne tačke iz zemlje) i na taj način smanjuju potrošnju goriva i ispuštanje emisijonih gasova u atmosferu. Postoje dva tipa vazdušnih puteva: Lower Airways (za avione koji lete na manjim visinama) i Upper Airways (za avione koji lete na većim visinama).

Vazdušni putevi su velika olakšica i prilikom popunjavanja plana leta, odnosno njegovog unošenja u FMC. Uzećemo kao primer plan leta između Beograda i Podgorice.

Plan leta bi izgledao ovako:

Nakon poletanja, prva tačka bi bila VAL, odnosno Valjevo VOR. Nakon VAL, uzimamo vazdušni put N732 do MOJ, odnosno Mojkovac NDB-a i nakon toga na Podgoricu. Ruta sa vazdušnim putem bi glasila *VAL N732 MOJ*. Ukoliko malo bolje pogledamo sliku 17 i vazdušni put N732 između Valjeva i Mojkovca, videćemo da se nalazi još nekoliko navigacionih tačaka. Ukoliko ne bismo koristili vazdušni put, morali bismo da unesemo sve tačke koje se nalaze između VAL i MOJ i u tom slučaju, naš plan leta bi izgledao ovako: *VAL KOSIV TORTO BEDAK MODRA MOJ*. Korišćenje vazdušnih puteva nam skraćuje vreme i minimalizuje grešku za pogrešan unos neke od vazdušnih tačaka. Obe opcije su ispravne, ali se stvar komplikuje što je ruta duža i uvek se preporučuje unošenje vazdušnih puteva.



ID	Ident
1	A1
2	A10
3	A100
4	A102
5	A103
6	A104
7	A105
8	A106
9	A107
10	A108
11	A109
12	A11
13	A110
14	A111
15	A112
16	A113
17	A114
18	A115
19	A116
20	A117
21	A118
22	A119
23	A120
24	A121
25	A122

Slika 18 Tabela popunjena podacima

Ova tabela ne funkcioniše sama. Kao što se može videti, tabela ima samo ID i identifikator, ali ne i tačke koje sačinjavaju te vazdušne puteve. Airwayslegs je druga tabela koja je vezana sa tabelom airways i koja preko AirwayID-a zna kom vazdušnom putu pripadaju tačke koje se nalaze u drugoj tabeli.

Ubacivanje podataka u tabelu se vrši SQL komandom INSERT na sledeći način:

```
INSERT INTO `fmc`.`airways` (`ID`, `Ident`) VALUES ('1', 'TEST');
```

5 Programska realizacija

5.1 Strana IDENT

Prva strana koja se otvara nakon paljenja FMC-a je IDENT strana. IDENT predstavlja skraćenicu od Identification i na njoj se nalaze bitne informacije o vazduhoplovu, navigacionoj bazi i motorima. Izgled pomenute strane, nalaza na slici 19.



Slika 19 Izgled strane IDENT

Kao što je već nagovešteno, ovde se nalaze relevantni podaci o modelu vazduhoplova, što možemo videti u gornjem levom uglu. 737-800WL znači da je u pitanju Boeing 737 serije 800 sa wingletima¹³.

Desno od modela nalazi se podatak o jačini motora (ENG RATING), odnosno njegovom potisku izražena u američkim funtama (lbs) i u našem slučaju je to motor koji proizvodi 26000 lbs potiska, ili 26K kako se to skraćeno označava.

Odmah ispod ENG RATING sekcije, nalaze se podaci o navigacionoj bazi podataka. Ovo je treća bitna stvar koju piloti proveravaju prilikom spremanja aviona. Da li je to model u kom se nalaze, da li su to ti motori i da li je navigaciona baza aktivna ili je istekla. Kao što se može videti na slici, na displeju je ispisana validnost navigacione baze. Veoma je bitno proveriti da li se period validnosti aktivne baze poklapa sa trenutnim datumom. Baza je validna 28 dana i nakon toga se ubacuje nova. Kako postoji veliki broj aerodroma, vazdušnih puteva, navigacionih tačaka, itd... neretko se dešava da se izvrše neke promene procedura. Ovo je glavni razlog zbog kog se baza obnavlja na svakih 28 dana. Naša baza je validna od 10. maja do 15. juna 2015. godine. Primetićete da se ispod naše aktivne baze (MAY10JUN15/15) nalazi i još jedna baza. To je sekundarna baza koja je zapravo starija mesec dana i može se aktivirati ukoliko nastanu neki problemi sa primarnom bazom, što se vrlo, vrlo retko dešava, ali backup opcija je uvek dostupna.

Poslednji red displeja je uvek rezervisan za navigaciju po ostalim stranama. U slučaju IDENT strane, imamo 2 opcije. Da odemo na INDEX stranu ili na POS INIT. O njima će biti reči nešto kasnije.

5.1.1 Programska realizacija strane

Svaka strana izrađena je na principu templejta (engl. *Template*). Sam izgled uređaja je zapravo fotografija koja je izmapirana sa tačnim koordinatama svakog dugmeta i njegovim ID-em. Funkcionalnost svakog dugmeta radi JavaScript, dok je izgled samih strana (displeja) najobičniji tekst ubačen u .page fajl čiji se sadržaj čita u vidu skripte.

Primećuje se da se u gotovo svakom .js fajlu nalaze određene *callback* funkcije. Njihova uloga je u većini slučajeva otvaranje drugih strana. Tako se na IDENT strani nalazi callback funkcija koja otvara INIT REF stranu i POSITION INIT.

```
1 function callback_IDENT() {  
2  
3     unbindAll();  
4     $(l6).on('click', function(){  
5         openPage('SCREEN_INIT_REF', "callback_SCREEN_INIT_REF");  
6     });  
7  
8     $(r6).on('click', function(){  
9         openPage('POSITION_INIT', "callback_POSITION_INIT");  
10    });  
11 }
```

Slika 20 JavaScript kod za otvaranje strana klikom na određeno dugme

Na slici 21 prikazan je kod koji radi mapiranje dugmića na slici i dodeljuje im ID

¹³Winglet je mali produžetak krila koji smanjuje protok vazduha preko same ivice krila i na taj način povećava efikasnost krila i povećava potisak.

```

24 
25 <map name="mapa" id="mapa">
26   <area id="0" shape="rect" coords="92,575,134,606" class="dugme">
27   <area id="1" shape="rect" coords="47,461,89,492" class="dugme">
28   <area id="2" shape="rect" coords="92,461,134,492" class="dugme">
29   <area id="3" shape="rect" coords="137,462,179,493" class="dugme">
30   <area id="4" shape="rect" coords="48,501,90,532" class="dugme">
31   <area id="5" shape="rect" coords="93,499,135,530" class="dugme">
32   <area id="6" shape="rect" coords="136,499,178,530" class="dugme">
33   <area id="7" shape="rect" coords="46,538,88,569" class="dugme">
34   <area id="8" shape="rect" coords="93,537,135,568" class="dugme">
35   <area id="9" shape="rect" coords="136,537,178,568" class="dugme">
36   <area id="a" shape="rect" coords="186,387,228,418" class="dugme">
37   <area id="b" shape="rect" coords="229,386,271,417" class="dugme">
38   <area id="c" shape="rect" coords="273,386,315,417" class="dugme">
39   <area id="d" shape="rect" coords="316,386,358,417" class="dugme">
40   <area id="e" shape="rect" coords="360,385,402,416" class="dugme">
41   <area id="f" shape="rect" coords="186,422,228,453" class="dugme">
42   <area id="g" shape="rect" coords="229,423,271,454" class="dugme">
43   <area id="h" shape="rect" coords="273,423,315,454" class="dugme">
44   <area id="i" shape="rect" coords="316,424,358,455" class="dugme">
45   <area id="j" shape="rect" coords="362,425,404,456" class="dugme">
46   <area id="k" shape="rect" coords="186,463,228,494" class="dugme">
47   <area id="l" shape="rect" coords="229,464,271,495" class="dugme">
48   <area id="m" shape="rect" coords="274,462,316,493" class="dugme">
49   <area id="n" shape="rect" coords="315,462,357,493" class="dugme">
50   <area id="o" shape="rect" coords="360,463,402,494" class="dugme">
51   <area id="p" shape="rect" coords="185,499,227,530" class="dugme">
52   <area id="q" shape="rect" coords="229,500,271,531" class="dugme">
53   <area id="r" shape="rect" coords="272,499,314,530" class="dugme">
54   <area id="s" shape="rect" coords="317,500,359,531" class="dugme">
55   <area id="t" shape="rect" coords="360,499,402,530" class="dugme">
56   <area id="u" shape="rect" coords="186,537,228,568" class="dugme">
57   <area id="v" shape="rect" coords="229,537,271,568" class="dugme">
58   <area id="w" shape="rect" coords="274,539,316,570" class="dugme">
59   <area id="x" shape="rect" coords="314,536,356,567" class="dugme">
60   <area id="y" shape="rect" coords="359,537,401,568" class="dugme">
61   <area id="z" shape="rect" coords="186,575,228,606" class="dugme">
62   <area id="11" shape="rect" coords="7,68,45,96" class="dugme">
63   <area id="12" shape="rect" coords="6,101,44,129" class="dugme">
64   <area id="13" shape="rect" coords="6,132,44,160" class="dugme">
65   <area id="14" shape="rect" coords="5,163,43,191" class="dugme">
66   <area id="15" shape="rect" coords="7,194,45,222" class="dugme">
67   <area id="16" shape="rect" coords="4,225,42,253" class="dugme">
68   <area id="r1" shape="rect" coords="408,70,446,98" class="dugme">
69   <area id="r2" shape="rect" coords="408,101,446,129" class="dugme">
70   <area id="r3" shape="rect" coords="409,132,447,160" class="dugme">
71   <area id="r4" shape="rect" coords="408,165,446,192" class="dugme">
72   <area id="r5" shape="rect" coords="408,194,446,221" class="dugme">
73   <area id="r6" shape="rect" coords="408,226,446,253" class="dugme">
74   <area id="pm" shape="rect" coords="135,574,177,605" class="dugme">
75   <area id="sp" shape="rect" coords="228,574,270,605" class="dugme">
76   <area id="del" shape="rect" coords="271,577,313,608" class="dugme">
77   <area id="slash" shape="rect" coords="318,575,360,606" class="dugme">
78   <area id="clr" shape="rect" coords="360,573,402,604" class="dugme">
79   <area id="init" shape="rect" coords="53,306,104,337" class="dugme">
80   <area id="rte" shape="rect" coords="109,307,160,338" class="dugme">
81   <area id="clb" shape="rect" coords="165,307,216,338" class="dugme">
82   <area id="crr" shape="rect" coords="220,307,271,338" class="dugme">
83   <area id="des" shape="rect" coords="277,308,328,339" class="dugme">
84   <area id="menu" shape="rect" coords="51,344,102,375" class="dugme">
85   <area id="legs" shape="rect" coords="107,344,158,375" class="dugme">
86   <area id="dep" shape="rect" coords="164,344,215,375" class="dugme">
87   <area id="hold" shape="rect" coords="218,345,269,376" class="dugme">
88   <area id="prog" shape="rect" coords="276,344,327,375" class="dugme">
89   <area id="nl" shape="rect" coords="54,382,105,413" class="dugme">
90   <area id="fik" shape="rect" coords="109,382,160,413" class="dugme">
91   <area id="pp" shape="rect" coords="55,422,106,453" class="dugme">
92   <area id="ap" shape="rect" coords="110,422,161,453" class="dugme">
93   <area id="exec" shape="rect" coords="349,350,400,381" class="dugme">
94   <area id="tacka" shape="rect" coords="48,575,90,606" class="dugme">
95 </map>

```

Slika 21 Mapiranje dugmića

Za izgled svake strane zadužen je template čiji se sadržaj menja u zavisnosti funkcionalnosti koju svaka strana obavlja. Strana uvek ima 13 redova i svaka neparna linija ima za nijansu veći font, po uzoru na pravi FMC.

Na slici 22 prikazan je kod ident.page strane:

1	IDENT	1/2
2	MODEL	ENG RATING
3	737-800WL	26K
4	NAV DATA	ACTIVE
5	EZY1050401	MAY10JUN15/15
6		
7		APR15MAY09/15
8	OP PROGRAM	
9	549849-014	(U10.5A)
10		SUPP DATA
11		
12	-----	
13	<INDEX	POS INIT>

Slika 22 IDENT PAGE

S obzirom da je IDENT strana čisto informativnog karaktera, nema mnogo koda koji radi proračune, rad sa bazom podataka i slično. Klik na šesto dugme sa leve strane (L6) nas void na INDEX stranu koju ćemo sledeću objasniti.



Slika 23 Izgled IDENT strane na pravom uređaju

5.2 INIT/REF INDEX strana



Slika 24 INIT/REF INDEX strana

Još jedna prilično statična strana, bez mnogo koda. Cilj ove INDEX strane je da izlista prečice do neke od najbitnijih strana i odradi redirekciju do istih.

Te bitne strane su:

- IDENT – strana o kojoj je bilo reči na prethodnim stranama
- POSition – strana na kojoj se upisuje trenutna pozicija aviona
- PERFormance – strana na kojoj se unose podaci o težinama aviona, i tako dalje
- TAKEOFF – strana na kojoj se unose podaci relevantni za poletanje
- APPROACH – strana na kojoj se unose podaci relevantni za sletanje
- OFFSET – strana na kojoj se po potrebi unosi offset od trenutne putanje

5.2.1 Programska realizacija strane

```

1 function callback_SCREEN_INIT_REF() {
2     unbindAll();
3
4     $(11).on('click', function(){
5         openPage('IDENT', 'callback_IDENT');
6     });
7     $(12).on('click', function(){
8         openPage('POSITION_INIT', "callback_POSITION_INIT");
9     });
10    $(13).on('click', function(){
11        openPage('PERF_INIT', "callback_PERF_INIT");
12    });
13    $(14).on('click', function(){
14        openPage('TAKEOFF', "callback_TAKEOFF");
15    });
16    $(15).on('click', function(){
17        openPage('APPROACH', "callback_APPROACH");
18    });
19 }

```

Slika 25 SCREEN_INIT_REF.js kod

Kao što je već rečeno, ova stranica samo izlistava prečice do određenih strana. Stoga, postoji samo jedna callback funkcija koja otvara strane klikom na dugmiće sa leve strane.

```
1  &nbsp; &nbsp; &nbsp; &nbsp; INIT/REF INDEX 1/1
2
3  &lt;IDENT NAV DATA&gt;
4
5  &lt;POS
6
7  &lt;PERF
8
9  &lt;TAKEOFF
10
11 &lt;APPROACH
12
13 &lt;OFFSET NAV STATUS&gt;
```

Slika 26 SCREEN_INIT_REF.page

5.3 POSITION INIT strana

Sledeća u nizu je Position Init strana. Position init označava stranu na kojoj se vrši inicijalizacija pozicije aviona, te stoga i skraćenica POS INIT. Na ovoj strani potrebno je uneti nekoliko podataka i proveriti ih.

Počevši sa leve strane, prva opcija koja nam se nudi je REF AIRPORT, što je skraćenica od Reference Airport. Odmah ispod toga se nalaze četiri crtice koje nam ukazuju na to da je na njihovo mesto potrebno uneti ICAO kod aerodroma na kom se nalazi avion (referentni aerodrom). Ispod toga imamo GATE i crtice za upisivanje geja na kom se avion nalazi radi tačnih koordinata. U gornjem desnom uglu nalazi se LAST POS, odnosno koordinate poslednje pozicije na kojoj se avion nalazio, a odmah ispod njega i prostor koji pokazuje koordinate aerodroma koji je unet kao referentni. Ispod toga, SET IRS POS, ili mesto za upisivanje koordinata kako bi se IRS¹⁴ inicijalizovao i kako bi preciznost navigacije bila na visokom nivou. Ispod toga, sa leve strane, nalazi se tačno vreme u UTC (GMT) formatu, kao i tačan mesec i godina.

¹⁴IRS – Inertial Referent System



Slika 27 POS INIT strana

5.3.1 Programaska realizacija strane

Za razliku od prethodne dve strane koje nisu bile toliko bogate kodom, ova je opozit. Nekoliko funkcija, rad sa bazom podataka i *timestamp* su neophodni kako bi strana funkcionisala kako treba. Za početak, razmotrićemo kod vezan za dobijanje trenutnog vremena i datuma. Takođe, biće pokazana i 2 načina pisanja koda bez refaktorisanja i sa refaktorisanjem.

Primer inicijalizacije i setovanje tačnog vremena i datuma:

```
var timer = setInterval('if(typeof TIME!="undefined")setVariable(TIME,
(new Date()).toUTCString().substr(16,8).replace(":", "").replace(":", ".")), 1000);
var date = setInterval("if(typeof DATE!='undefined')setVariable(DATE,
(new Date()).getMonth()+1+'/'+'((((new Date()).getFullYear()+1900)+").substr(2, 2))",
1000);
```

Prva linija, koristi funkciju JavaScript funkciju setInterval kako bi izvukla tačno vreme i pretvorila ga u oblik koji je nama potreban.

Druga linija, koristi funkciju JavaScript funkciju setInterval kako bi izvukla tačan datum i pretvorila ga u oblik koji je nama potreban.

Obe funkcije rade dobro svoj posao, ali su malo teže za čitanje, kao i *debugging* ukoliko dođe do nekih problema kao što ih je u ovom primeru bilo sa neispravnim pokazivanjem datuma, te je stoga odrađen refactoring koda za dobijanje datuma.

Napravljena je posebna funkcija getDateString() koja sada glasi ovako:

```
function getDateString(){
    var month = new Date().getMonth() + 1;
    var year = (((new Date()).getFullYear()+1900).substr(2, 2));
    if(typeof DATE!='undefined')
        setVariable(DATE,(month + '/' + year))
}
```

A zatim je napravljena promenljiva *date* koja izvršava funkciju setInterval kojoj se prosleđuje prethodno kreirana funkcija getDateString() i vreme osvežavanja u trajanju od 1000ms, odnosno 1s.

```
var date = setInterval (getString() , 1000);
```

Na ovaj način, dobili smo čistiji i jasniji kod koji je mnogo lakše debugovati i pokazali značaj refaktorisanja koda.

Na ovoj strani već vidimo male izmene u vidu promenljivih koje se nalaze između uspravnih crta koji predstavljaju separatore i izdvajaju ih od ostatka statičnog teksta.

```

1      POS INIT      1/1
2      LAST POS
3      N44°49.2 E020°18.4
4      REF AIRPORT
5      |APT|      |COORDINATES|
6      GATE
7      |GATE|
8      SET IRS POS
9      |IRS|
10     GMT - MON/DY
11     |TIME|z |DATE|
12     -----
13     &lt;INDEX      ROUTE&gt;

```

Slika 28 POSITION_INIT.page

Naredna bitna stavka koja se izdvaja je unos referentnog aerodroma i proveru da li se unet ICAO kod nalazi u bazi, odnosno da li aerodrom postoji ili je došlo do greške prilikom unosa. Za ovu proveru, neophodan nam je rad sa navigacionom bazom.

Svaka komunikacija sa bazom podataka obavlja se preko posebno kreiranih API-ja koji se sastoji od najobičnijeg SQL koji nam omogućava dobijanje željenih podataka i izbacuje određeni statusni kod greške kako bismo lakše utvrdili da li je problem u interakciji sa bazom ili negde drugde u kodu. API koji nam izvlači podatke o aerodromu se zove „apt_provider.php“ i nalazi se u api folderu, zajedno sa ostalim provajderima.

```

1  <?php
2
3  @include "../DB.php";
4
5  $apt = preg_replace('/[^A-Z]/', '', $_GET["apt"]);
6
7  $SQL = "SELECT * FROM airports where ICAO = '{$apt}'";
8
9  $res = mysql_query ($SQL);
10
11  header("Content-type: text/json; charset=utf-8");
12
13  if($res and mysql_num_rows($res)){
14      $row = mysql_fetch_assoc($res);
15      echo json_encode ($row);
16      die();
17  }else{
18      echo json_encode(array("error"=>"- 1001"));
19      die();
20  }
21

```

Slika 29 apt_provider.php

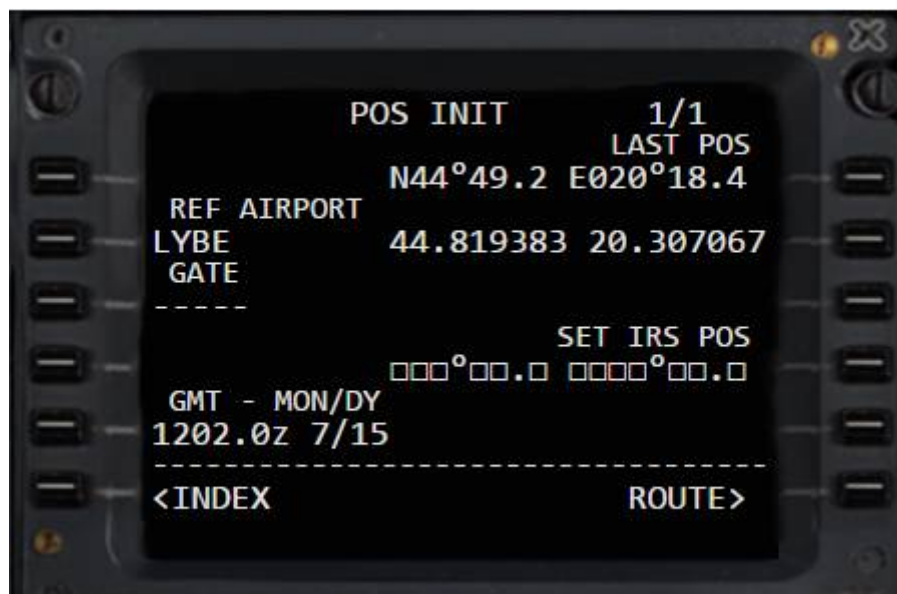
Ovaj SQL upit vraća JSON podatke koji se kasnije izvlače po potrebi. Na slici 30 nalazi se kod funkcije koja uzima podatke iz JSON-a, radi proveru da li se unet ICAO kod nalazi u bazi i vadi njegove koordinate koje postavlja u odgovarajuću labelu. Ukoliko se uneti kod ne nalazi u bazi, ispisaće se poruka da aerodrom ne postoji i biće omogućen ponovni unos.

```

56 function POSITION_INIT_L2(){
57     var input = $('#labela').html().trim();
58     $.getJSON("api/apt_provider.php?apt=" + encodeURIComponent(input), {}, function(data){
59         console.log(data);
60         if( data.ICAO == input ){
61             $('#APT').html(data.ICAO);
62             $('#COORDINATES').html(+data.Latitude + " " + data.Longitude);
63             localStorage.setItem('POSITION_INIT_L2_data', JSON.stringify(data));
64             $('#labela').html("");
65         }else{
66             $('#APT').html('---');
67             $('#COORDINATES').html('&square;&square;&square;°&square;&square;.&square; &square;&square;&square;&square;°&square;&square;.&square;');
68             $('#labela').html('AIRPORT DOES NOT EXIST');
69             localStorage.removeItem('POSITION_INIT_L2_data');
70         }
71     });
72 }
73

```

Unos odgovarajućeg ICAO koda aerodroma (kao i svih drugih podataka) se unosi preko alfanumeričkih karaktera na samom uređaju. Uzećemo primer aerodroma Beograd, čiji kod glasi LYBE i nakon unosa, kliknućemo na drugo dugme sa leve strane, odmah pored četiri crtice ispod natpisa REF AIRPORT. Ukoliko je sve uneto kako treba, stranica bi trebala da izgleda kao na slici 31.



Slika 31 Strana ukoliko je tačno unet kod referentnog aerodroma

Ukoliko je došlo do greške prilikom unosa koda, nakon klika na L2 dugme za unos referentnog aerodroma, pojaviće se greša da aerodrom ne postoji kao na slici 32.



Slika 32 Greška pri pogrešnom unosu koda

Nakon uspešnog postavljanja referentnog aerodroma, potrebno je otići na stranu ROUTE na kojoj se unose podaci o odlaznom i dolaznom aerodromu, broj leta i staza za poletanje.

5.4 ROUTE strana

Proces postaje složeniji. Strana ROUTE očekuje prosleđivanje bitnih podataka za let. Podaci koje ova strana zahteva da se unesu su:

- ORIGIN – odlazni aerodrom
- DEST – dolazni aerodrom (*destination*)
- RUNWAY – poletno sletna staza koja će se koristiti za poletanje
- FLT NO. – broj leta (flight number)
- CO ROUTE – kompanijska ruta (company route) je neobavezno polje u koje se može upisati već sačuvan plan leta koji će automatski učitati rutu između polaznog i dolaznog aerodroma.

Početna strana izgleda kao na slici 33 i sadrži prazna polja koja očekuju podatke.



Slika 33 ROUTE strana

5.4.1 Programska realizacija strane

Template za ovu stranu prikazan je na slici 34 i ne razlikuje se mnogo od prethodnih. Ukupan broj polja dostupnih za izmenu na ovoj strani je četiri, od kojih su 3 povezani sa bazom podataka i rade proveru da li uneta vrednost već postoji ili je došlo do greške prilikom unosa.

Ova strana spada u grupu obaveznih a popunjavanje jer bez unetih podataka o odlaznom i dolaznom aerodromu, avion neće znati svoju početnu i krajnju tačku.

1	RTE	1/2
2	ORIGIN	DEST
3	DEP	ARR
4	CO ROUTE	FLT NO.
5	COROUTE	FLTNO
6	RUNWAY	
7	RUNWAY	
8	-----	
9		
10		
11		
12		
13	<ERASE	ACCEPT>

Slika 34 RTE_ACCEPT.page

Kao i na prethodnoj strani i ovde je neophodno proveriti unete podatke da li se slažu sa podacima u bazi i onda izvršiti njihov upis. S obzirom da se ovde pored aerodroma izvlače i identifikatori poletno sletnih staza, neophodan je još jedan API. Ovaj API zove se `rwyr_provider.php` i izvršava SQL upit koji izvlači identifikator za uneti aerodrom.

Ukoliko se unese broj poletno sletne staze koja odgovara aerodromu, upit će se izvršiti bez problema i identifikator će biti sačuvan. U suprotnom, na ekranu se pojavljuje poruka da uneta staza ne postoji i mora se uneti odgovarajuća.

```

1  <?php
2  @include "../DB.php";
3
4  $dep = preg_replace('/^[^A-Z]/', '', $_GET["dep"]);
5  $ide = preg_replace('/^[^0-9A-Z]/', '', $_GET["ide"]);
6  $SQL = "SELECT ident from runways where AirportID = (select id from airports where ICAO = '{$dep}') AND Ident = '{$ide}'";
7  $res = mysql_query($SQL);
8
9  header("Content-type: text/json; charset=utf-8");
10 if ($res and mysql_num_rows($res)){
11     $row = mysql_fetch_assoc($res);
12     echo json_encode($row);
13     die();
14 }else{
15     echo json_encode(array("error"=>"-2001"));
16     die();
17 }

```

Slika 35 `rwyr_provider.php`

Popunjena strana za naš demo let bi izgledala kao na slici 36. Odlazni aerodrom je Beograd (LYBE), dolazni je Tivat (LYTV), broj leta ASL72L i poletna staza 12.

Polja za unos odlazno i dolaznog aerodroma, kao i polje za unos poletne staze imaju proveru unetih podataka. Ukoliko se unese podatak koji ne postoji u bazi, ispisuje se greška da aerodrom ne postoji ili da staza koja je uneta ne postoji na odlaznom aerodromu.



Slika 36 Popunjena ROUTE strana

Ukoliko malo detaljnije pogledamo sadržaj ove strane (Slika 36), primećuje se da se u gornjem desnom uglu nalazi brojač strana koji prikazuje trenutno aktivnu stranu od ukupnog broja strana. Klikom na dugme NEXT PAGE na fmc-u, otvara se druga strana na kojoj se vrši upis rute (putanje).

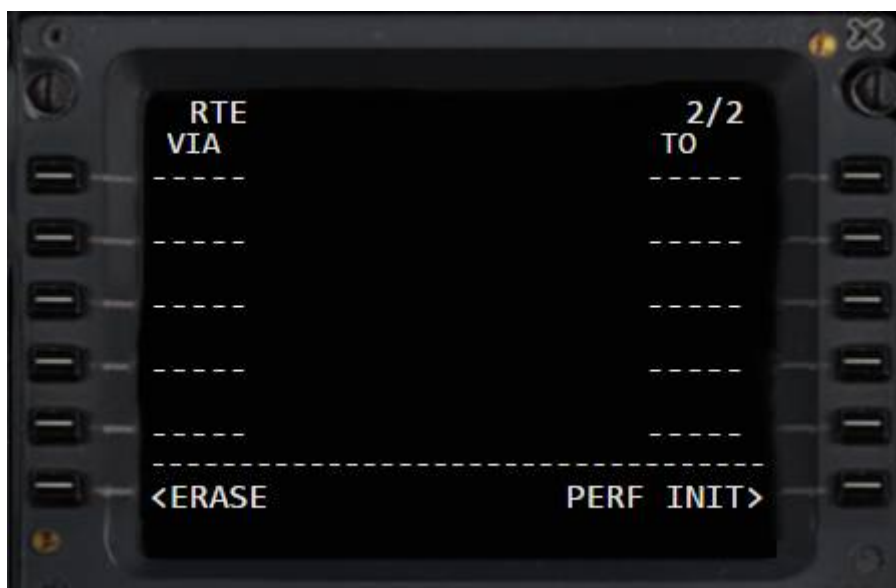
Svaka ruta predstavlja kombinaciju vazdušnih puteva, navigacionih tačaka. Redosled popunjavanja rute je:

- vazdušni put – navigaciona tačka
- vazdušni put – navigaciona tačka

dok ne dođemo do kraja naše predefinisane rute. Naravno, postoji i druga opcija koja se zove „Company route“ i koja automatski ispisuje rutu čim se unesu odlazni i dolazni aerodrom.

Naša ruta, kao što smo već napomenuli glasi VAL N732 POD. Primetićete da ona počinje i završava se navigacionom tačkom. Nije moguće započeti vazdušnim putem, jer je neophodna tačka ulaza na isti. Ako bismo napravili paralelu sa automobilom i autoputem, to bi značilo da automobil može da se pojavi na autoputu bez prethodnog prolaska kroz uključenje. Pošto sam već napomenuo da je redosled upisivanja rute vazdušni put - navigaciona tačka, to se malo kosi sa rečenicom da je nemoguće započeti rutu vazdušnim putem. Kako bi se ovaj mali „nesporazum“ izbegao, ukoliko se direktno uključujemo na neku tačku, bez prethodnog prolaska kroz neku tačku, neophodno je u vazdušni put ukucati DCT, što je skraćeno od „direct“. Na ovaj način, naš fmc zna da se uključujemo direktno na neku tačku i neće biti problema prilikom unosa rute ili u toku samog procesa navigacije.

Na slici 37, predstavljen je izgled ove strane u nepopunjenom obliku. Na crticama sa leve strane potrebno je upisati vazdušni put, dok se sa desne strane upisuje navigaciona tačka. Kako ne bi došlo do zabune, u gornjim uglovima ove strane je ispisano VIA i TO (preko-do).



Slika 37 Route - druga strana

Slika 38 predstavlja popunjenu stranu za naš let iz Beograda za Tivat.



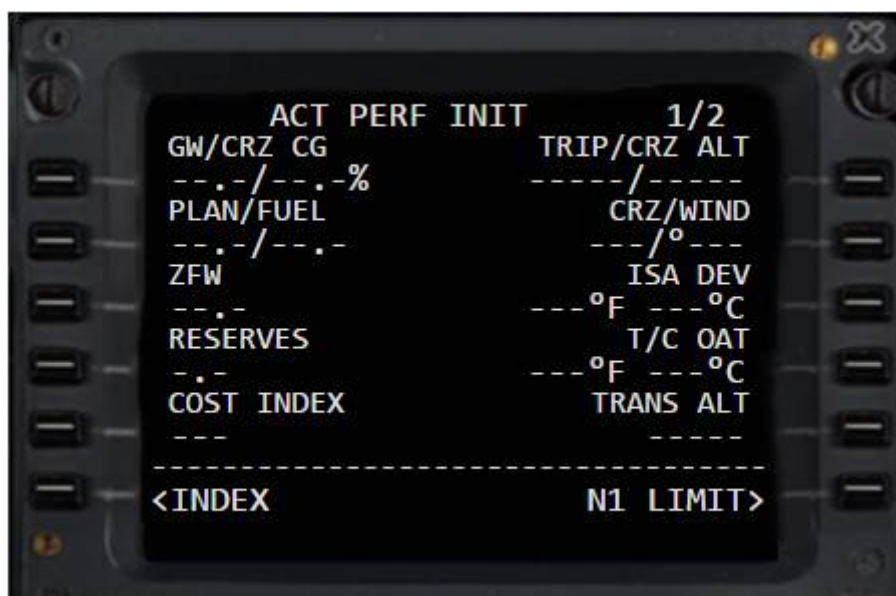
Slika 38 Route - popunjena druga strana

Nakon ove strane, pritiskom na dugme R6 (PERF INIT) odlazi se na stranu na kojoj se upisuju performanse vazduhoplova.

5.5 PERF INIT strana

Polako dolazimo do jedne od najzanimljivijih i najznačajnijih strana, a to je Performance Init page. Kao što joj samo ime kaže, radi se o performansama i raznim kalkulacijama, što ovu stranu izdvaja od drugih u pogledu zanimljivih dešavanja.

Već u samom startu, primećujemo da se od nas zahteva da unesemo više različitih parametara. Pojedini parametri se sami upisuju nakon što upišemo neke druge koji su u direktnoj vezi sa njima, poput proračuna ukupne težine aviona, dok se pojedini parametri upisuju čisto informativno, poput vetra. Svaki parametar na ovoj strani u određenoj meri utiče na celokupne performanse vazduhoplova tokom sve tri faze leta (poletanje, krstarenje, sletanje), ali s obzirom da se oslanjaju na podatke iz još nekoliko sistema koji se nalaze u avionu, u ovom slučaju i neće imati toliko uticaja.

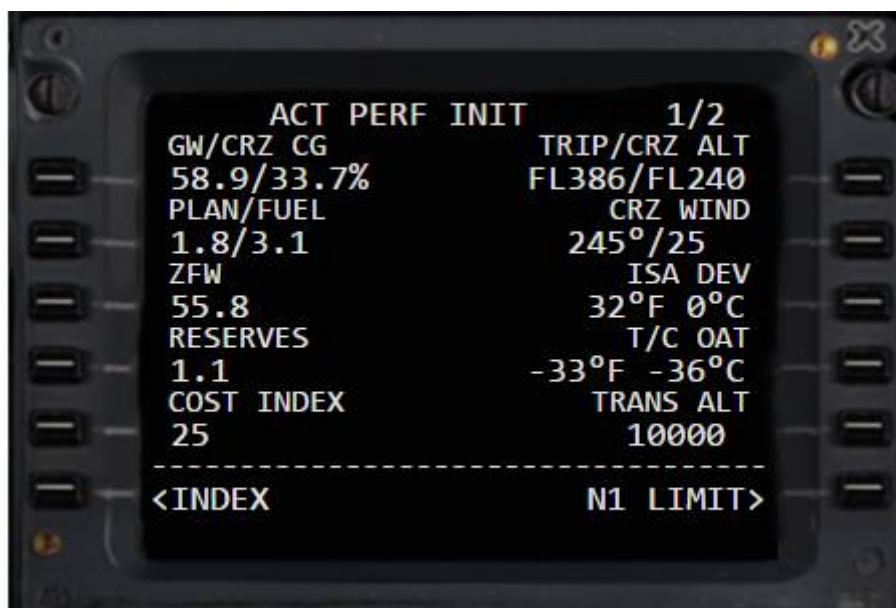


Slika 39 Performance page

Za početak, neophodno je objasniti pojmove sa ove stranice. Za pojedine možete i naslutiti šta znače, ali hajde da prođemo kroz svaki od njih.

- GW/CRZ CG – Gross Weight / Cruise Center of Gravity, odnosno ukupna težina aviona koja se automatski izračunava i centar gravitacije tokom faze krstarenja izražen u procentima;
- PLAN/FUEL – Planned fuel / Total Fuel, odnosno koliko goriva planiramo da potrošimo za naš put i koliko je goriva ukupno sipano u avion.
- ZFW – Zero Fuel Weight, odnosno težina aviona sa putnicima, teretom i prtljagom, ali sa nultom težinom goriva, tj bez goriva. Parametar goriva se sabira sa parametrom ZFW i dobija se GW.
- RESERVES – rezervno gorivo
- COST INDEX – po definiciji predstavlja odnos troškova goriva u odnosu na ostale troškove. Za ovaj avion, minimalna vrednos je 0, a maksimalna 500. U suštini, manji CI (Cost Index) će rezultovati manjoj potrošnji goriva, manjoj krstarećoj brzini, manjoj stopi penjanja i spuštanja aviona, itd. Veći CI radi suprotno.

- TRIP/CRZ ALT – TRIP je automatski proračunata optimalna visina leta, dok se u CRZ ALT upisuje visina krstarenja (Cruising Altitude)
- CRZ WIND – prosečan pravac i jačina vetra na visini krstarenja
- ISA DEV - International Standard Atmosphere Deviation, odnosno atmosferski model kako se pritisak, temperature, gustina i viskoznost Zemljine atmosfere menja na različitim visinama.
- T/C OAT – Top of Climb Outside Temperature, odnosno spoljna temperatura tokom faze dostizanja konačne visine krstarenja
- TRANS ALT – Transition Altitude, odnosno visina iznad nivoa mora na kojoj se uređaj za merenje visine (koji radi uz pomoć atm. pritiska) postavlja na standardnih 1013 milibara i visina se izražava u nivoima leta



Slika40 Performance strana - popunjena

5.5.1 Programska realizacija strane

Na slici 40 nalazi se primer popunjene strane. Kao ZFW, stavljena je težina od 55.8t, dok je za gorivo uneta ukupna težina od 3.1t. Funkcija koja vrši proračun Gross Weight-a uz pomoć 2 gore navedena parametra se nalazi na slici 41.

Da bi simulacija proračuna bila realnija, stavljena je maksimalna težina aviona koja je uzeta iz originalnog Boeing-ovog dokumenta, kao i maksimalna težina goriva koju ovaj tip aviona može poneti izražena u tonama. Ukoliko je broj unet za ZFW manji ili jednak maksimalnoj vrednosti, taj broj se sabira sa količinom goriva i upisuje kao GW. U suprotnom se na fmc-u ispisuje greška da je uneta težina iznad maksimalne i postupak se mora ponoviti.

```

214 function calculateGrossweight_L1(){
215
216     var maxZFW = 62.732;
217     var maxFuel = 20.894;
218     console.log('izvršava se');
219     if(Number($('#ZFW').html()) <= maxZFW){
220
221         var grossweight = Number($('#ZFW').html()) + Number($('#FUEL').html());
222         $('#GW').html(grossweight);
223
224     }else{
225         $('#labela').html("GROSS WEIGHT ABOVE MAX VALUE");
226     }
227
228 }

```

Slika 41 Gross Weight funkcija

Funkcija koja proverava unetu količinu goriva postoji i ima za cilj da proveri da li je uneta količina goriva ispravna. Na ovaj način, izbegnuta je opcija unosa veće količine goriva nego što avion može primiti. Radi na sličnom principu kao i prethodna funkcija koja proračunava ukupnu težinu aviona, osim što ima dodatnu proveru jer se upisuje gorivo planirano za let i ukupna količina goriva koja se nalazi u avionu.

```

143 function unosFuelZaGW_L2(){
144
145     var maxFuel2 = 20.9;
146     var input = $('#labela').html().trim();
147     var parts = input.split("/");
148
149
150     if (parts.length==1){
151         input = Number(input);
152         if(input<=maxFuel2){
153             $('#FUEL').html(input);
154             localStorage.setItem('unosFuelZaGW_L2_input', input);
155             $('#labela').html("");
156
157         }else{
158
159             $('#labela').html("INCORRECT FUEL 1");
160
161         }
162     }
163     else{
164         parts[1] = Number(parts[1]);
165         console.log(parts[1]);
166         if(parts[1]<=maxFuel2){
167             $('#PLAN').html(parts[0]);
168             $('#FUEL').html(parts[1]);
169             $('#labela').html("");
170
171         }else{
172
173             $('#labela').html("INCORRECT FUEL 2");
174
175         }
176     }
177
178 }

```

Slika 42 Funkcija za upis goriva

Preostale funkcije koje se bave upisivanjem unetih podataka u odgovarajuće lebele se nalaze na slici 43.

```
90
91 function TransitionLevel_R5(){
92     var input = Number($('#labela').html().trim());
93     console.log(input);
94     $('#TA').html(input);
95     localStorage.setItem('TransitionLevel_R5_input', input);
96     $('#labela').html("");
97 }
98
99 function ALTitude_R1(){
100     var input = $('#labela').html().trim();
101     console.log(input);
102     $('#OPTCRZ').html(input);
103     localStorage.setItem('Altitude_R1_input', input);
104     $('#labela').html("");
105 }
106
107
108 function COST_INDEX_L5(){
109     var input = Number($('#labela').html().trim());
110     $('#CI').html(input);
111     localStorage.setItem('COST_INDEX_L5_input', input);
112     $('#labela').html("");
113 }
114
115 function RESERVES_L4(){
116     console.log("klik");
117     var input = $('#labela').html().trim();
118     console.log(input);
119     $('#RESERVES').html(input);
120     localStorage.setItem('RESERVES_L4_input', input);
121     $('#labela').html("");
122 }
123
124
```

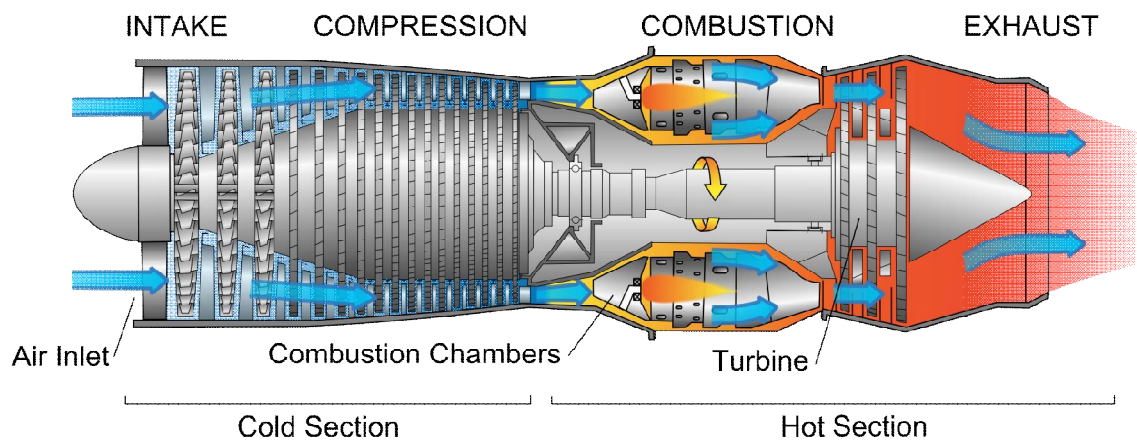
Slika 43 Ostale funkcije

5.6 N1 LIMIT strana

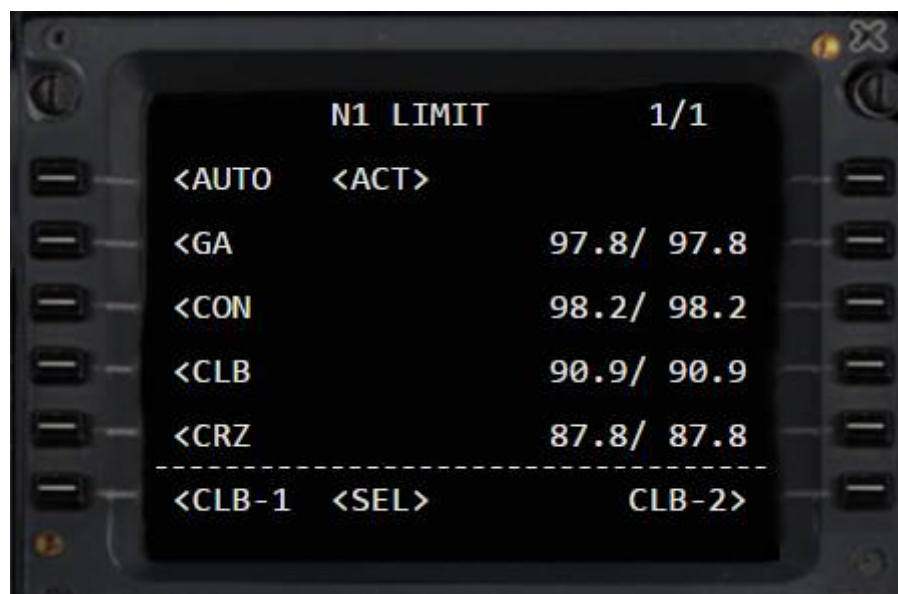
Strana koja nema mnogo interakcije, tu je radi pružanja informacija koje dobija iz senzora koji prate rad motora. Glavni zadatak ove strane je da prikaže maksimalan broj obrtaja prednjih lopatica motora izražen u procentima. Prednje lopatice (fan blades) imaju zadatak da uvuku određenu količinu vazduha do kompresora, koji zatim taj vazduh sabija i meša sa gorivom koje se zapali unutar komore za sagorevanje, a zatim dolazi do turbine koja tu mešavinu goriva i vazduha izbacuje sa druge strane motora i na taj način stvara potisak. S obzirom da se lopatice okreću velikom brzinom, neophodno je konstantno praćenje njihovog broja obrtaja (u procentima) kako ne bi došlo do eventualnog oštećenja istih koje zatim mogu uleteti u kompresor i izazvati eksploziju unutar motora.

Kako bi se to izbeglo, u samom motoru se nalazi nekoliko senzora koji prate sve parametre, počevši od obrtaja spoljašnjih (N1) lopatica, preko temperature unutar komore za sagorevanje do broja obrtaja unutrašnjih lopatica i još nekoliko stvari koje u ovom radu nećemo detaljnije obrađivati.

Na slici 44 nalazi se poprečni presek mlaznog motora radi lakšeg uvida u gore objašnjenu situaciju.



Slika 44 Poprečni presek mlaznog aviona



Slika 45 N1 Limit strana

5.6.1 Programska realizacija strane

Na slici 45 je izgled N1 Limit strane. Pošto se svi podaci koji se ovde prikazuju dobijaju iz gore pomenutih senzora čiji rad nije simuliran, ova strana je statična sa „hard coded“ demo podacima. Svi elementi su uneti u N1_LIMIT.page stranu, dok se N1_LIMIT.js strana sastoji samo iz jedne callback funkcije (callback_N1_LIMIT()) koja poziva otvaranje strana na klik.

```
1      N1 LIMIT      1/1
2
3      &lt; AUTO      &lt; ACT&gt;
4
5      &lt; GA      97.8/ 97.8
6
7      &lt; CON      98.2/ 98.2
8
9      &lt; CLB      90.9/ 90.9
10
11     &lt; CRZ      87.8/ 87.8
12     -----
13     &lt; CLB-1  &lt; SEL&gt;      CLB-2&gt;
14
```

Slika 46 N1_LIMIT.page

```
1  function callback_N1_LIMIT() {
2      unbindAll();
3      $(init).on('click', function(){
4          openPage('SCREEN_INIT_REF', "callback_SCREEN_INIT_REF");
5      });
6      $(crz).on('click', function(){
7          openPage('ECON_CRZ', "callback_ECON_CRZ");
8      });
9
10 }
11
```

Slika 47 N1_LIMIT.js

5.7 DEP/ARR Index strana

Ova strana omogućava posadi da odabere odlazne i dolazne putanje (SID, STAR). Ukoliko se strana otvori direktno klikom na dugme DEP ARR bez prethodno unetih polaznih i dolaznih aerodroma, na mestu gde bi oni trebalo da stoje upisani se nalaze (kao i na većini strana koje nisu popunjene) samo crtice. Strana se automatski popunjava odmah nakon unosa aerodroma unutar route strane. Na slici 48 predstavljena je strana bez podataka, dok se na slici 49 vidi kako strana izgleda nakon što su podaci povučeni preko javascriptfunkcije koja preko JSON.parse opcije iz lokalne memorije pretraživača izvlači podatke unete u identifikatore unete za oba aerodroma, a zatim ih postavlja unutar odgovarajućeg div-a.



Slika 48 DEP/ARR prazna strana

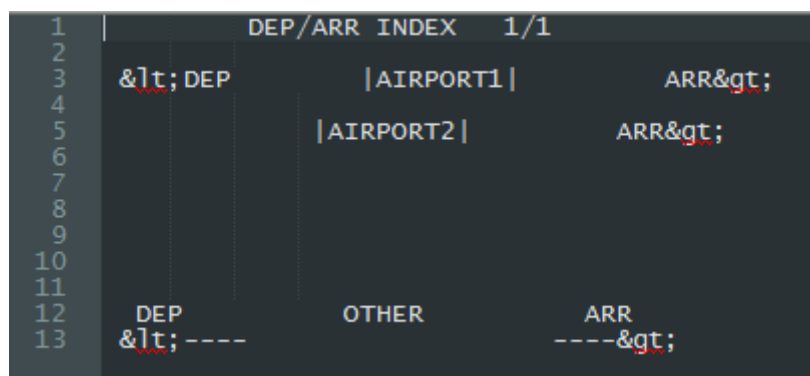


Slika 49 DEP/ARR Strana sa podacima

5.7.1 Programska realizacija strane

Programska realizacija je vrlo jednostavna. S obzirom da se radi sa svega dve promenljive, nije potrebno detaljno objašnjavati svaku liniju koda. Jedna promenljiva čuva podatke polaznog aerodroma, dok druga čuva podatke dolaznog. Podaci se izvlače iz lokalne memorije i parsiraju u vidu JSON-a, a zatim ispisuju u vidu stringa na odgovarajuće mesto.

Na slikama 50 i 51 nalaze se template i javascript strane.



DEP/ARR INDEX 1/1		
<DEP	AIRPORT1	ARR>
	AIRPORT2	ARR>
DEP	OTHER	ARR
<----		---->

Slika 50 DEP/ARR page



```
1 function callback_DEP_ARR() {
2   unbindAll();
3
4   setVariable(AIRPORT1, '----');
5   setVariable(AIRPORT2, '----');
6
7   $(init).on('click', function(){
8     openPage('PERF_INIT', "callback_PERF_INIT");
9   });
10
11   $(l1).on('click', function(){
12     openPage('SID', "callback_SID");
13   });
14   $(r1).on('click', function(){
15     openPage('STAR', "callback_STAR");
16   });
17   $(r2).on('click', function(){
18     openPage('STAR', "callback_STAR");
19   });
20
21   DEP_ARR11();
22 }
23
24 function DEP_ARR11(){
25   var d = JSON.parse(localStorage.getItem('RTE_ACCEPT_L1_data'));
26   var a = JSON.parse(localStorage.getItem('RTE_ACCEPT_R1_data'));
27   $('#AIRPORT1').html(d.ICA0);
28   $('#AIRPORT2').html(a.ICA0);
29 }
30
```

Slika 51 Funkcije unutar DEP/ARR javascript fajla

5.8 Take off reference page

Poletanje i sletanje su dve ključne faze svakog leta i ujedno nose najveći faktor rizika. Ukoliko dođe do nepredviđene situacije tokom faze poletanja, postoje unapred propisane procedure sa kojima su članovi posade upoznati. Piloti su prošli obuku na simulatorima i u potpunosti su spremni za svaki mogući (poznati) scenario. Ova stranica ne upućuje na te procedure, ali daje relevantne podatke vezane za bezbednost vazduhoplova tokom faze poletanja, kao i kalkulaciju performansi.

Na prvi pogled strana izgleda zatrpana gomilom podataka, ali njih ima tačno onoliko koliko je potrebno. Ukoliko se ekran подели na levu i desnu stranu, postaje jasnije i uočljivije o čemu se radi. Leva strana sadrži podatke o stepenu izvučenosti flapsova¹⁵, procentima N1 obrtaja predviđenih za poletanje, centru gravitacije izraženom u procentima i jedinica za trim¹⁶. Sa desne strane nalaze se V brzine i ukupna težina vazduhoplova.

Svaki avion, bez obzira na veličinu, ima svoje V brzine koje se izračunavaju. U ovom slučaju (i u slučaju svakog komercijalnog aviona) postoje tri različite brzine.

- V1 – Brzina izražena u čvorovima (knots) nakon koje avion mora poleteti bez obzira na otkaze ili kvarove. V1 je žargonski zove i „brzina posle koje nema povratka“, odnosno, ukoliko je avion prešao V1 brzinu, neće moći da se zaustavi do kraja poletno sletne staze i mora poleteti.
- Vr – Rotate brzina, odnosno brzina izražena u čvorovima po čijem se dostizanju inicira podizanje aviona u vazduh, odnosno njegova „rotacija“, te otud i ime V rotate.
- V2 – Minimalna brzina pri kojoj avion može bezbedno ostati u vazduhu.

¹⁵Flapsovi su ekstenzije na napadnim ivicama i zadnjem delu krila čijim se izvlačenjem na određenu poziciju (pod određenim uglom) smanjuje brzina a povećava uzgon, što znatno smanjuje potrebnu distancu za poletanje i sletanje.

¹⁶Trimovanje je mogućnost pomeranja horizontalnog stabilizatora (repna horizontalna krila) pod određenim uglom radi dobijanja boljeg uzgona.



Slika 52 TakeOff strana

5.8.1 Programska realizacija strane

S obzirom da se većina podataka na ovoj stranici učitava preko raznih senzora, odnosno računara koji prikuplja podatke, obrađuje ih i šalje fmc-u, nije ostalo mnogo toga da se programira sem funkcije koja proverava da li je uneti podatak o flapsovima unutar propisane granice i funkcije koja uzima ukupnu težinu aviona.

Sama funkcija je prikazana na slici 53, a sastoji se iz nekoliko promenljivih, od kojih jedna uzima uneti broj, druga čuva maksimalnu poziciju izvlačenja flapsova za poletanje i treća koja čuva minimalnu poziciju.

Prostom proverom preko „if”petlje, utvrđujemo da li je uneti broj manji ili jednak maksimalnoj vrednosti i istovremeno veći ili jednak minimalnoj, a zatim se kroz „switch” naredbu propušta kroz 4 različita slučaja. Ukoliko uneti broj nije jedinica, dvojka ili petica, ispisuje se greška o netačno unetom podatku i od korisnika se zahteva da ponovo unese ispravan podatak.

Na slici 54 se nalazi i sam templejt ove strane koja ne odstupa mnogo od svih prethodnih.

```

22 function flaps(){
23
24     var position = Number ($('#labela').html().trim());
25     var positionMAX = '5';
26     var positionMIN = '1';
27
28     console.log(position);
29
30     if(position <= positionMAX && position >= positionMIN){
31
32         console.log('1-5');
33         switch(position){
34
35             case 1:
36                 position = '1&deg;';
37                 break;
38             case 2:
39                 position = '2&deg;';
40                 break;
41             case 5:
42                 position = '5&deg;';
43                 break;
44             default:
45                 position = '--&deg;';
46                 $('#labela').html('INCORRECT FLAPS SETTINGS');
47         }
48
49         $('#DEGREES').html(position);
50         $('#labela').html('');
51     }else{
52         $('#labela').html('INCORRECT FLAPS SETTINGS');
53         $('#DEGREES').html(position = '--&deg;');
54     }
55 }
56
57 }

```

Slika 53 Funkcija za proveru flapsova

1		TAKEOFF REF	1/2	
2	FLAPS		QRH	V1
3	DEGREES		131>	131
4	RED		VR	
5	87.1/87.1		135>	135
6	CG	TRIM		V2
7	26.1%	5.12	142>	142
8			GW /	TOW
9			GW /	
10	RUNWAY		RWY	REMAIN
11	RW12		RW12 ---00FT	
12			-----SELECT	
13	&t; INDEX		QRH OFF>	

Slika 54 Takeoff reference page

5.9 Climb page

Climb page, ili stranica sa podacima o penjanju, sadrži podatke dobijene putem raznih proračuna računara koji konstantno vrši obradu podataka dobijenih sa raznih senzora i

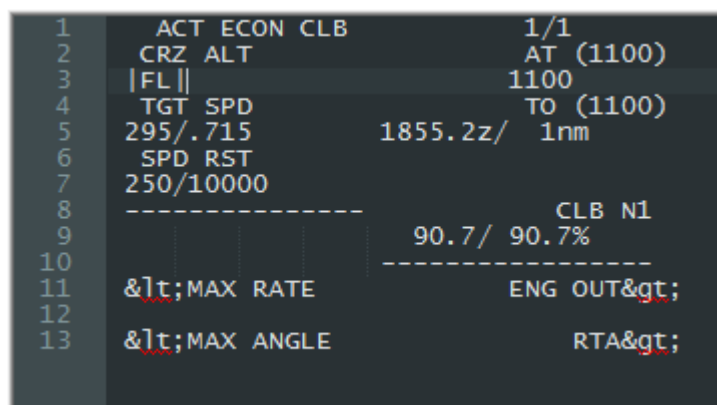
instrumenata. Tako na primer, da bi izračunao optimalnu brzinu za penjanje, računar mora uzeti u obzir nekoliko podataka:

- Težinu;
- Centar gravitacije;
- Pravac i jačinu vetra na svim visinama tokom faze penjanja;
- Atmosferski pritisak;
- Tropo pauzu;
- N1 rotacije;
- Restrikcije u vidu brzine i/ili visine na određenim usputnim tačkama, itd...

Iza ovih stvari stoji vrlo kompleksna matematika u koju nećemo zalaziti s obzirom da nemamo potrebne izvore podataka koji su neophodni za realizaciju ove stranice.

5.9.1 Programska realizacija strane

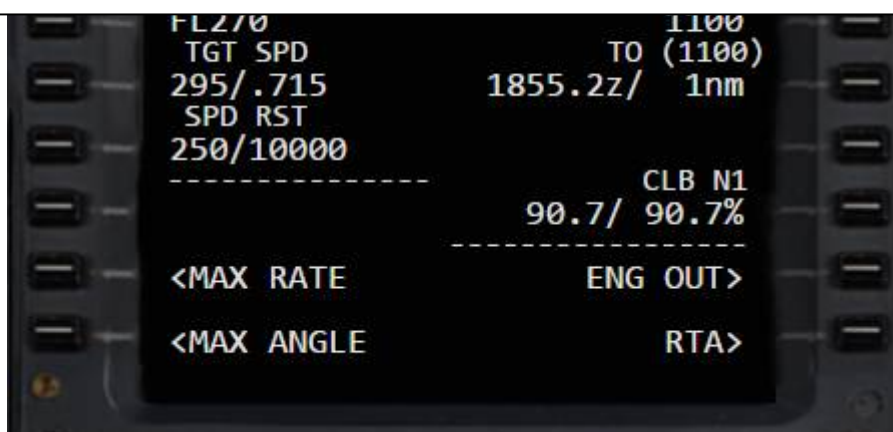
Templejt je prikazan na slici 55, dok se na slici 56 može videti kako stranica izgleda kada se pokrene.



Slika 55 Climb page template

Primer funkcije koja uzima vrednost unetu za visinu leta na performance strani i prosleđuje je i lepi unutar CRZ ALT polja:

```
function takeCrz(){  
    var a = JSON.parse(localStorage.getItem('Altitude_R1_input'));  
    $('#FL').html(a.altitude);  
}
```



5.10 Cruise page

Poput prethodne i ova stranica je puna (za nas) statičkih elemenata. Svrha stranice je praćenje situacije tokom faze krstarenja koje podrazumevaju praćenje pravca i jačine vetra, predviđena količina preostalog goriva na destinaciji i ostalih parametara.

Sa leve strane displeja, nalaze se parametri visine leta, broja obrtaja N1 turbine i predviđeno preostalo gorivo na destinaciji, do se sa desne strane nalaze parametri koji pokazuju optimalnu visinu leta u odnosu na trenutnu težinu aviona i maksimalnu visinu. Takođe, vide se i očitavanja sa senzora koji prikuplja podatke o vetru, tako da je i to dostupno.

5.10.1 Programska realizacijastrane

Na slikama 57 i 58, mogu se videti templejt strane i izgled prilikom selekcije iste.

Funkcija koja uzima visinu je ista kao i u prethodnom primeru:

```
function takeCrz(){
    var cruiseAltitude = JSON.parse(localStorage.getItem('Altitude_R1_input'));
    $('#FL').html(cruiseAltitude.altitude);
}
```

Funkcija koja uzima unetu destinaciju je takođe vrlo slična i izgleda ovako:

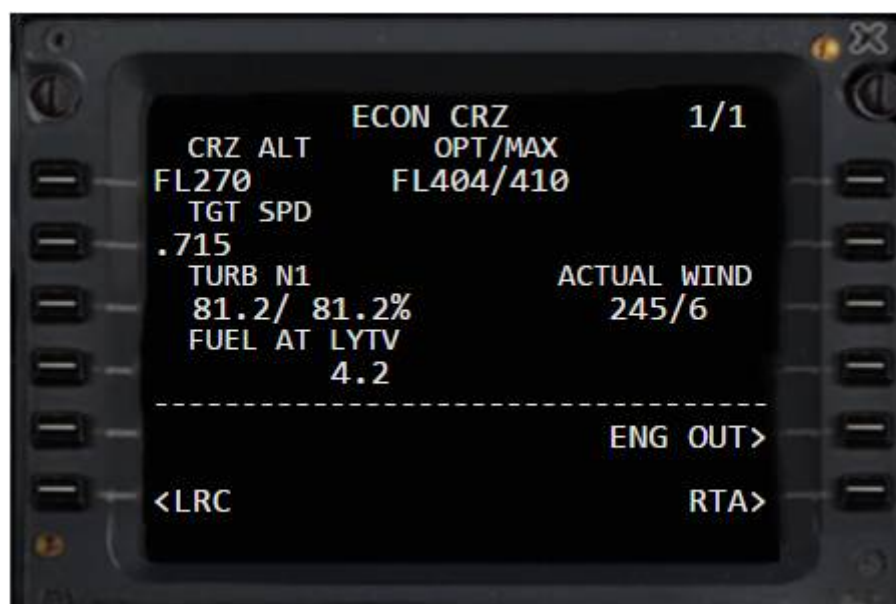
```
function takeDestination(){
    var destination = JSON.parse(localStorage.getItem('RTE_ACCEPT_R1_input'));
    $('#AIRPORT2').html(destination.ICAO);
}
```

```

1      ECON CRZ 1/1
2      CRZ ALT OPT/MAX
3      |CRZFL| |OPTFL|/|MAXFL|
4      TGT SPD
5      |TGTSPD|
6      TURB N1 ACTUAL WIND
7      |N1TURB|/ |N1TURB|% |DIR|/|KT|
8      FUEL AT |DEST|
9      |REMFUEL|
10     -----
11     ENG OUT>
12
13     <LRC RTA>|

```

Slika 57 Cruise page template



Slika 58 Cruise strana izgled

5.11 Descent page

Poslednja, ali ne i najmanje bitna strana koja je bitna za celu priču je „Descent” strana.

Na ovoj strani, prikazaci su podaci relevantni za bezbedno spuštanje aviona od krstareće visine, do visine propisane za započinjanje prilaza radi bezbednog sletanja. Ovde se mogu videti podaci poput: Visine na kojoj bi avion trebao biti, brzina kojom bi trebao da se spušta radi optimalne potrošnje goriva, zabrane u vidu brzine ispod određenih visina (u ovom slučaju ne brže od 240kt ispod 10000ft, daljinu do tačke spuštanja, kao i vreme u koje će se do tačke stići.

Ovaj deo nažalost nije odsimuliran, zbog toga što nije rađena simulacija kretanja kroz prostor pri određenim brzinama u realnom vremenu.

5.11.1 Programska realizacija strane

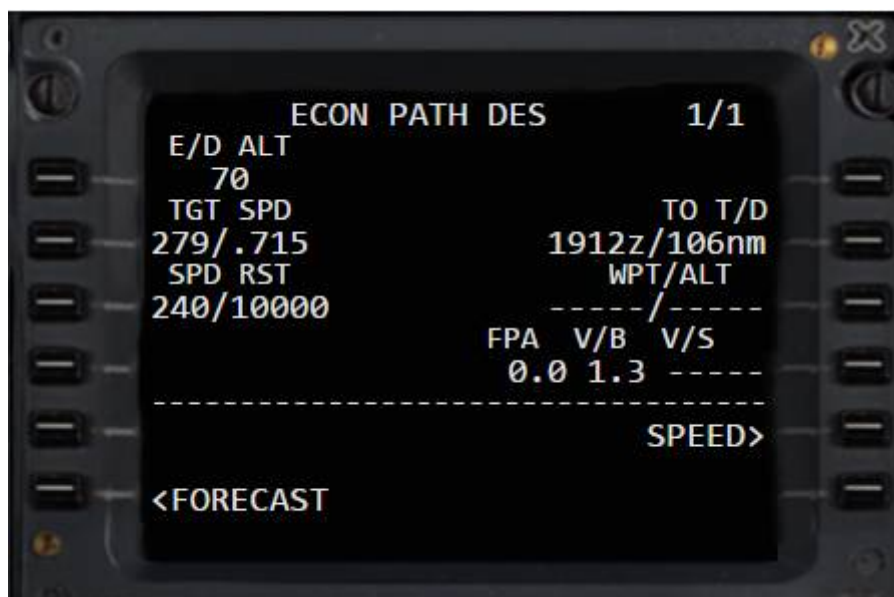
Templejt strane je prikazan na slici 59. Funkcija koja je ovde korišćena je za uzimanje unite vrednosti za visinu leta:

```
function takeCrz(){  
    var cruiseAltitude = JSON.parse(localStorage.getItem('Altitude_R1_input'));  
    $('#FL').html(cruiseAltitude.altitude);  
}
```

```
1      ECON PATH DES      1/1  
2      E/D ALT            AT |WPT1|  
3      |ALT|              |SPD|/|ALT1|  
4      TGT SPD            TO T/D  
5      |MACH|/|SPD|      |TIMETO|z/|MILES|nm  
6      SPD RST            WPT/ALT  
7      |SPD2|/|REST|      |WPT1|/|ALT1|  
8      FPA V/B V/S  
9      0.0 1.3 |VSPD|  
10     -----  
11     SPEED&gt;  
12  
13     &lt;t; FORECAST
```

Slika 59 Descent page template

Slika 60 predstavlja izgled strane nakon njenog otvaranja.



Slika 60 Descent page - izgled

6 Zaključak

Značaj simulacije je veoma veliki. Sam process izrade predstavlja izuzetno zahtevan, odgovoran i vrlo obiman posao. Osim podrazumevanog poznavanja programskih, skriptnih i jezika za označavanje, neophodno je vladanje materijom iz oblasti matematike, fizike, dinamike i drugih relevantnih disciplina vezanih za obrađivanu oblast.

Zahvaljujući višegodišnjem iskustvu i odličnom poznavanju sistema aviona tipa Boeing 737-800, nisam nailazio na probleme prilikom realizacije ovog simulatora.

Tokom izrade aplikacije, težilo se ispunjenju cilja u vidu izrade kvalitetnog i dobrog koda, lako čitljivog i lakog za dajle održavanje, kako bi se aplikacija dalje razvijala do višeg stepena funkcionalnosti. Tome su doprinele nove tehnologije i alati, a ukoliko bi se ovom projektu posvetilo još vremena, mogao bi ugledati svetlost dana i na svetskom tržištu kao svojevrsan simulator za obuku posade iz učionice. Ovo bi uštedelo vreme i novac, s obzirom da ne postoji drugi način treninga na ovom uređaju, osim u specijalno napravljenom simulatoru koji ima odrađenu celu pilotsku kabinu.

Funkcionalnost aplikacije, kao najzahtevniji i najbitniji zadatak, rešen je uz pomoć Notepad ++ alata kao i Sublime Text 3 alata. Kada je rad sa bazom podataka u pitanju,

korišćeni su: Navicat alat za modelovanje i dizajniranje baze podataka i PHPMyAdmin koji dolazi uz instalaciju XAMPP-a, koji lokalni računar pretvara u server i omogućava izradu web aplikacija u lokalnom okruženju.

U samoj bazi, smešteni su podaci o aerodromima, navigacionim tačkama, vazдушnim putevima, raznim frekvencijama, i tako dalje.

Efikasnost rada aplikacije je takođe veoma bitan faktor. Sama aplikacija nije zahtevna i nije neophodan moderan hardver da bi se izvršavali zadaci koje očekujemo. S obzirom da je aplikacija web based, jedino što je potrebno za njen rad je neki od mnogobrojnih pretraživača koji su svima dostupni. Jedina obimnija stvar je sama baza podataka koja ima nešto više od 200MB zbog obima podataka koje sadrži. Ukoliko bi se aplikacija postavila na hosting, ne bi bilo nikakvih problema u radu. Kod je HTML5kompatibilan i validiran od strane W3C organizacije.

Ostaje mogućnost daljeg razvoja aplikacije ili prevođenja koda na neki drugi programski jezik, a uz hardversku podršku u vidu izrade samog kućišta uređaja i displeja, moguće je i prodati ga na tržištu uz konkurentnu cenu.

7 Literatura

- [1]Vujaklija M. (2004): Leksikon stranih reči i izraza, Prosveta, Beograd, strana 817.
- [2]Janković R. i Nikolić N. (2009): Institut za strategijska istraživanja, Beograd, strana 8.
- [3]B. Radenković, M. Stanojević, A. Marković, „Računarska simulacija“, Fakultet organizacionih nauka, Saobraćajni fakultet, CIP – Katalogizacija u publikaciji – Narodna biblioteka Srbije, Beograd, 2004.
- [4] Internet Tehnologije, Mladen Veinovic, Aleksandar Jevremovic, Univerzitet Singidunum, Beograd 2013.