Izrada sustava za kreiranje i prikaz interaktivnih lista provjere za letačke operacije

Pađen, Maja

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:211734

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository

SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

MAJA PAĐEN

IZRADA SUSTAVA ZA KREIRANJE I PRIKAZ INTERAKTIVNIH LISTA PROVJERE ZA LETAČKE OPERACIJE

ZAVRŠNI RAD

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
ODBOR ZA ZAVRŠNI RAD

Zagreb, 30. ožujka 2018.

Zavod: **Zavod za aeronautiku**
Predmet: **Operativni postupci u zrakoplovstvu**

ZAVRŠNI ZADATAK br. 4793

Pristupnik: **Maja Pađen (0135238799)**
Studij: Aeronautika
Smjer: Pilot
Usmjerenje: Civilni pilot

Zadatak: **Izrada sustava za kreiranje i prikaz interaktivnih lista provjere za letačke operacije**

Opis zadatka:

Uvod. Opisati i definirati liste provjere. Opisati razvojno okruženje. Kreirati sučelje za izradu lista provjera. Izraditi sučelja za prikaz lista provjera. Izraditi sustav za pohranu lista provjere
Primjer prikazati interaktivne liste provjere. Zaključak

Mentor:

prof. dr. sc. Doris Novak

Predsjednik povjerenstva za završni ispit:

Sveučilište u Zagrebu

Fakultet prometnih znanosti

ZAVRŠNI RAD

IZRADA SUSTAVA ZA KREIRANJE I PRIKAZ INTERAKTIVNIH LISTA PROVJERE ZA LETAČKE OPERACIJE

DESIGNING OF INTERACTIVE CHECKLISTS SYSTEM FOR FLIGHT OPERATIONS

Mentor: izv. prof. dr. sc. Doris Novak

Student: Maja Pađen

JMBAG: 0135238799

Zagreb, svibanj 2019.

Sažetak

Rad opisuje proces dizajniranja i izrade sustava za interaktivne liste provjere koji obuhvaća definiranje zrakoplova, izradu listi provjere, interaktivno odrađivanje listi provjere te mogućnost uklanjanja listi ili zrakoplova iz sustava. Opisani su programski jezici html, css, php i javascript koji su korišteni za izradu sustava, kao i povijest listi provjera u zrakoplovstvu. Objašnjeno je sučelje sustava, programska logika na kojoj se temelji princip rada sustava, a sam rad sustava objašnjen je na primjeru. Razmotreni su i mogući pravci daljnjeg usavršavanja sustava u vidu estetike, sinkronizacije s određenim sustavima te pakiranja u samostalne aplikacije.

Ključne riječi: interaktivne liste provjere; liste provjere u zrakoplovstvu; razvoj web stranica

Summary

This thesis describes the process of designing and creation of interactive checklists system which consists of defining airplanes, creating checklists, interactive using of checklists and a possibility of removing checklists or airplanes. Programming languages such as html, css, php and javascript are described for they have been used for development of the system, as well as history of checklists in aviation. Interface of the system has been described as well as programming logic on which the principles of operation are based. The operation itself is explained on an example. Various ways of further improvement are also described. These include aesthetics, synchronization with other systems and packing into standalone applications.

Key words: interactive checklists; checklists in aviation; webpage development

Sadržaj

1. Uvod 1
2. Liste provjera u zrakoplovstvu 2
   2.1 Povijest listi provjera 2
   2.2 Važnost i značaj suvremenih lista provjera 3
3. Jezici korišteni u razvoju sustava 8
   3.1 HTML 8
   3.2 CSS 9
   3.3 PHP 10
   3.4 JavaScript 10
4. Razvoj sustava za izradu i prikaz lista provjere 12
   4.1 Programska potpora korištena u razvoju sustava 12
   4.2 Sučelje sustava za kreiranje i prikaz interaktivnih lista provjere 14
      4.2.1 Sučelje dijela sustava za kreiranje 15
      4.2.2 Sučelje dijela sustava za prikaz 16
   4.3 Razrada kôda sustava 18
5. Mogući pravci daljnjeg razvoja 24
   5.1 Unaprijeđenje sustava 24
   5.2 Integracija u druge sustave 25
6. Zaključak 27

1. Uvod

Od kada je Boeing 1935. prvi puta predstavio liste provjere povodom pada jednog od zrakoplova iz njihove flote, liste provjere postale su općeprihvaćene prvo među pilotima kompanije Boeing, a kasnije i među ostalim zrakoplovnim kompanijama te su kroz povijest uvelike doprinjele sigurnosti u avijaciji i smanjile broj nezgoda i nesreća [1].

U začecima zrakoplovstva liste provjere nisu bile potrebne zbog jednostavnosti ondašnjih zrakoplova, no kako se tehnologija razvijala i integrirala u nove zrakoplove, oni su postajali sve kompleksniji. Sukladno tome, povećavao se opseg posla i opterećenje pilota. Stoga su se počele događati nesreće s čestim tragičnim ishodima zbog ograničenja mentalnog kapaciteta čovjeka, a jedan od primjera je nesreća zrakoplova Boeing 299 koja je detaljnije objašnjena u slijedećem poglavlju. Liste provjere su predstavljale adekvatno rješenje za spomenuti problem zbog toga što su podsjetile pilota na radnje koja mora obaviti u određenoj fazi leta kao i na njihov redoslijed.

Ovaj završni rad opisuje način izrade sustava za interaktivne liste provjere koji predstavlja sredstvo za osiguravanje dodatne razine sigurnosti u zrakoplovstvu s obzirom da papirnate verzije u određenim fazama leta mogu biti nepraktične. Opisana je programska logika nužna za kôdiranje i programiranje na kojoj se temelji princip izrade sustava, način korištenja i ograničenja sustava, mogućnosti daljnjeg razvoja te popratnu programsku potporu i sadržaje koji su omogućili razvoj sustava na način kako je opisano.

Rad se sastoji od šest poglavlja. Nakon uvoda, u drugom poglavlju detaljnije je opisan povijesni razvoj i važnost listi provjera u zrakoplovstvu. Treće poglavlje opisuje kodne i programske jezike korištene za razvoj sustava za izradu i prikaz listi provjera. U četvrtom poglavlju se opisuje grafičko sučelje sustava, razrađuje se sklop logičkih procesa i programska pozadina korištena prilikom kodiranja sustava, a opisani su i primjeri korištenja sustava za izradu i prikaz listi provjera. Peto poglavlje razmatra daljnje moguće pravce razvoja i unaprijeđivanja sustava te integracije i prilagođavanja različitim zrakoplovima i zrakoplovnim sustavima. Potom slijedi zaključak koji obuhvaća najvažnije dijelove svakog razrađenog poglavlja. Na kraju rada se nalazi popis literature i slika te dodatak radu.

2. Liste provjera u zrakoplovstvu

2.1 Povijest listi provjera

Nakon petogodišnjeg razvoja, kompanija Boeing je 1935. pokrenula revoluciju vojnog zrakoplovstva s novim modelom 299, poznatijim i kao „B-17 Flying Fortress“ [2][3]. Tadašnji najbolji piloti bili su odabrani za let, no usprkos njihovim kvalifikacijama i dostignućima, nakon zatrčavanja i početnog penjanja zrakoplov je doživio slom uzgona, nagnuo se i pao s tragičnim posljedicama za posadu. Istraga nesreće pokazala je da zrakoplov nije imao nikakvu tehničku grešku već je uzrok nesreće bio ljudski faktor. Naime, B-17 je bio najnapredniji zrakoplov toga doba s uvlačećim podvozjem, novom vrstom zakrilaca, četiri motora s pripadajućim kontrolama, električnim trimerima i propelerima s konstantnom brzinom vrtnje čiji se napadni kut morao namještati ovisno o brzini. To je, u odnosu na prijašnje manje zahtjevne zrakoplove, značilo značajno povećanje u opsegu posla posade te su zaboravili otpustiti blokadu upravljačkih površina zrakoplova što je uzrokovalo nesreću. Zrakoplov je proglašen prekompleksnim da bi čovjek upravljao njime.

Kompanija Boeing je, kako bi izbjegla stečaj i smirila javnost, ponudila rješenje u vidu listi provjere čija je svrha bila podsjetiti posadu na stavke koje je potrebno odraditi u određenim fazama leta kao i na redoslijed odrađivanja [1]. Stoga su, inicijalno, četiri liste provjere postale obvezne za sve pilote kompanije Boeing: polijetanje, let, prije slijetanja i poslije slijetanja. Model 299 je nakon uvođenja navedenih sigurnosnih mjera uvršten u flotu Američkih zračnih snaga gdje je godinama letio bez incidenta.

Povodeći se za primjerom Boeing-a, postupno su i druge zrakoplovne kompanije uvele praksu korištenja listi provjere kako bi povećale razinu sigurnosti kako na vojnim tako i na civilnim letovima. S napretkom tehnologije i razvojem sve kompleksnijih zrakoplova sa sofisticiranijim sustavima povećao se i broj lista provjera za određeni zrakoplov što je dovelo do suvremenih lista provjere koje sadržavaju razrađene postupke za svaku fazu leta uključujući i radnje za postupke u nuždi.

2.2 Važnost i značaj suvremenih lista provjera

Suvremene liste provjere za određeni zrakoplov potječu od samog proizvođača koji izrađuje pripadajuću dokumentaciju za tip zrakoplova. One, uz ostale informacije, sačinjavaju priručnik zrakoplova, a na temelju njega operator kreira svoje liste provjere koje ne smiju biti u sukobu s informacijama iz priručnika. Liste provjere koje kreira operator objavljuju se u operativnom priručniku kompanije te ih odobrava nadležna zrakoplovna vlast. Prema Commission Regulation (EU) No 965.2012 dijelu ORO.GEN.110 „Operator će uspostaviti sustav listi provjera za svaki tip zrakoplova koji će posada koristiti za sve faze leta u normalnim i abnormalnim uvjetima te uvjetima koji zahtjevaju postupke u nuždi kako bi se osiguralo da se poštuju operativne procedure propisane u operativnom priručniku. Dizajn i način primjene listi uzeti će u obzir ljudski faktor i važeću relevantnu dokumentaciju proizvođača zrakoplova“ [4]. Sadržaj listi provjera utvrđuje se prema Consolidated unofficial AMC/GM to Annex III (Part-ORO) SUBPART MLR — Manuals, Logs and Records: „Normalne procedure i odgovornosti posade trebaju uključivati slijedeće:

- prijeletne postupke,
- postupke prije polijetanja,
- postavke visinomjera,
- postupke za vožnju,
- polijetanje,
- penjanje,
- smanjenje buke,
- krastarenje,
- spuštanje,
- prilaz,
- pripreme za slijetanje,
- vizualni prilaz,
- instrumentalni prilaz,

- prilaženje kruženjem,
- produžavanje na novi prilaz,
- normalno slijetanje,
- poslije slijetanja,
- postupke za operacije na mokrim i kontaminiranim površinama.

Procedure i odgovornosti posade u nuždi trebaju uključivati slijedeće:

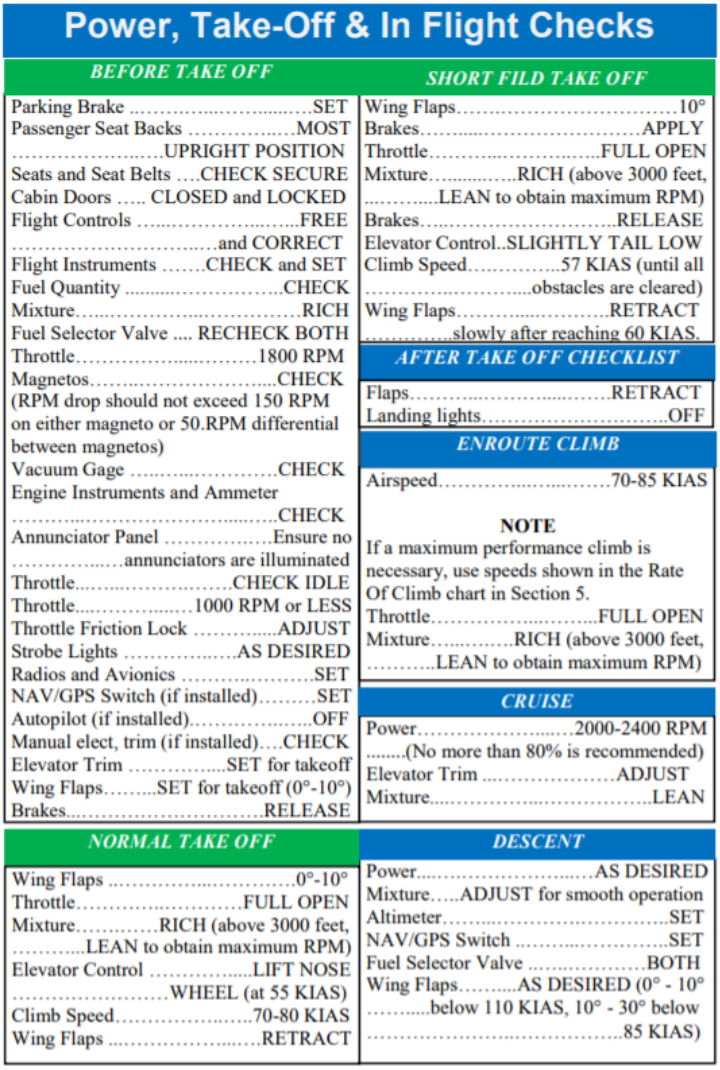
- nesposobnost posade,
- postupke u slučaju požara,
- gubitka prešurizacije,
- prekoračenja strukturalnih limita zrakoplova,
- komunikaciju u slučaju nužde,
- udara groma,
- otkaza motora i drugih sustava,
- paljenja sustava za izbjegavanje terena ili drugih zrakoplova,
- smicanja vjetra,
- prisilnog slijetanja.“ [4].

Liste provjere mogu biti u papirnatoj, kao što je vidljivo na slici broj 1 ili elektronskoj verziji, obje imaju svoje prednosti i nedostatke. Papirnate liste imaju nedostatke u vidu nepreglednosti i težeg snalaženja dok su elektronske vezane za uređaje poput tableta, što otežava dostupnost prilikom raznih tehničkih poteškoća poput prazne baterije ili pregrijavanja.

Generalno gledajući, liste se dijele na liste za normalne postupke, koje obuhvaćaju postupke za sve faze leta koji se odvija bez poteškoća, i za postupke u nuždi. Potonje sadržavaju radnje koje je potrebno odraditi u slučaju neke izvanredne situacije kao što su otkazi raznih sustava ili kvarovi.

Primjer liste provjere Hrvatskog Zrakoplovnog Nastavnog Središta za zrakoplove generalne avijacije korištene u školovanju pilota prikazan je na slici broj 1 [5]. Liste provjere su prikazane različitim bojama kako bi označile različite faze leta pa tako zelena

boja predstavlja radnje na tlu, a plava radnje u zraku. Kao i većina listi provjere, odrađuje se na način da se pročita stavka pa potom odradi radnja nakon čega se još jednom provjeri stavka.



Power, Take-Off & In Flight Checks

BEFORE TAKE OFF

Parking Brake .....SET
Passenger Seat Backs .....MOST
.....UPRIGHT POSITION
Seats and Seat Belts ....CHECK SECURE
Cabin Doors ..... CLOSED and LOCKED
Flight Controls .....FREE
.....and CORRECT
Flight Instruments .....CHECK and SET
Fuel Quantity .....CHECK
Mixture.....RICH
Fuel Selector Valve .... RECHECK BOTH
Throttle.....1800 RPM
Magnetos.....CHECK
(RPM drop should not exceed 150 RPM
on either magneto or 50.RPM differential
between magnetos)
Vacuum Gage .....CHECK
Engine Instruments and Ammeter
.....CHECK
Annunciator Panel .....Ensure no
.....annunciators are illuminated
Throttle.....CHECK IDLE
Throttle.....1000 RPM or LESS
Throttle Friction Lock .....ADJUST
Strobe Lights .....AS DESIRED
Radios and Avionics .....SET
NAV/GPS Switch (if installed).....SET
Autopilot (if installed).....OFF
Manual elect, trim (if installed)....CHECK
Elevator Trim .....SET for takeoff
Wing Flaps.....SET for takeoff (0°-10°)
Brakes.....RELEASE

NORMAL TAKE OFF

Wing Flaps .....0°-10°
Throttle.....FULL OPEN
Mixture.....RICH (above 3000 feet,
.....LEAN to obtain maximum RPM)
Elevator Control .....LIFT NOSE
.....WHEEL (at 55 KIAS)
Climb Speed.....70-80 KIAS
Wing Flaps .....RETRACT

SHORT FILD TAKE OFF

Wing Flaps.....10°
Brakes.....APPLY
Throttle.....FULL OPEN
Mixture.....RICH (above 3000 feet,
.....LEAN to obtain maximum RPM)
Brakes.....RELEASE
Elevator Control..SLIGHTLY TAIL LOW
Climb Speed.....57 KIAS (until all
.....obstacles are cleared)
Wing Flaps.....RETRACT
.....slowly after reaching 60 KIAS.

AFTER TAKE OFF CHECKLIST

Flaps.....RETRACT
Landing lights.....OFF

ENROUTE CLIMB

Airspeed.....70-85 KIAS

NOTE
If a maximum performance climb is necessary, use speeds shown in the Rate Of Climb chart in Section 5.

Throttle.....FULL OPEN
Mixture.....RICH (above 3000 feet,
.....LEAN to obtain maximum RPM)

CRUISE

Power.....2000-2400 RPM
.....(No more than 80% is recommended)
Elevator Trim .....ADJUST
Mixture.....LEAN

DESCENT

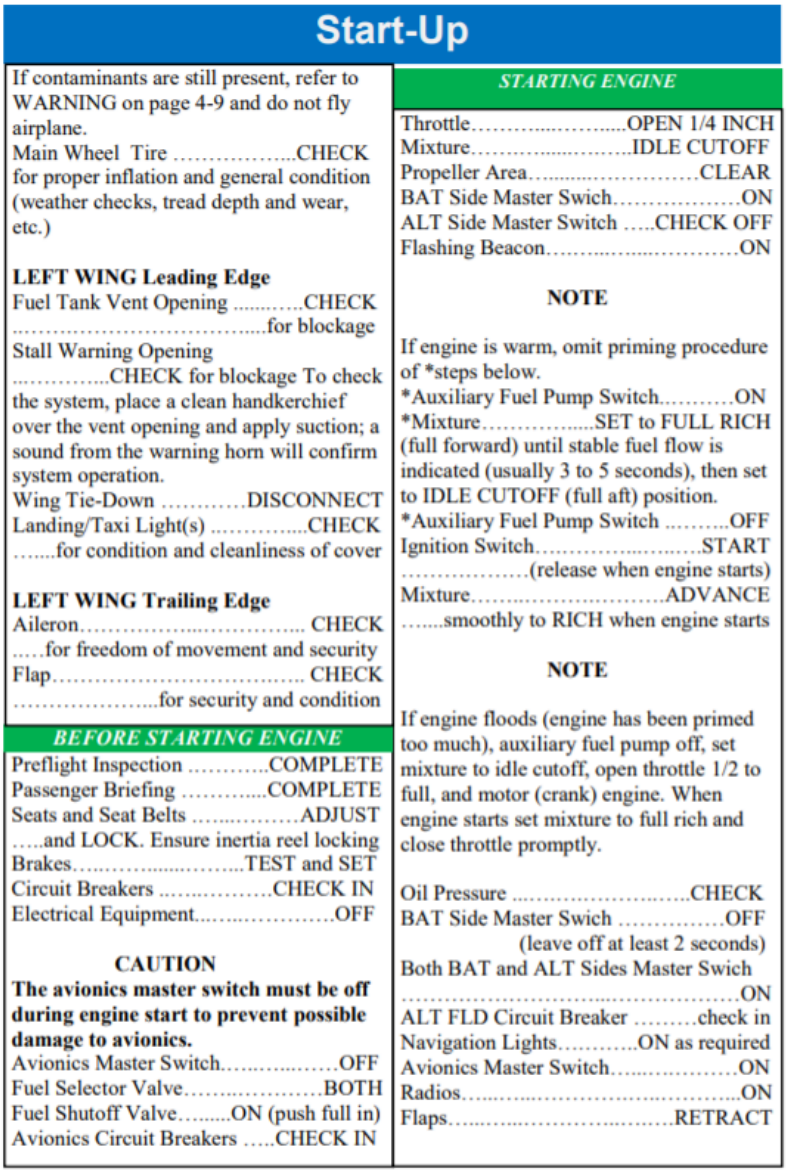
Power.....AS DESIRED
Mixture.....ADJUST for smooth operation
Altimeter.....SET
NAV/GPS Switch .....SET
Fuel Selector Valve .....BOTH
Wing Flaps.....AS DESIRED (0° - 10°
.....below 110 KIAS, 10° - 30° below
.....85 KIAS)

Slika 1: Lista provjere za zrakoplov Cessna 172R [5]

Namjera listi provjera u avijaciji je da pomognu posadi pri postavljanju konfiguracije za određenu fazu leta kako bi potvrdili da će zrakoplov uspješno svladati dotičnu fazu s minimalnim dodatnim fizičkim i psihičkim opterećenjem posade, utvrde pravilan redoslijed obavljanja radnji, pomognu pri uočavanju eventualnih odstupanja od normativa, definiraju dužnosti svakog pojedinog člana posade kako bi se osigurala koordinacija i usklađenost u zrakoplovima s višečlanom posadom te da posluže kao kontrola kvalitete

od strane nadređenih.

Tijekom povijesti liste provjere su se produljivale i skraćivale ovisno o kompleksnosti zrakoplova. Primjer predetaljne, a s time i nepregledne liste provjere vidljiv je na slici broj 2 na kojoj se nalazi lista provjere za zrakoplov Cessna 172R Hrvatskog Zrakoplovnog Nastavnog Središta.



Start-Up

If contaminants are still present, refer to WARNING on page 4-9 and do not fly airplane.
Main Wheel Tire ……..CHECK for proper inflation and general condition (weather checks, tread depth and wear, etc.)

LEFT WING Leading Edge
Fuel Tank Vent Opening .....…..CHECK ……..for blockage
Stall Warning Opening ……..CHECK for blockage To check the system, place a clean handkerchief over the vent opening and apply suction; a sound from the warning horn will confirm system operation.
Wing Tie-Down ……DISCONNECT
Landing/Taxi Light(s) ……...CHECK ……..for condition and cleanliness of cover

LEFT WING Trailing Edge
Aileron…….. CHECK ……for freedom of movement and security
Flap…….. CHECK ……..for security and condition

BEFORE STARTING ENGINE

Preflight Inspection ……..COMPLETE
Passenger Briefing ……...COMPLETE
Seats and Seat Belts ……..……ADJUST …..and LOCK. Ensure inertia reel locking
Brakes…….……..……..TEST and SET
Circuit Breakers ……..…….CHECK IN
Electrical Equipment……..……OFF

CAUTION
The avionics master switch must be off during engine start to prevent possible damage to avionics.
Avionics Master Switch……..……OFF
Fuel Selector Valve……..……BOTH
Fuel Shutoff Valve……..ON (push full in)
Avionics Circuit Breakers …..CHECK IN

STARTING ENGINE

Throttle……..……..OPEN 1/4 INCH
Mixture……..……..IDLE CUTOFF
Propeller Area…..……CLEAR
BAT Side Master Swich……ON
ALT Side Master Switch …..CHECK OFF
Flashing Beacon……..……..……ON

NOTE

If engine is warm, omit priming procedure of *steps below.
*Auxiliary Fuel Pump Switch……ON
*Mixture……...SET to FULL RICH (full forward) until stable fuel flow is indicated (usually 3 to 5 seconds), then set to IDLE CUTOFF (full aft) position.
*Auxiliary Fuel Pump Switch ……..OFF
Ignition Switch……..……START ……(release when engine starts)
Mixture……..……..……ADVANCE ……..smoothly to RICH when engine starts

NOTE

If engine floods (engine has been primed too much), auxiliary fuel pump off, set mixture to idle cutoff, open throttle 1/2 to full, and motor (crank) engine. When engine starts set mixture to full rich and close throttle promptly.

Oil Pressure ……..……..CHECK
BAT Side Master Swich ……OFF
(leave off at least 2 seconds)
Both BAT and ALT Sides Master Swich ……..……ON
ALT FLD Circuit Breaker ……check in
Navigation Lights……..ON as required
Avionics Master Switch…..……ON
Radios……..……..……..ON
Flaps……..……..……..……RETRACT

Slika 2: Primjer predetaljne liste provjere [5]

Iz primjera kompleksnijih zrakoplova bilo je vidljivo da one mogu imati i negativne posljedice na sigurnost samog leta pogotovo u kombinaciji s drugim negativnim čimbenicima kao što su nedovoljna fokusiranost posade na odrađivanje liste ili zamor posade. Preduge i previše detaljne liste provjere dovode do toga da ih posada odrađuje zbog primoranosti umjesto iz interesa za vlastitom sigurnošću i sigurnošću leta te se u takvim situacijama često događa da se određene stavke liste previde, ne odrade na pravi način ili se odrade u krivom trenutku.

3. Jezici korišteni u razvoju sustava

3.1 HTML

HTML[1] je prezentacijski jezik često korišten za razvoj web stranica, a njegovo prikazivanje omogućeno je pomoću web preglednika. Prva verzija HTML jezika predstavljena godine 1993., a inačica koja je danas raširena godine 1999. [6]. Razlikuje se od programskih jezika zbog toga što ne služi programiranju u užem smislu riječi – njime se ne može programirati određeni objekt niti izvoditi računske operacije već on samo služi upućivanju preglednika na zahtjevani način prikazivanja elemenata definiranih ovim jezikom.

Kodiranje u ovom jeziku nije složeno već hijerarhijski uređeno definiranjem elemenata koji određuju način prikazivanja u pregledniku te stoga ne zahtijeva složenu programsku potporu za izradu kôda. Hipertekstualni dokument je u suštini običan tekstualni dokument s .html ili .htm nastavkom te se stoga može s lakoćom pisati u bilo kojem tekstualnom pregledniku s mogućnošću uređivanja. Naravno, postoje programi prilagođeni namjeni kodiranja koji olakšavaju pronalaženje eventualnih grešaka napravljenih prilikom kodiranja elemenata sustava, kao što je Notepad++ koji je korišten u razvoju sustava za izradu i prikazivanje listi provjera.

Kodiranje u HTML jeziku vrši se kombiniranjem HTML elemenata koji se sastoje od parova oznaki (engl. tags) unutar kojih se definiraju atributi i sadržaj elemenata. Na slici broj 3 može se vidjeti primjer najjednostavnijeg HTML dokumenta koji predstavlja web stranicu.

[1] HyperText Markup Language

```
1   <!DOCTYPE html>
2
3   <html>
4
5     <head>
6       <title>Naziv web stranice</title>
7     </head>
8
9     <body>
10      <p>Ovo je odlomak.</p>
11    </body>
12
13  </html>
14
```

Slika 3: Primjer jednostavne HTML stranice

Početak svakog elementa dokumenta definiran je znakovima < i >, a kraj s < i />. Na samom početku elementa definira se jezik i inačica u kojem se kodira, a iza nje slijede oznake <html> koje označavaju početak sadržaja stranice. Unutar oznaka <head> definiraju se atributi same stranice kao što je naziv stranice i pozivaju se vanjski dokumenti koji određuju stiliziranje i interaktivno ponašanje stranice. Dio <body> označava segment unutar čijih oznaka se nalazi sadržaj koji će se prikazivati korisnicima. Tako se, na primjer, na slici broj 3 može vidjeti par <p> oznaka koje definiraju HTML odlomak teksta. Postoje i mnoge druge vrste oznaka koje definiraju različite sadržaje i njihov način prikazivanja bez određivanja dodatnih atributa pa tako oznake <img> određuju sliku, a <h> naslov teksta. Na kraju dokumenta zatvaraju se oznake </html> što označava kraj kôda.

3.2 CSS

Suvremene web stranice stilski su uređene kako bi bile što privlačnije posjetiteljima i usmjerile ih na sadržaje koje pružaju što nije moguće koristeći samo HTML jezik. Jezik koji se koristi za stilizaciju elemenata definiranih HTML jezikom je CSS[2] [6]. Iako i HTML pruža određene mogućnosti formatiranja elemenata, CSS ima puno raširenije mogućnosti pa se tako njime mogu odrediti fontovi, veličine, poravnanja, boje, pozadine, prozirnost,

[2] Cascading Style Sheets

kao i neka jednostavnija interaktivna ponašanja elemenata poput promjene boje teksta prilikom pritiska gumba ili postavljanja pokazivača iznad elementa. Dodatna prednost jezika CSS je mogućnost određivanja stila za više elemenata koji spadaju u određenu klasu. Tako se može definirati da će sadržaji sadržani u svim oznakama odlomka <p> biti plave boje ili nakošeni. Izgled se može odrediti i samo za neke odlomke <p> ukoliko se jezikom HTML imenuju željeni odlomci, a potom se jezikom CSS postavi željeni izgled za elemente koji su nosioci tog imena.

3.3 PHP

PHP[3] je skriptni programski jezik koji omogućava dinamičko generiranje HTML kôda [7]. Za razliku od javascripta, kôd se pokreće na serveru što znači da je sav kôd učitan i obrađen prilikom otvaranja web stranice te se servira pregledniku u obliku HTML kôda kao i sama web stranica. Upravo to mu daje određene prednosti kao što su mogućnosti enkripcije, sakupljanja podataka i izmjene datoteka na serveru. Potonje je posebno korišteno prilikom razvoja sustava za liste provjere kako bi se pojednostavio sustav i izbjegla potreba za korištenjem drugih jezika za upravljanje bazama podataka. Također omogućuje elementarni rad s bazama podataka i obrađuje podatke iz obrazaca koje korisnik stranice ispunjava.

Sam kôd se također može pisati u običnim tekstualnim preglednicima, no postoje i naprednije sučelja poput programa Laravel. Datoteke nose nastavak .php te mogu sadržavati i ostale programske jezike poput jezika HTML, CSS i Javascript. [6]

3.4 JavaScript

JavaScript je objektno orjentirani programski jezik baziran na Javi i C++ koji se pokreće na klijentskoj strani web stranice što mu omogućava odaziv na određene radnje korisnika u vremenu kao što je odrađivanje prethodno definiranih funkcija na zahtjev korisnika [6]. Većinski, no ne i isključivo, se koristi za razvoj web stranica. Omogućava implementaciju

[3] PHP: Hypertext Preprocessor

kompleksnih radnji na web stranicu što ga čini idealnom nadopunom jezicima HTML i PHP. Primjer kompleksnih radnji uključuje razne animacije, dinamičko osvježavanje sadržaja ili implementaciju vanjskih izvora informacija poput geolociranja i raznih nadogradnji. Implementacija u HTML ili PHP dokument vrši se pakiranjem kôda u <script> oznake ili pozivanjem na vanjski dokument nastavka .js.

4. Razvoj sustava za izradu i prikaz lista provjere

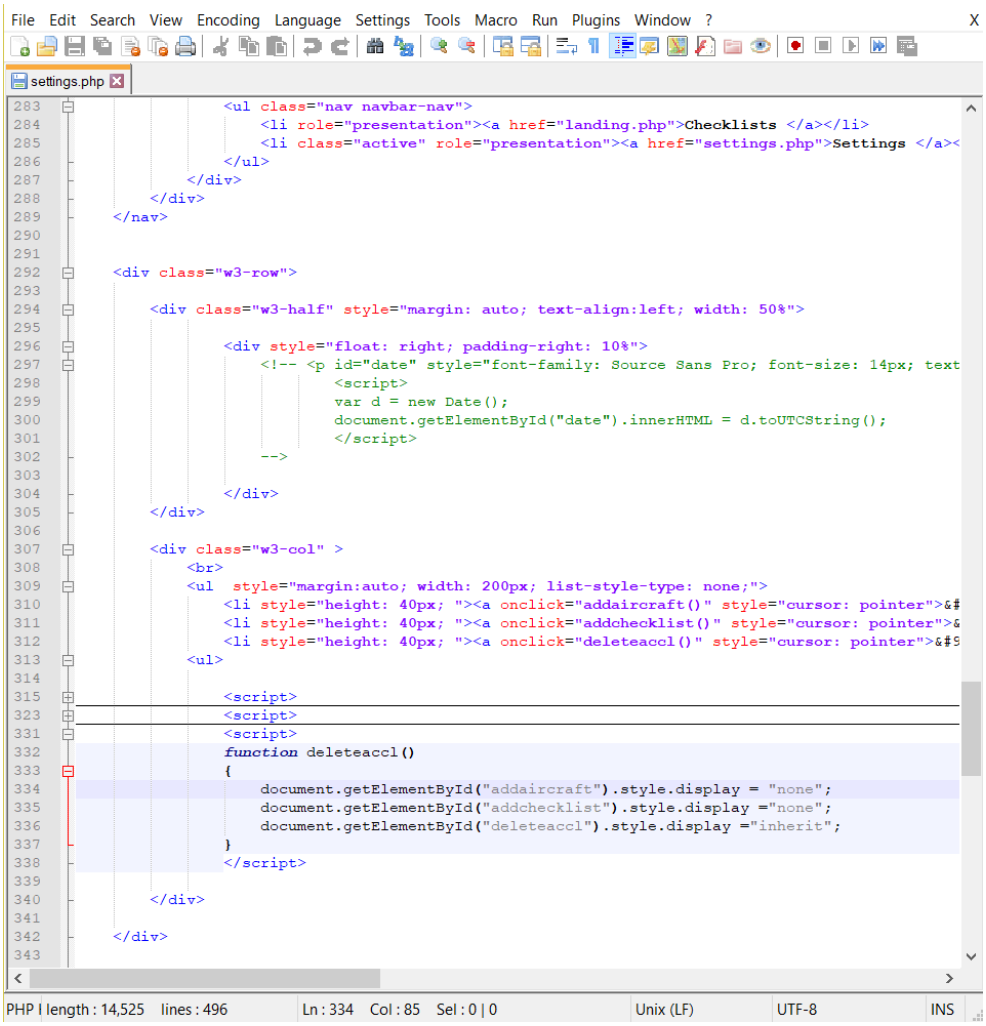
4.1 Programska potpora korištena u razvoju sustava

Temelj vizualnog dijela web stranice izrađen je u programu „Bootstrap Studio“ . Bootstrap Studio je program čija je namjena razvoj nekompleksnih web stranica bez potrebe za poznavanjem programskih jezika HTML i CSS. Njime se ostvarila struktura izgleda i responzivnost koja osigurava adekvatnu preglednost na svim vrstama uređaja na kojima je predviđeno korištenje sustava što uključuje računala, tablete i pametne telefone. Koristeći program također su definirane boje i fontovi koje će se koristiti za tekst i hiperveze te navigacijska traka na vrhu web stranice.

Korištena je profesionalna inačica Bootstrap Studio 2.2.4 bez ograničenja što je omogućilo razvojno okruženje po principu WYSIWYG[4] ali i dodatno uređivanje samog kôda generiranog spomenutom WYSIWYG metodom. S obzirom da program ima ograničenja poput nemogućnosti integracije PHP kôda, dio razvoja sustava je iz tog razloga, a i zbog preglednosti strukture kôda, nastavljen u programu Notepad++.

Notepad++ je u suštini blok za pisanje sličan verziji koja je ugrađena u operativni sustav Windows uz određena unaprijeđenja koja olakšavaju uočavanje nastalih grešaka prilikom kodiranja. Tako će, na primjer, program upozoriti korisnika ukoliko nije zatvorio oznake HTML kôda, omogućiti prividno skraćivanje kôda radi lakšeg snalaženja u duljim dokumentima i prikazati različite segmente kôda u drugačijim bojama. Od ostalih značajki program nudi i mogućnosti paralelnog uređivanja više dokumenata, prepoznavanje više od 20 programskih jezika, pokretanje skripti, integriranje raznih vanjskih dodataka i prikazivanje web stranice u nekoliko najčešće korištenih preglednika s obzirom da postoji mogućnost da različiti preglednici različito interpretiraju isti kôd.

[4] Engl. What You See is What You Get, označava razvojno okruženje koje se može koristiti za izradu web stranica bez potrebe za poznavanjem programskih jezika



Slika 4: Razvojno okruženje programa Notepad++

Na slici 4 vidljiv je primjer preglednog okruženja programa Notepad++. S lijeve strane brojčano su označeni redovi zbog lakšeg snalaženja. Prateći brojeve vidljivo je da su redovi 315 do 331 kolabirani zbog preglednosti ostatka kôda što je korisno ukoliko je korisnik usmjerio pažnju na drugi segment kôda. Kôd je prikazan različitim bojama pa je tako JavaScript koji se nalazi u redovima 315 do 338 prikazan crno-sivom kombinacijom, dok je HTML prikazan raznim varijacijama boja. Posebno se ističe zelena koja označava kôd koji je označen kao komentar što znači da se neće prikazati na web stranici niti će se izvršiti ono što se nalazi u njegovom sadržaju.

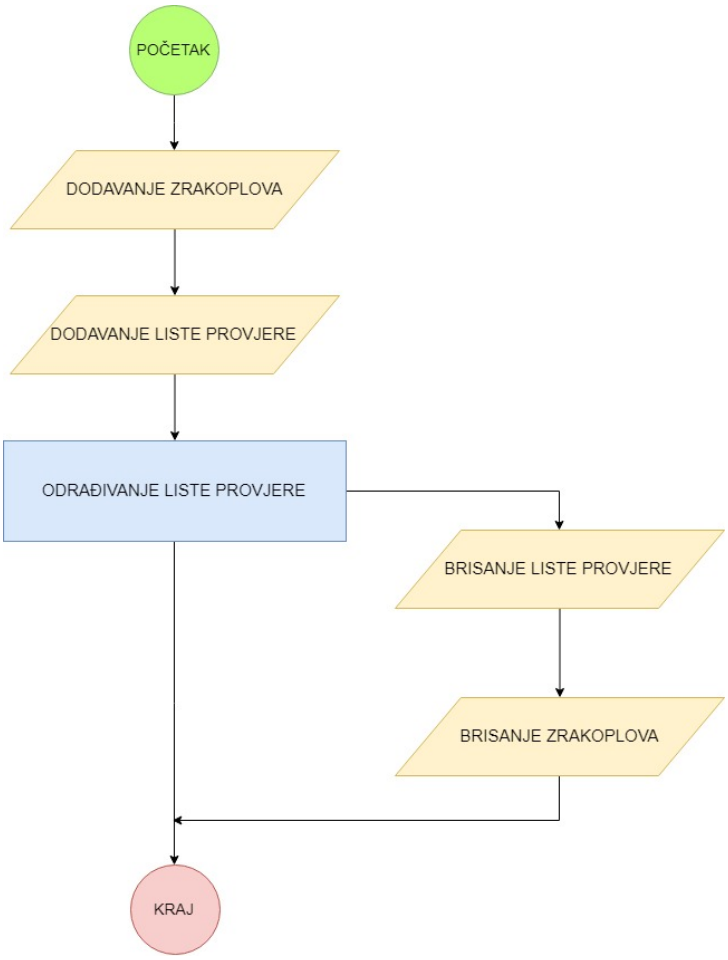
S obzirom da je sustav za izradu i prikaz interaktivnih listi provjere osmišljen kao web stranica kojoj će studenti i osoblje organizacije moći pristupiti sa svojih uređaja, za

pretpregled stranice tijekom razvoja korišteni su internet preglednici Google Chrome, Mozilla Firefox i Microsoft Internet Explorer.

4.2 Sučelje sustava za kreiranje i prikaz interaktivnih lista provjere

Vizualni dio sučelja za izradu i prikaz interaktivnih lista provjere dijeli se na dvije cjeline. Jedna je sučelje za odabir zrakoplova i odrađivanje lista provjere. Kako bi korisnik mogao odabrati zrakoplov za koji želi odraditi listu provjere, prvo ih treba kreirati. Dodavanje novih zrakoplova, kreiranje lista provjera i brisanje postojećih lista i zrakoplova vrši se u dijelu sustava koji logički predstavlja prvi dio sustava i označava postavke.

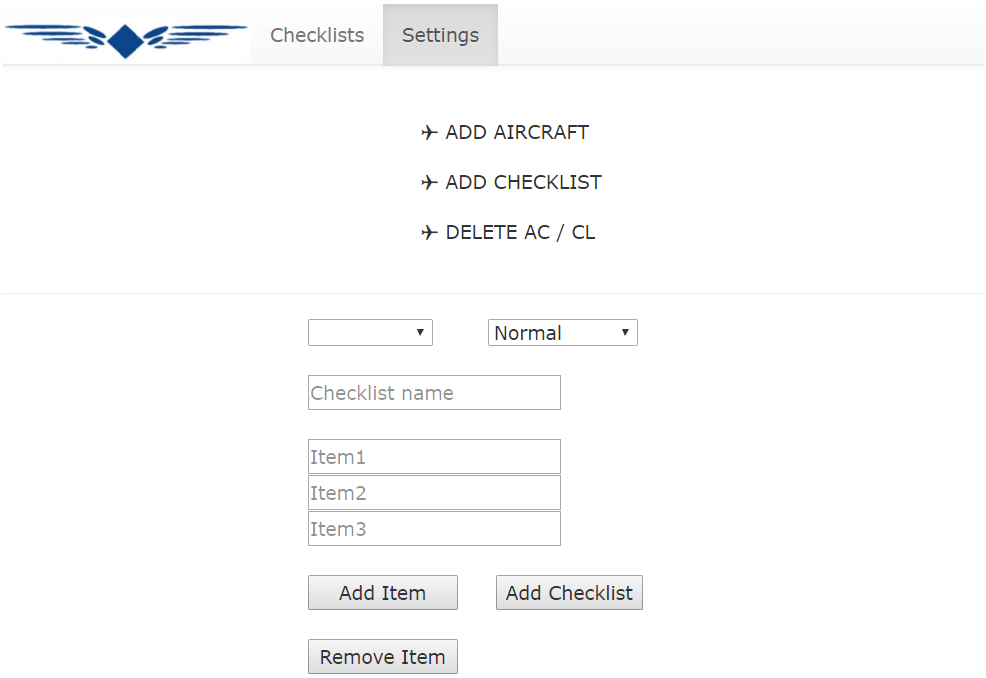
Cijeli princip korištenja sustava prikazan je na dijagramu na slici 5.



Slika 5: Dijagram toka korištenja sustava

4.2.1 Sučelje dijela sustava za kreiranje

Na slici 6 vidljiv je dio sučelja sustava za izradu lista provjera.



Slika 6: Sučelje sustava za izradu lista provjera

Logička cjelina od koje korisnik polazi je dodavanje novog zrakoplova što se vrši unošenjem četveroznamenkaste oznake zrakoplova, na primjer C172 ili A380, nakon čega sustav dodaje taj zrakoplov u svoju bazu podataka. Taj postupak je predstavljen blokom „Dodavanje zrakoplova“ na slici 5. Time se zrakoplov pojavljuje u svim padajućim izbornicima za odabir zrakoplova koji se nalaze na naslovnoj stranici za odrađivanje lista, u izborniku za brisanje zrakoplova i u sustavu za kreiranje lista provjere te na taj način omogućuje kreiranje lista provjere za taj zrakoplov.

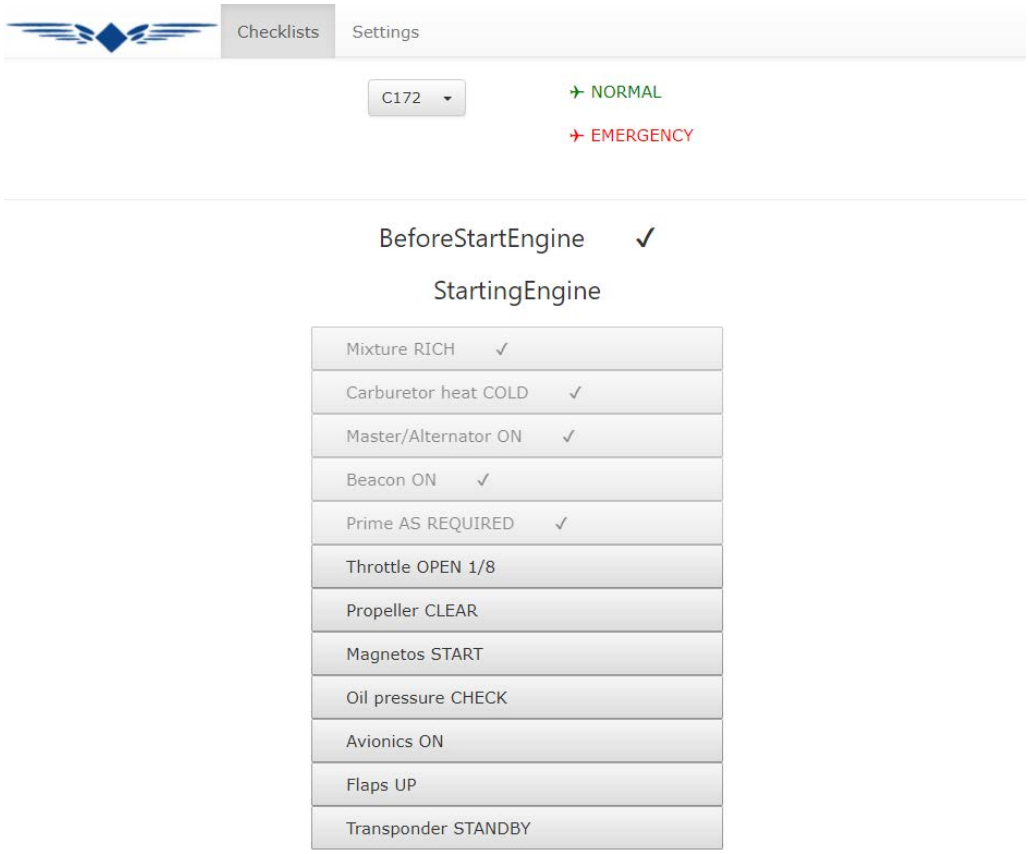
Slijedeći korak je kreiranje liste provjere predstavljeno blokom „Dodavanje liste provjere„ na slici 5. Sučelje od korisnika u obliku padajućih izbornika traži da unese zrakoplov i

vrstu liste provjere koja može biti normalna ili za postupke u nuždi. Korisnik pomoću gumba za dodavanje i oduzimanje određuje broj stavaka liste provjere te ih popunjava tekstom nakon čega potvrđuje listu, a sustav ju sprema u bazu podataka. Time se lista pojavljuje na naslovnoj stranici za odrađivanje lista provjere.

U slučaju potrebe, brisanje listi i zrakoplova omogućeno je u postavkama u izborniku za brisanje gdje se jednostavno odabere željena lista provjere ili zrakoplov te potvrdi odgovarajućim gumbom za brisanje nakon čega se liste automatski uklanjaju iz padajućih izbornika sustava, a predstavljeno je blokovima „Brisanje liste provjere“ i „Brisanje zrakoplova“ na slici broj 5.

4.2.2 Sučelje dijela sustava za prikaz

Prikaz i odrađivanje liste provjere predstavljeno je blokom „Odrađivanje liste provjere“ na slici broj 5, a izrađeno je na interaktivni način da omogućuje korisniku u svakom trenutku da zna gdje je stao s listom, čak i u slučaju prekida odrađivanja liste. Svaka stavka koju korisnik označi kao odrađenu poprima drugačiju boju, a pored teksta se pojavljuje kvačica. Označene stavke je nemoguće odznačiti kako bi se izbjeglo slučajno odznačavanje stavki u slučaju turbulencija ili drugih ometanja. Gumb za završavanje liste provjere pojavljuje se tek nakon označavanja svih stavki te liste provjere, a po njegovom stiskanju pojavljuje se kvačica pored imena liste provjere kako bi korisnik znao koje liste je odradio. Sve liste provjere za pojedini zrakoplov nalaze se na istoj stranici te funkcioniraju na principu padajućeg sadržaja što znači da su u početku vidljva samo imena listi, a ukoliko korisnik želi vidjeti njihov sadržaj ili ih odraditi mora stisnuti na naziv pojedine liste provjere. Normalne liste i liste za postupke u nuždi odvojene su izbornikom kako bi u slučaju nužde bila osigurana preglednost i olakšano nalaženje tražene liste provjere. Na slici 7 prikazan je dio sustava za prikaz i odrađivanje lista provjera.



Slika 7: Sučelje sustava za odrađivanje lista provjera

Ovakav, elektronski oblik listi provjera ima mnoge prednosti u odnosu na konvencionalnu, papirnatu verziju listi. Svim potrebnim listama je mnogo lakše pristupiti, preglednost je bolja, pilot jasno vidi koje stavke i koje liste je odradio što smanjuje mogućnost od nenamjernog preskakanja stavaka uslijed ometanja tijekom odrađivanja liste. Također, jedna od bitnijih prednosti je olakšana navigacija do listi provjera za postupke u nuždi što može biti od životne važnosti. Iako imaju mnoge prednosti, elektronske liste imaju i nedostatke, uglavnom vezane za tehničke nedostatke elektronskih uređaja na kojima se liste odrađuju. Tako glavne boljke predstavljaju autonomija i pregrijavanje uređaja, gašenje ekrana te interferencija s drugim aplikacijama. Stoga je preporučljivo da kompanije koje koriste elektronske verzije listi provjera imaju i papirnate verzije u zrakoplovu, što uglavnom i jest praksa, te da papirnata i elektronska verzija budu što sličnije kako bi se olakšalo snalaženje i odrađivanje.

4.3 Razrada kôda sustava

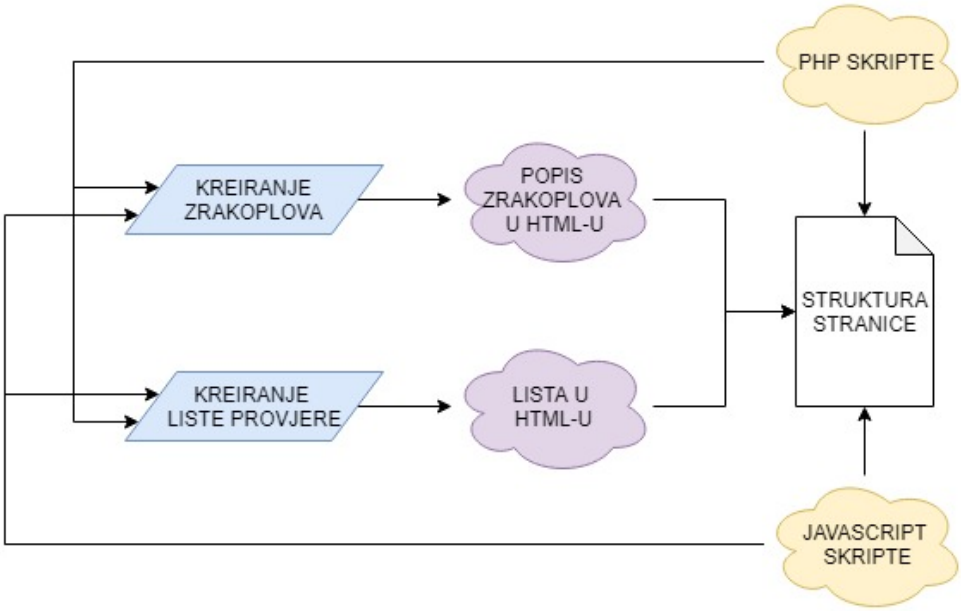
Struktura stranice sustava je pomoću jezika HTML i CSS, koristeći Bootstrap Studio i ručno pisani kôd, podijeljena na redove i stupce u svrhu strukturiranja sadržaja kako bi se osigurala responzivnost. Pojam responzivnost označava svojstvo web stranice da se prilagodi širini prozora odnosno rezoluciji ekrana uređaja pritom osiguravajući adekvatnu preglednost sadržaja. Po potrebi se može definirati restrukturiranje sadržaja pa tako sadržaji koji su bili smješteni jedan pored drugog se premještaju jedan ispod drugog kako se rezolucija smanjuje. Moguće je i odrediti da neki sadržaji nisu vidljivi ukoliko je veličina ekrana ispod određene vrijednosti, da se prikazuje na drugačiji način kao što je slučaj s navigacijskom trakom sustava za liste provjere ili da se prikazuju u drugoj veličini ili boji.

Kodiranjem u ta dva jezika također su određeni:

- poravnanja teksta
- margine
- veličine elemenata
- fontovi
- poveznice
- veze s drugim dokumentima
- padajući izbornici
- popunjavajući obrasci
- boje
- tablice
- neka jednostavnija ponašanja elemenata.

Potonje uključuje promjenu teksta na način da se dodaje kvačica i mijenja prozirnost stavke liste provjere kada korisnik klikne na nju te ju time označi kao odrađenu. Takvo svojstvo elemenata može se definirati i pomoću JavaScripta no ovaj način smanjuje mogućnost međusobne interferencije ponašanja elemenata jer se ne definira u zasebnoj skripti već se veže direktno uz element koristeći „onClick“ HTML naredbu. Prednost definiranja istog takvog ponašanja u JavaScriptu je veća preglednost i manja kompleksnost kôda, a s time i lakše otklanjanje eventualnih grešaka s obzirom da se više

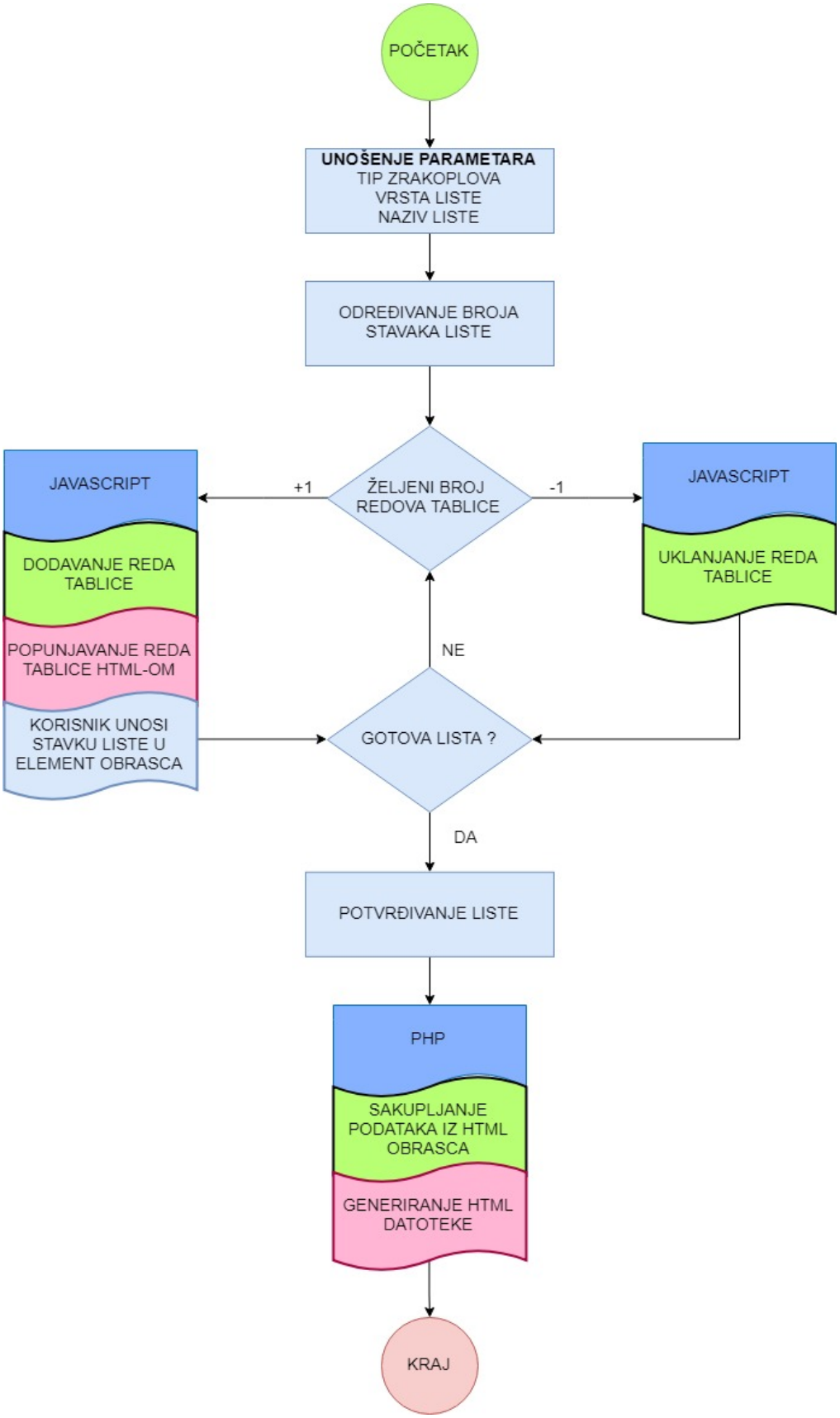
elemenata koji imaju isto ponašanje prilikom klika može povezati na jednu JavaScript skriptu. Radi lakšeg poimanja cjelokupne strukture web stranice ista je vizualno prikazana na slici 8, a potom pojašnjena.



Slika 8: Struktura web stranice

Jezikom HTML je definirana i nevidljiva tablica u dijelu sustava za izradu listi provjere na način da svaka stavka buduće liste provjere predstavlja red tablice. Osim što olakšava daljnje strukturiranje sadržaja i generiranje gotove liste koja je kasnije vidljiva u dijelu za odrađivanje, ova programerska prečica omogućuje dinamičko dodavanje i oduzimanje redova tablice, odnosno dodavanje i oduzimanje stavki buduće liste provjere. Samo dodavanje i oduzimanje redova te generiranje sadržaja redova tablice vrši se korištenjem JavaScripta koji definira novi HTML element obrasca unutar nove ćelije novonastalog reda. Elementi obrasca su sva polja koja korisnik ispunjava kako bi stvorio novu listu provjere, a završava pritiskom na gumb za dodavanje liste provjere. Tom radnjom aktivira se PHP skripta koja sakuplja informacije iz svih polja koja je korisnik popunio i na kôdom određeni način izrađuje novi HTML dokument ili sadržaj dodaje postojećem dokumentu ukoliko se radi o tipu zrakoplova za koji već postoje liste provjere. Dotični HTML dokument sadrži samo listu provjere spremnu za odrađivanje, bez strukture stranice, i kao takva se umeće u postojeću strukturu stranice u zasebnom dokumentu.

Na sličan ali manje kompleksan način vrši se i dodavanje novih tipova zrakoplova u sustav uz bitnu razliku što se ne izrađuje cijeli HTML dokument već se samo dodaje red postojećem dokumentu koji predstavlja opciju za taj tip zrakoplova u padajućim izbornicima sustava. Na slici 9 grafički je prikazan proces u kôdu koji vodi do nastanka nove liste provjere. Svjetlijom plavom bojom su prikazane radnje i odluke korisnika sustava za izradu lista provjere, tamnijom plavom je definiran programski jezik u kojem se odvija određeni logički slijed kôda, zelenom interakciju programskog jezika s jezikom HTML, a ružičastom generaciju HTML kôda naredbom drugog programskog jezika.

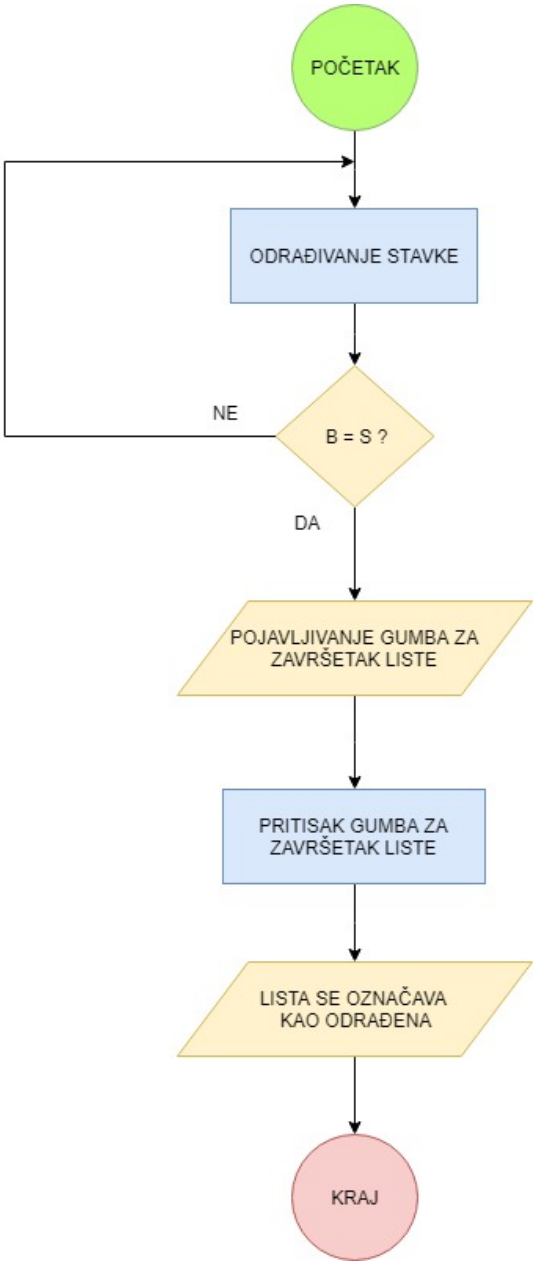


Slika 9: Dijagram toka izrade liste provjere

Dio sustava za uklanjanje zrakoplova i lista provjere iz sustava kodiran je u jeziku PHP zbog toga što taj programski jezik na najjednostavniji mogući način omogućuje komunikaciju i interakciju s ostalim dijelovima i datotekama sustava. U slučaju kada korisnik iz sustava želi ukloniti zrakoplov, PHP skripta briše dokument koji sadrži liste provjere koje su za njega napravljene te otvara dokument s popisom zrakoplova, pretražuje ga i uklanja traženu sintaksu koja sadrži ime zrakoplova. Ukoliko korisnik želi ukloniti samo određenu listu provjere, aktivira se PHP skripta koja otvara dokument s listama provjere za taj zrakoplov, pretražuje sintaksu koju sačinjava poveznica na padajući izbornik za tu listu i stavlja ju u kodne zagrade kako nebi više bila vidljiva na stranici. Ta lista još uvijek postoji u sustavu no zbog prevelike kompleksnosti kôda za uklanjanje i mnogih mogućnosti interferencija s ostalim skriptama, uklonjena je samo poveznica čime se onemogućuje njezino pojavljivanje. Višak kôda u dotičnoj datoteci ne utječe na stabilnost sustava niti bitno mijenja zahtjevani podatkovni prostor za pohranu na web poslužitelju.

Za kreiranje dijela za odrađivanje listi provjere korišteni su HTML, CSS i JavaScript. Neke dijelove bi bilo lakše i stabilnije iskodirati u PHP jeziku no međusobna suradnja JavaScripta i PHP jezika je kompleksnija i zahtjeva određene resurse poput poznavanja MySQL-a[5] za prijenos podataka između ta dva programska jezika. Jezikom HTML je postavljena struktura stranice i definirani su gumbi unutar nevidljive tablice koji predstavljaju stavke liste provjere dok je stiliziranje stranice i njezinih elemenata usavršeno jezikom CSS. Promjena izgleda gumba prilikom pritiska je, kao što je već spomenuto, izvršena JavaScriptom. Kako bi se spriječila mogućnost završavanja liste provjere dok nisu sve stavke izvršene, bilo je potrebno onemogućiti gumb za završetak. To je odrađeno na način da je dotični skriven dok sve stavke liste nisu označene, kada se pojavljuje. Taj segment je također programiran u JavaScriptu koristeći brojač koji je svojstven svakoj pojedinoj listi provjere, a povećava se prilikom pritiska na svaku stavku dotične liste provjere. Kada brojač dostigne vrijednost koja odgovara broju stavki liste, gumb za završavanje liste se pojavljuje. Korištena programska logika prikazana je na slici 10. Pritom je broj stavaka liste provjere označen slovom S, a brojač s B. Plavom bojom su prikazane radnje korisnika, a žutom radnje i kriteriji odluke sustava.

[5] Sustav za upravljanje bazom podataka [6]



Slika 10: Programska logika pojavljivanja gumba za završetak liste provjere

Sustav u cjelini nije jako kompleksan. Neki dijelovi su odrađeni prečicama u kodiranju i programiranju što kod sustava ove kompleksnosti i karakteristika neće dovesti do grešaka i zatajenja, a niti ne onemogućava mogućnost daljnjeg razvoja i implemetaciju u druge sustave.

5. Mogući pravci daljnjeg razvoja

5.1 Unaprijeđenje sustava

Sustav za kreiranje i prikaz interaktivnih lista provjera napravljen je kao model te je stoga predviđen za daljnje nadograđivanje i unaprijeđivanje. Osim mogućih neotkrivenih grešaka prilikom kodiranja i programiranja, a koje će se otkriti korištenjem sustava, postoje i drugi mogući pravci razvoja koji bi proširili područje primjenjivosti i praktičnost korištenja. Jedan od tih pravaca je dodavanje mogućnosti bojanja stavki liste provjere kako bi se naglasili segmenti liste koji su od drugačije važnosti i prioriteta prilikom odrađivanja. Izvedivo je i dodavanje detaljnijeg opisa pojedine stavke kako bi ona bila što razumljivija pilotima uz mogućnost dodavanja fotografije, skice ili animacije povezane s radnjom zahtjevanom u stavci liste provjere. To je posebno primjenjivo kod organizacija za školovanje pilota s obzirom da su oni manje iskusni i skloniji greškama. Takvo svojstvo bi se moglo integrirati u sustav na način da se dodatni sadržaj pokazuje prilikom postavljanja pokazivača iznad dotične stavke ili, kod prikazivanja na zaslonima na dodir, prilikom pritiska određenog gumba naupadljivih dimenzija s obzirom da takvi ekrani nemaju pokazivač. Negativni aspekt dodatnog sadržaja eventualno bi bila smanjena preglednost ali i sasvim sigurno mnogo veći potrebni prostor na web poslužitelju na kojem se nalazi stranica ili pohrani uređaja ukoliko se odabere taj način izvedbe sustava, a što će biti detaljnije objašnjeno u slijedećem odlomku.

Imajući na umu da pristup internetu u zrakoplovu nebi smio biti preduvjet za funkcioniranje sustava, isti je osmišljen da bude dostupan u vanmrežnom načinu rada te da prilikom povezivanja s internetom sinkronizira podatke o odrađenim listama provjere. Taj način rada je kompliciraniji za programiranje i nepraktičan za rad jer bi tada stranica trebala biti otvorena u trenutku kada je uređaj povezan na internet, nebi se smjela zatvarati prije kraja rada odnosno kraja leta i sav rad bi se trebao odvijati unutar jedne web stranice. Postoji i mogućnost da se kôd pomoću odgovarajuće programske potpore zapakira u aplikaciju koja će biti dostupna za instalaciju na uređajima s Android i iOS operativnim sustavima s obzirom da su najrašireniji na tabletima i pametnim telefonima. Tada bi aplikacija sama obavljala sav rad bez pristupa internetu, a prilikom povezivanja bi

postavljala datoteke s podacima na server od kuda bi one bile dostupne operateru zrakoplova.

5.2 Integracija u druge sustave

Spremanje podataka o odrađenim listama provjere uvjetovano je potrebama organizacije koja koristi sustav. Predviđeno je da za korištenje sustava svaki djelatnik ili student organizacije ima svoje podatke za prijavu koji će mu ovisno o poziciji u organizaciji omogućiti djelomičnu ili potpunu razinu dostupnosti. Tako bi piloti i studenti imali na raspolaganju samo dio za odrađivanje listi dok bi rukovodeće osobe u organizaciji imale mogućnost izrade lista provjere. Ukoliko bi se sustav na primjer prilagođavao Hrvatskom Zrakoplovnom Nastavnom Središtu, idealni način integracije bi bio putem AAI-Edu identiteta koji posjeduju svi studenti i nastavnici, no može se izraditi i nova baza podataka svojstvena sustavu.

Informacija koja je predviđena za spremanje sadržala bi ime osobe koja je odradila listu, ime liste, vrijeme u koje je odrađena i koliko je vremenski odrađivana. S obzirom da moderni mobilni uređaji imaju i mogućnost geolociranja, informaciji koja se sprema mogao bi se dodati i podatak s geografskim koordinatama. Informaciji se prema potrebi mogu dodati i drugi podaci koji se smatraju potrebnima, a idealni način spremanja bi bio u PHP dokumentu u obliku tablice na zasebnoj web stranici kako bi se podaci s lakoćom mogli uvesti u radnu knjigu u nekom od programa za rad s tablicama kao što je Microsoft Excel.

U nekim fazama leta, kao što su faze završnog prilaženja ili tijekom nužde, je veoma nepraktično koristiti tablet te bi iz tog razloga glasovno upravljanje sustavom bilo veoma prikladno. Pilot bi glasovno zahtjevao listu koju u datom trenutku želi odraditi ili ju aktivirao pritiskom na nju, a sustav bi čitao listu provjere što smanjuje potrebu odvraćanja pažnje prema tabletu prilikom zahtjevnih faza leta. U tom slučaju tablet bi morao biti spojen na slušalice pilota ili zvučnike što za negativnu posljedicu ima opasnost od interferencije prilikom komunikacije s kontrolom leta.

Na velikim zrakoplovima poput tvrtke Airbus već postoji ECAM[6] sustav za nadziranje zrakoplova koji koji je integriraran i u sam zrakoplov na način da se u slučaju prepoznatog kvara na određenom sustavu odgovarajuća lista provjere u nuždi automatski otvara za odrađivanje na jednom od zaslona zrakoplova [8]. U njemu su kvarovi klasificirani u tri kategorije ovisno o ozbiljnosti kvara i utjecaju na sigurnost leta, te sukladno tome imaju prioritet ukoliko postoji više kvarova istovremeno.

[6] Electronic Centralized Aircraft Monitor

6. Zaključak

Od trenutka kada su se prvi puta pojavile u zrakoplovstvu liste provjere su se mjenjale kroz povijest te su danas mnogo opsežnije nego što su bile u početku što je logično s obzirom da su se razvijale paralelno s razvojem tehnologije i zrakoplovstva. Suvremeni zrakoplovi su sve kompleksniji ali imaju i sve više autonomnih sustava koji rasterećuju pilota no i dalje zahtijevaju nadgledanje.

Cjelokupni sustav za kreiranje i prikaz interaktivnih lista provjera kodiran je i programiran na temelju vlastitog znanja i iskustva. Svaki programer ima svoj način razmišljanja i svoju programsku logiku na temelju koje će dobiti funkcionalno rješenje, stoga je i kôd koji stoji iza sustava jedinstven i objašnjen u radu. Model je izrađen i testiran na osobnom računalu u različitim preglednicima, uočene greške su ispravljene, a daljnje korištenje sustava i iskustva korisnika će omogućiti ispravljanje do sada neuočenih grešaka i unaprijeđivanje sustava.

Iako su do sada već uvelike doprinjele i sigurnosti u zrakoplovstvu, sustav za kreiranje i prikaz interaktivnih lista provjera je osmišljen kako bi dodatno poboljšao statistiku i smanjio trend zrakoplovnih nesreća uzrokovanih preopterećenjem mentalnih kapaciteta pilota. Opisani model je izrađen na temelju iskustva u generalnom zrakoplovstvu i poznavanja rada organizacije za školovanje pilota ali princip funkcioniranja sustava može se primjeniti u većini organizacija u zrakoplovstvu i prilagoditi zahtjevima pojedinih operatera.

Literatura

[1] https://www.flightsafetyaustralia.com/2018/11/one-thing-at-a-time-a-brief-history-of-the-checklist/, ožujak 2019.

[2] Grant, R.G.: Flight – The Complete History of Aviation, DK, 2017

[3] The Aircraft Book – The Definitive Visual History, DK, 2013

[4] https://www.easa.europa.eu/document-library/regulations/, travanj 2019.

[5] www.fpz.unizg.hr/zan, ožujak 2019.

[6] www.w3schools.com, ožujak 2019.

[7] Achour, Mehdi; Betz, Friedhelm; Dovgal, Antony; Lopes, Nuno; Magnusson, Hannes; Richter, Georg; Seguy, Damien; Vrana, Jakub: PHP Manual, the PHP Documentation Group, 2019

[8] https://www.skybrary.aero/index.php/Electronic_Centralized_Aircraft_Monitor_(ECAM), travanj 2019

Popis slika

Slika 1: Lista provjere za zrakoplov Cessna 172R [5] 5
Slika 2: Primjer predetaljne liste provjere [5] 6
Slika 3: Primjer jednostavne HTML stranice 9
Slika 4: Razvojno okruženje programa Notepad++ 13
Slika 5: Dijagram toka korištenja sustava 14
Slika 6: Sučelje sustava za izradu lista provjera 15
Slika 7: Sučelje sustava za odrađivanje lista provjera 17
Slika 8: Struktura web stranice 19
Slika 9: Dijagram toka izrade liste provjere 21
Slika 10: Programska logika pojavljivanja gumba za završetak liste provjere 23

Dodatak

Dio kôda sustava za kreiranje i prikaz interaktivnih lista provjere

```
<?php


if(!empty($_POST['delete-cl-submit'])) {

$deletechecklist = $_POST['deletechecklist'];

$deletedFormat = "";
$str=file_get_contents('cllist.php');
$str=str_replace("<option value=\"". $deletechecklist ."\">".
$deletechecklist ."</option>", "$deletedFormat",$str);
file_put_contents('cllist.php', $str);

$ac = substr($deletechecklist, 0, 6);
$name = substr($deletechecklist, 7);

$str=file_get_contents($ac .'.php');
$str=str_replace("<a href='#dropdown". $name ."' data-toggle='collapse' ><h3
id='". $name ."'  style='margin: auto; text-align:center; width: 50%'>".
$name ."</h3></a>", "$deletedFormat",$str);
file_put_contents($ac .'.php', $str);

}


if(!empty($_POST['delete-ac-submit'])) {

    $deleteaircraft = $_POST['deleteaircraft'];

unlink($deleteaircraft .'.php');
unlink($deleteaircraft .'-n.php');
unlink($deleteaircraft .'-e.php');


$deletedFormat = "";
$str=file_get_contents('acoptions.php');
$str=str_replace("<option value=\"". $deleteaircraft ."\">". $deleteaircraft
."</option>", "$deletedFormat",$str);
file_put_contents('acoptions.php', $str);

$str=file_get_contents('aircraft.php');
$str=str_replace("<li role='presentation'><a href='". $deleteaircraft
.".php'>". $deleteaircraft ."</a></li>", "$deletedFormat",$str);
file_put_contents('aircraft.php', $str);

}

if(!empty($_POST['ac-submit'])) {
    $newac = $_POST['newac'];
```

```
    $handle = fopen("aircraft.php", "a");
    fwrite($handle, "<li role='presentation'><a href='" . $newac . ".php'>" .
$newac . "</a></li>");
    fclose($handle);


    $handle1 = fopen($newac ."-n.php", "a");
    fclose($handle1);

    $handle2 = fopen($newac ."-e.php", "a");
    fclose($handle2);

    $handle4 = fopen("acoptions.php", "a");
    fwrite($handle4, "<option value=\"". $newac ."\">". $newac ."</option>");
    fclose($handle4);

    $handle3 = fopen($newac .".php", "a");
    fwrite($handle3, "


<!DOCTYPE html>
<html>

<head>
    <meta charset='utf-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1.0'>
    <title>HZNS Checklists</title>
    <link rel='stylesheet' href='assets/bootstrap/css/bootstrap.min.css'>
    <link rel='stylesheet'
href='https://fonts.googleapis.com/css?family=Cookie'>
    <link rel='stylesheet'
href='https://fonts.googleapis.com/css?family=Source+Sans+Pro'>
    <link rel='stylesheet' href='assets/css/styles.css'>
    <link rel='stylesheet' href='assets/css/Google-Style-Login.css'>
    <link rel='stylesheet' href='assets/css/Hero-Technology.css'>
    <link rel='stylesheet' href='assets/css/Pretty-Header.css'>
    <link rel='stylesheet' href='assets/css/w3.css'>
</head>

<body>
    <nav class='navbar navbar-default'>
        <div class='container-fluid'>
            <div class='navbar-header'>
                <a class='navbar-brand navbar-link' href='#'> </a>
                <button class='navbar-toggle collapsed' data-
toggle='collapse' data-target='#navcol-1'><span class='sr-only'>Toggle
navigation</span><span class='icon-bar'></span><span class='icon-
bar'></span><span class='icon-bar'></span></button>
            </div>
            <div class='collapse navbar-collapse' id='navcol-1'>
                <ul class='nav navbar-nav'>
                    <li class='active' role='presentation'><a
href='landing.php'>Checklists </a></li>
                    <li role='presentation'><a href='settings.php'>Settings
</a></li>
                </ul>
            </div>
        </div>
```

```
    </nav>


    <div class='w3-row'>

        <div class='w3-half' style='margin: auto; text-align:left; width:
50%'>

                <div style='float: right; padding-right: 10%'>

                    <div class='dropdown'>
                        <button class='btn btn-default dropdown-toggle' data-
toggle='dropdown' aria-expanded='false' type='button'>  $newac   
 <span class='caret'></span></button>
                        <ul class='dropdown-menu' role='menu'>

                            <?php include 'aircraft.php'; ?>

                        </ul>
                    </div>
                </div>
        </div>

        <div class='w3-half' style='margin: auto; text-align:left; width:
50%'>

            <ul  style='float: left; padding-left: 10%; list-style-type:
none;'>
                <li style='height: 40px; '><a  style='color: green; cursor:
pointer' onclick='normal()'>&#9992 NORMAL</a></li>
                <li style='height: 40px; '><a  style='color: red; cursor:
pointer' onclick='emergency()'>&#9992 EMERGENCY</a></li>
            <ul>

                <script>
                function normal()
                {
                    document.getElementById('normal').style.display
='inherit';
                    document.getElementById('emergency').style.display
='none';
                }
                </script>
                <script>
                function emergency()
                {
                    document.getElementById('normal').style.display = 'none';
                    document.getElementById('emergency').style.display
='inherit';
                }
                </script>

        </div>

    </div>

    <meta charset='UTF-8'>
```

```
<hr>
<div class='w3-row'>

        <div class='w3-col'>

            <div id='normal' style='display: inherit'>
                <?php include '$newac-n.php'; ?>
            </div>

            <div id='emergency' style='display: none'>
                <?php include '$newac-e.php'; ?>
            </div>

        </div>
</div>




    <script src='assets/js/jquery.min.js'></script>
    <script src='assets/bootstrap/js/bootstrap.min.js'></script>
    <script src='assets/js/table.js'></script>
</body>

</html>



    ");

    fclose($handle3);


    }


if(!empty($_POST['cl-submit'])) {
    $aircraft = $_POST['aircraft'];
    $type = $_POST['type'];
    $clname = $_POST['clname'];
    $counter = $_POST['counter'];

    if ($type == 'Normal') {$type = '-n';} else {$type = '-e';};

    $handle = fopen($aircraft . $type .".php", "a");
    fwrite($handle, "

        <a href='#dropdown".$clname."' data-toggle='collapse' ><h3 id='".
$clname ."'  style='margin: auto; text-align:center; width: 50%'>". $clname
."</h3></a>
        <div id='dropdown".$clname."' class='collapse' >
```

```
        <br><table id='". $clname ."-table' style='margin: auto; text-
align:left; width: 40%'>
        <form action'' method='POST'>

    ");


        for ($j = 1; $j <= $counter ; $j++) {
        $item = $_POST[$j];

        fwrite($handle, "
        <tr><td><input type='button' id='". $clname . $j ."' class='toggle'
value='". $item ."' style='height: 40px; width: 100%; padding-left: 30px;
text-align:left'
        onclick= \" this.style.opacity = '0.6'; value='". $item ."  
    &#x2714'; skripta".$clname."(); disabled=true \"></td></tr>
        "); }


    fwrite($handle, "

                                <tr><td><br></td></tr>
                                <tr><td><input type='button' id='". $clname
."finish' style=' visibility: collapse' value='Finish Checklist'
                                onclick='check". $clname ."()'></td></tr>

                                <script>
                                    window.check". $clname ." = function() {
                                    document.getElementById('". $clname
."').innerHTML= '". $clname ."       &#x2714';
                                    document.getElementById('". $clname
."finish').style.visibility= 'collapse';}
                                </script>

                                <script>

                                var counter".$clname."=0;
                                window.skripta".$clname." = function() {
                                counter".$clname."++;

if(counter".$clname."==".$counter."){document.getElementById('".$clname."fini
sh').style.visibility= 'visible';};
                                }

                                </script>

                            </form>
                            </table>
                    </div>

                    <br>

    ");
    fclose($handle);


    $handle1 = fopen("cllist.php", "a");
```

```
    fwrite($handle1, "<option value=\"". $aircraft . $type ." ". $clname
."\">". $aircraft . $type ." ". $clname ."</option>");
    fclose($handle1);

}

?>


<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HZNS Checklists</title>
    <link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Cookie">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro">
    <link rel="stylesheet" href="assets/css/styles.css">
    <link rel="stylesheet" href="assets/css/Google-Style-Login.css">
    <link rel="stylesheet" href="assets/css/Hero-Technology.css">
    <link rel="stylesheet" href="assets/css/Pretty-Header.css">
    <link rel="stylesheet" href="assets/css/w3.css">
</head>

<body>
    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand navbar-link" href="#"> </a>
                <button class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navcol-1"><span class="sr-only">Toggle
navigation</span><span class="icon-bar"></span><span class="icon-
bar"></span><span class="icon-bar"></span></button>
            </div>
            <div class="collapse navbar-collapse" id="navcol-1">
                <ul class="nav navbar-nav">
                    <li role="presentation"><a href="landing.php">Checklists
</a></li>
                    <li class="active" role="presentation"><a
href="settings.php">Settings </a></li>
                </ul>
            </div>
        </div>
    </nav>


    <div class="w3-row">

        <div class="w3-half" style="margin: auto; text-align:left; width:
50%">

                <div style="float: right; padding-right: 10%">
```

```
                </div>
        </div>

        <div class="w3-col" >
            <br>
            <ul  style="margin:auto; width: 200px; list-style-type: none;">
                <li style="height: 40px; "><a onclick="addaircraft()"
style="cursor: pointer">&#9992 ADD AIRCRAFT</a></li>
                <li style="height: 40px; "><a onclick="addchecklist()"
style="cursor: pointer">&#9992 ADD CHECKLIST</a></li>
                <li style="height: 40px; "><a onclick="deleteaccl()"
style="cursor: pointer">&#9992 DELETE AC / CL</a></li>
            <ul>

                <script>
                function addaircraft()
                {
                    document.getElementById("addaircraft").style.display
="inherit";
                    document.getElementById("addchecklist").style.display
="none";
                    document.getElementById("deleteaccl").style.display
="none";
                }
                </script>
                <script>
                function addchecklist()
                {
                    document.getElementById("addaircraft").style.display =
"none";
                    document.getElementById("addchecklist").style.display
="inherit";
                    document.getElementById("deleteaccl").style.display
="none";
                }
                </script>
                <script>
                function deleteaccl()
                {
                    document.getElementById("addaircraft").style.display =
"none";
                    document.getElementById("addchecklist").style.display
="none";
                    document.getElementById("deleteaccl").style.display
="inherit";
                }
                </script>

        </div>

    </div>

    <meta charset="UTF-8">


<hr>
<div class="w3-row">
```

```
        <div class="w3-col">

            <div id="addaircraft">

                <form action"" method="POST">
                <table style="margin: auto; width: 200px">

                    <tr>
                    <td><input name="newac" type="text" maxlength="4"
placeholder="Aircraft Type" name="aircraft" required /></td>
                    <td>   </td><td>   </td>
                    <td><input type="submit"  name="ac-submit" value="Add
Aircraft"></td>
                    </tr>
                    <tr><td>  </td></tr>
                    </td></tr>

                </table>
                </form>

            </div>

            <div id="addchecklist" style="display: none">

                <form action"" method="POST">
                    <table style="margin: auto; width: 300px">

                        <tr>
                        <td>
                        <select style="width: 100px" name="aircraft"
required>
                            <?php include "acoptions.php"; ?>
                        </select>
                        </td>

                        <td>   </td><td>   </td>
                        <td>
                        <select style="width: 120px" name="type" required>
                            <option value="Normal">Normal</option>
                            <option value="Emergency">Emergency</option>
                        </select>
                        </td></tr>

                        <tr><td>  </td></tr>
                    </table>

                    <table id="tablica" style="margin: auto; width: 300px">
                        <tr><td><input type="text"  name="clname"
placeholder="Checklist name"></td></tr>
                        <tr><td><br></td></tr>
                        <tr><td><input type="text"  name="1"
placeholder="Item1"></td></tr>
                    </table><br>

                    <table style="margin: auto; width: 300px">

                        <tr>
```

```
                        <td><button type="button" value="additem"
style="width: 120px" onclick="additem()">Add Item</button></td>

                        <td><input type="submit"  name="cl-submit" value="Add
Checklist" ></td></tr>
                        <tr><td>   </td></tr>
                        <tr><td><button type="button" value="removeitem"
style="width: 120px" onclick="removeitem()">Remove Item</button></td></tr>

                    </table>
                    <table>

                        <tr><td><input style="visibility:hidden" type="text"
name="counter" id="counter" value="" ></td><td>  </td></tr>
                    </table>

                    <script>
                    i=1;

                    function additem() {
                         i=i+1;
                      var table = document.getElementById("tablica");
                      var row = table.insertRow(-1);
                      var cell1 = row.insertCell(0);
                      cell1.innerHTML = "<input type='text' name='"+ i +"'
placeholder='Item"+ i +"'>";
                      document.getElementById("counter").value = i;

                    }

                    function removeitem() {
                        i=i-1;
                        var table = document.getElementById("tablica");
                        var row = table.deleteRow(-1);
                        document.getElementById("counter").value = i;
                    }

                    </script>

                </form>

            </div>


            <div id="deleteaccl" style="display: none">

            <form action"" method="POST">
                <table style="margin: auto; width: 200px">

                    <tr>
                    <td>
                    <select style="width: 100px" name="deleteaircraft"
required>
                        <?php include "acoptions.php"; ?>
                    </select>
                    </td><td>  </td><td>  </td>
                    <td><input type="submit"  name="delete-ac-submit"
value="Delete Aircraft"></td>
```

```
                    </tr>
                    <tr><td>  </td></tr>
                    </td></tr>

                </table>
                </form>


                <form action"" method="POST">
                <table style="margin: auto; width: 200px">

                    <tr>
                    <td>
                    <select style="width: 200px" name="deletechecklist"
required>
                        <?php include "cllist.php"; ?>
                    </select>
                    </td><td>  </td><td>  </td>
                    <td><input type="submit"  name="delete-cl-submit"
value="Delete Checklist"></td>
                    </tr>
                    <tr><td>  </td></tr>
                    </td></tr>

                </table>
                </form>



            </div>

        </div>
</div>




    <script src="assets/js/jquery.min.js"></script>
    <script src="assets/bootstrap/js/bootstrap.min.js"></script>
    <script src="assets/js/table.js"></script>
</body>

</html>
```

Sveučilište u Zagrebu
Fakultet prometnih znanosti
Vukelićeva 4, 10000 Zagreb

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOSTI

Izjavljujem i svojim potpisom potvrđujem da je završni rad isključivo rezultat mojega vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju upotrijebljene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu završnog rada pod naslovom "Izrada sustava za kreiranje i prikaz interaktivnih lista provjere za letačke operacije", na mrežnim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Studentica:

U Zagrebu 06.04.2019