

Podporni informacijski sistem

Strokovno poročilo

Mentor: Darjan Toth, prof.

Avtor: Nejc Vrčon Zupan, R 4. B

Ljubljana, marec 2020

Povzetek

V poročilu bom predstavil podatkovno bazo in spletno aplikacijo, katera zagotavlja informacijsko podporo sejmu rabljenih izdelkov. Poročilo zajema podroben opis podatkovne baze, spletne aplikacije in njene uporabnosti, kot rešitev težav v praksi. Prav tako bom omenil težave, na katere sem naletel pri razvijanju podatkovne baze in aplikacije, ter možnih izboljšavah končnega izdelka.

Ključne besede: Firebird, PHP, SQL, HTML, PSQL

Abstract

The purpose of this paper is to report on the development of my database and web application which together create my support information system. The report describes the database and web application as an example of a useful solution that can be used in the real world. I also talk about the problems, which I encountered while developing my information system and possible improvements one could make to the end product.

Key words: Firebird, PHP, SQL, HTML, PSQL

Kazalo

1. UVOD	5
2. CILJI	5
2.1 Uporabljene tehnologije	5
2.1.1 PHP	5
2.1.2 HTML	6
2.1.3 CSS	6
2.1.4 Firebird	7
2.1.5 XAMPP	7
3. SHEMA POTEKA SEJMA.....	8
4. SHEMA PODATKOVNE BAZE IN TABEL	10
4.1 Tabela Member	10
4.2 Tabela Staff	10
4.3 Tabela Ptype	10
4.4 Tabela Product.....	10
4.5 Tabela Event.....	11
4.6 Shema podatkovne baze	11
4.7 Shranjene procedure.....	12
4.7.1 Procedura CASH_LOG	12
4.7.2 Procedura CASH_SUM	12
4.7.3 Procedura FIND_PRODUCT	12
4.7.4 Procedura GEN_MEM	12
4.7.5 Procedura INPUT_EVENT	12
4.7.6 Procedura INPUT_PRODUCT	13
4.7.7 Procedura INPUT_PTYPE	13
4.7.8 Procedura INPUT_STAFF	14
4.7.9 Procedura PP_FLOW	14
4.7.10 Procedura PTYPE_OUT	14
4.7.11 Procedura OUT_FLOW.....	14
4.7.12 Procedura STAFF_LOGIN	16

4.7.13	Procedura FIND_STAFF	16
4.7.14	Procedura SHOW_BASKET	16
4.8	Sprožilci	18
4.8.1	Sprožilec IP_LOG	18
4.8.2	Sprožilec E_CHK	18
4.9	Pomožna skripta	20
5.	SPLETNA APLIKACIJA	20
5.1	Zakaj spletna aplikacija	20
5.2	Povezovanje z bazo v PHP	21
5.3	Videz spletne aplikacije	21
5.4	Generiranje dokumentov	22
6.	UPORABA INFORMACIJSKEGA SISTEMA	23
7.	MOŽNE IZBOLJŠAVE	24
8.	ZAKLJUČEK	24
9.	VIRI	25
9.1	Citati	25

1. UVOD

V poročilu je opisan izdelek, ki sem ga izdelal za poklicno maturo. Izdelek je namenjen informacijski podpori sejma rabljenih izdelkov. Realiziran je s podatkovno bazo Firebird in spletno aplikacijo, napisano v jeziku PHP. Izdelek ima poudarek predvsem na podatkovni bazi, natančneje shranjenih procedurah in sprožilcih. Idejo za izdelek sem dobil med opravljanjem PUD, saj sem v sklopu PUD pomagal pri razvoju informacijskega sistema.

Opisal in predstavil bom shemo podatkovne baze, spletno aplikacijo, procedure, ki so potrebne za delovanje oz. praktično uporabo izdelka in težave na katere sem naletel pri razvijanju podatkovne baze in aplikacije, ter možne izboljšave končnega izdelka.

2. CILJI

Pred samim pisanjem programske kode in načrtovanjem baze sem si postavil nekaj ciljev. To so:

- zbrati podatke za artikel in jih vpisati v podatkovno bazo,
- ustvariti račune in dokazila, s katerimi udeleženci dokažejo lastništvo,
- omogočiti samodejni izračun razlike v ceni pri košarici udeleženca,
- beležiti vse pomembne dogodke v bazo in tako ustvariti dnevnik oz. log,
- izdelati spletno aplikacijo, ki bo služila kot vmesnik med bazo in uporabnikom,
- izdelati uporaben in intuitiven izdelek, ki bo služil svojemu namenu.

2.1 Uporabljene tehnologije

2.1.1 PHP

PHP je programski jezik, ki ga uporabljamo za pisanje spletnih aplikacij. PHP je skriptni jezik, ki se izvaja na strežniški strani in za njegovo funkcionalnost potrebujemo spletni strežnik. Strežnik ima vgrajen tolmač, ki prevede kodo in jo pošlje brskalniku v obliki

HTML kode. S PHP-jem lahko pregledujemo podatke, jih spreminjamo, šifriramo, se povežemo na podatkovno bazo s spletno stranjo ter oblikujemo dinamično spletno aplikacijo.

```
<?
....//PHP

....$jezik = 'PHP';
....echo 'To je napisano v jeziku '.$i.'!';
?>
```

Slika 1: Primer kode programskega jezika PHP (lasten vir)

2.1.2 HTML

HTML je označevalni jezik, ki se uporablja za izdelovanje spletnih strani. Pogosto ga najdemo uporabljenega v kombinaciji z PHP-jem oziroma CSS-om, ki skrbi za postavitev in izgled HTML – elementov. HTML ukaze prepoznamo po elementih: večje, manjše in poševnica (primeri: <p> </p>).

```
<!DOCTYPE html>
<html>
<body>
....<h1>HTML primer</h1>
....<span>To je napisano v jeziku HTML</span>
</body>
</html>
```

Slika 2: Primer kode jezika HTML

2.1.3 CSS

CSS je jezik, ki ga uporabljamo v kombinaciji z HTML. Je jezik oblikovanja spletnih strani, torej HTML elementom doda oblikovne lastnosti.

```

/*css*/
html{
  background-color: #brown;
}

h1{
  font-size: 30px;
  color: #blueviolet;
  text-align: center;
}

span{
  font-size: 12px;
  color: #aqua;
  text-align: center;
}

```

Slika 3: Primer kode jezika CSS (lasten vir)

2.1.4 Firebird

Firebird je odprtokodni sistem za upravljanje s podatkovnimi bazami. Jezik SQL je programski jezik 4. generacije, ki se uporablja za delo s podatki.

```

SELECT *
FROM Table
WHERE name='sql';

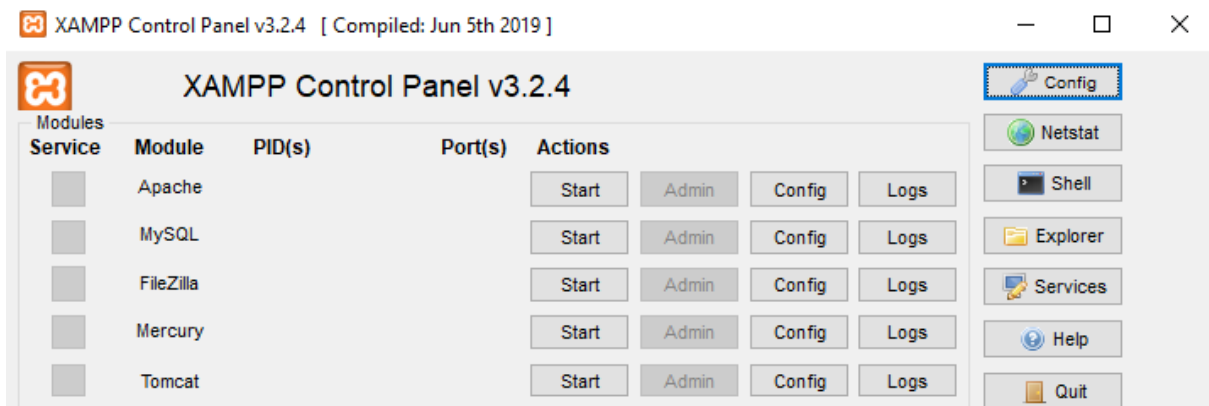
```

Slika 4: Primer kode jezika SQL (lasten vir)

2.1.5 XAMPP

Xampp je brezplačno in odprtokodno programsko – razvojno okolje, ki je zaradi svoje kompleksnosti in preprostega postopka namestitve zelo razširjen med PHP razvijalci in

razvijalci podatkovnih baz. Del orodja Xampp vsebuje Apache server in programske tolmače (angl. Interpreter), ki kodo prevedejo in jo pošljejo brskalniku v HTML obliki.

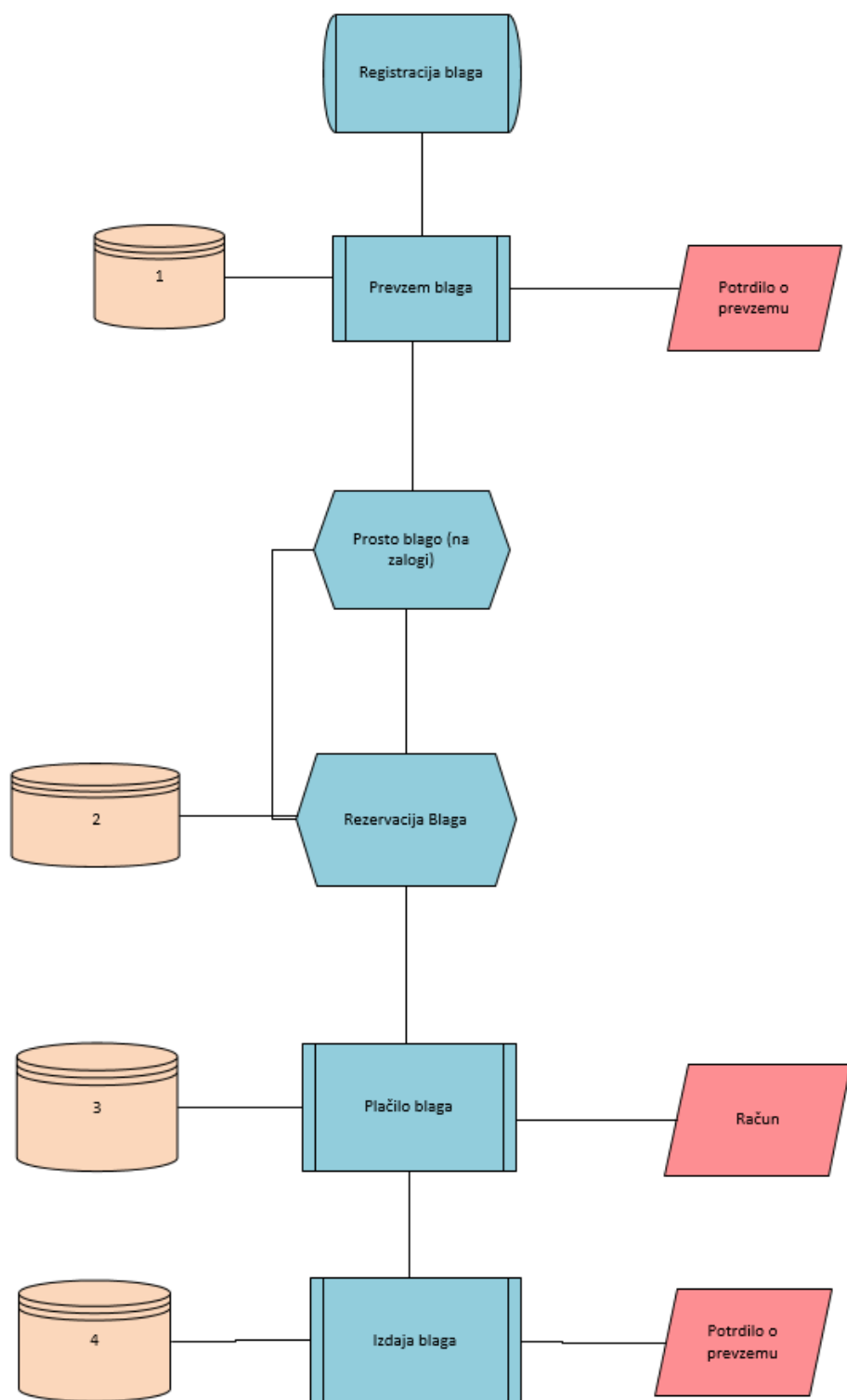


Slika 5: Nadzorna plošča orodja Xampp (lasten vir)

3. SHEMA POTEKA SEJMA

Pred načrtovanjem podatkovne baze in spletne aplikacije, sem se najprej lotil analiziranja problema. Ravnal sem se po citatu: "**Prvo reši problem. Nato napiši kodo.**" (Johnson, 2009).

Modri liki predstavljajo pomemben dogodek na sejmu. Rdeči liki predstavljajo dokument, ki se generira, ko artikel doseže ta dogodek. Oranžni liki predstavljajo zapis v podatkovno bazo (tabelo Event). Shema mi je pomagala pri načrtovanju baze in načrtovanju spletne aplikacije.



Slika 6: Shema poteka sejma (lasten vir)

4. SHEMA PODATKOVNE BAZE IN TABEL

Podatkovno bazo, katera je napisana v jeziku SQL sem se odločil goniti v orodju Firebird, saj je sintaksa proceduralnega SQL-a veliko boljša v primerjavi z MySQL.

4.1 Tabela Member

Ena vrstica v tabeli Member predstavlja enega udeleženca. Tabela vsebuje eno polje MemberID tipa varchar(5). Identifikacijska številka (ID), ki jo naključno generira baza, služi sledenju lastništva oz. interakcij z bazo.

4.2 Tabela Staff

Ena vrstica v tabeli Staff predstavlja enega blagajnika oz. osebo z dostopom do baze in spletne aplikacije. Tabela vsebuje polja:

- StaffID varchar(5), ID, ki ga naključno generira baza služi sledenju interakcij,
- S_name varchar(45), ime blagajnika,
- S_surname varchar(200), priimek blagajnika.

4.3 Tabela Ptype

Ena vrstica v tabeli Ptype predstavlja en tip izdelka. Tabela vsebuje polja:

- PtypeID varchar(5), ID, ki ga naključno generira baza služi sledenju interakcij,
- T_Name varchar(50), ime tipa izdelka,
- T_Descr varchar(200), opis tipa izdelka,
- T_commission float, stopnja provizije,
- Tax float, davčna stopnja.

4.4 Tabela Product

Ena vrstica v tabeli Product predstavlja en artikel. Tabela vsebuje polja:

- ProductID varchar(5), ID, ki ga naključno generira baza služi sledenju interakcij,
- P_Name varchar(50), ime izdelka,

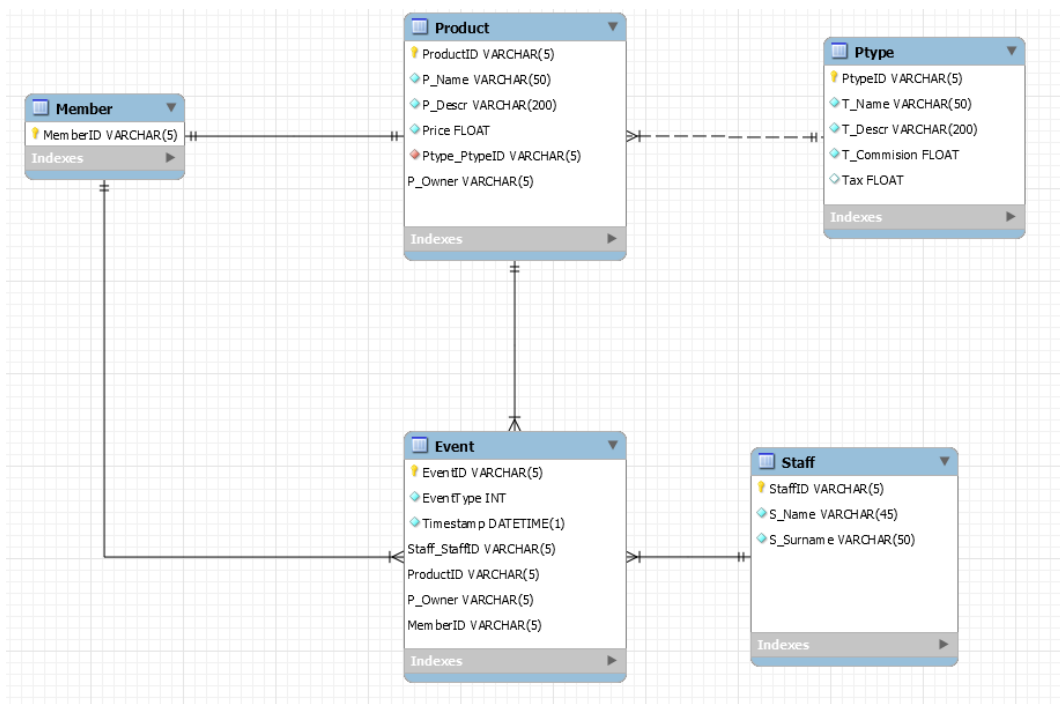
- P_Descr varchar(200), opis izdelka,
- Price float, cena izdelka,
- PtypeID varchar(5), ID tipa izdelka (tabela Ptype),
- P_Owner varchar(5), ID lastnika izdelka (tabela Member).

4.5 Tabela Event

Ena vrstica v tabeli Event predstavlja eno beleženje dogodka. Tabela vsebuje polja:

- EventID varchar(5), ID, ki ga naključno generira baza (številka transakcije).
- Eventtype int, tip transakcije.
- Time_Stamp TIMESTAMP, čas transakcije.
- StaffID varchar(5), ID uslužbenca (tabela Staff).
- ProductID varchar(5), ID izdelka (tabela Product).
- P_Owner varchar(5), ID lastnika izdelka (tabela Member).
- MemberID varchar(5), ID kupca (tabela Member).

4.6 Shema podatkovne baze



Slika 7: ER model podatkovne baze (lasten vir)

4.7 Shranjene procedure

Podatkovna baza ima 14 shranjenih procedur. Shranjene procedure zagotavljajo pravilne izpise podatkov, ki jih potrebuje uporabnik, večjo varnost in hitrejšo delovanje baze. Skrbijo tudi za pravilne vnose v podatkovno bazo, ter izvajanje poslovnih pravil.

4.7.1 Procedura CASH_LOG

Procedura CASH_LOG izpiše vse denarne transakcije in jih poveže s fizičnimi osebami, lastnik, kupec in blagajnik. Služi predvsem v analitične namene.

4.7.2 Procedura CASH_SUM

Procedura CASH_SUM izpiše, koliko je bilo na sejmu skupno denarnega prometa, prihodkov (iz provizij) in plačanega davka. Služi predvsem v analitične namene.

4.7.3 Procedura FIND_PRODUCT

Procedura FIND_PRODUCT izpiše ID izdelka, ime izdelka, ceno izdelka in ID lastnika izdelka na podlagi ID-ja izdelka. Ima splošen namen in se veliko uporablja.

4.7.4 Procedura GEN_MEM

Procedura GEN_MEM naključno generira x število vpisov v tabelo Member. Ker udeležence na sejmu zaznamuje samo MemberID v tabeli Member jih lahko vnaprej naključno generiramo in tako zmanjšamo čakalne vrste.

4.7.5 Procedura INPUT_EVENT

Procedura INPUT_EVENT dodaja zapise v tabelo Event. V tabeli Event se beležijo vsi pomembni dogodki.

```

--INPUT_EVENT PROCEDURA
SET TERM !! ;
CREATE PROCEDURE input_EVENT (tip int, staff VARCHAR(5), kupec VARCHAR(5), izdelek VARCHAR(5)) RETURNS (Izpis char(50)) AS
DECLARE VARIABLE lastnik VARCHAR(5);
DECLARE VARIABLE cas TIMESTAMP;
DECLARE VARIABLE cash_kupec VARCHAR(5);
BEGIN
:staff=UPPER(:staff);
:kupec=UPPER(:kupec);
:izdelek=UPPER(:izdelek);

--SELECT ZA ISKANJE lastnika izdelka na podlagi id izdelka
FOR
SELECT p.P_Owner
FROM Product p
WHERE p.productID=:izdelek
INTO :lastnik
DO BEGIN
END

IF(:tip=4) THEN
BEGIN
--SELECT ZA ISKANJE kupca pri prodaji izdelka na podlagi id izdelka
FOR
SELECT e.memberID
FROM Event e
WHERE e.productID=:izdelek AND e.eventtype=3
INTO :cash_kupec
DO BEGIN
END
END

:Izpis = 'Dodajanje neuspesno';
IF ((:tip=1 AND :kupec = :lastnik) OR (:tip=4 AND :kupec = :cash_kupec) OR :tip=2 OR :tip=3) THEN
BEGIN
--INSERT V TABELO EVENT
INSERT INTO Event
VALUES(SUBSTRING(UUID_TO_CHAR(GEN_UUID())) FROM 1 FOR 5), :tip, CURRENT_TIMESTAMP, :staff, :kupec, :izdelek, :lastnik);
--CE LASTNIK IZDELEK PREVZAME NAZAJ SETAMO CENO Izdelka na 0
IF(:tip=3 AND :lastnik=:kupec) THEN
BEGIN
UPDATE Product
SET Price=0
WHERE productid=:izdelek;
END
:Izpis='Dodajanje uspesno';
END
END !!
set TERM ; !!

```

Slika 8: Procedura INPUT_EVENT (lasten vir)

4.7.6 Procedura INPUT_PRODUCT

Procedura INPUT_PRODUCT dodaja izdelke v podatkovno bazo (tabela Product).

4.7.7 Procedura INPUT_PTYPE

Procedura INPUT_PTYPE dodaja tipe izdelkov v podatkovno bazo (tabela Ptype).

4.7.8 Procedura INPUT_STAFF

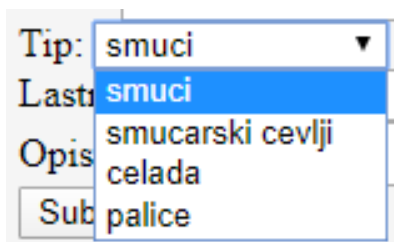
Procedura INPUT_STAFF dodaja uporabnike v podatkovno bazo (tabela Staff).

4.7.9 Procedura PP_FLOW

Procedura PP_FLOW izpiše število aktivnih udeležencev in število izdelkov na sejmu. Služi predvsem v analitične namene.

4.7.10 Procedura PTYPE_OUT

Procedura PTYPE_OUT izpiše vse tipe izdelkov. Uporablja se pri vnašanju izdelkov in omogoča uporabniku izbiranje med tipi izdelkov.



Slika 9: Uporaba procedure PTYPE_OUT (lasten vir)

4.7.11 Procedura OUT_FLOW

Procedura OUT_FLOW izpiše izdelke udeleženca sejma (MemberID), ki so pripravljeni na izdajo.

```

--Out_flow
SET TERM !! ;
CREATE PROCEDURE Out_flow(mID varchar(5))
RETURNS(id VARCHAR(5), ime VARCHAR(50)) AS
DECLARE VARIABLE tmp VARCHAR(5);
DECLARE VARIABLE zmp VARCHAR(5);
DECLARE VARIABLE flag int;
BEGIN
    :mID=UPPER(:mID);
    :flag=0;

    FOR
        SELECT e.productid
        FROM Event e
        INNER JOIN Product p
        ON p.productID=e.productID
        WHERE e.eventtype=3 AND e.memberID=:mid AND p.P_Owner!:=mid
    INTO :tmp
    DO BEGIN
        FOR
            SELECT e.productID
            FROM Event e
            WHERE e.eventtype=4 AND e.productID=:tmp
            INTO :zmp
            DO BEGIN
                :flag=1;
            END

            IF(:flag=0) THEN BEGIN
                FOR
                    SELECT p.productid,p.P_Name
                    FROM Product p
                    WHERE p.productid=:tmp
                    INTO :id, :ime
                    DO BEGIN
                        SUSPEND;
                    END
                END
            END
        END
    END
END!!
SET TERM ; !!

```

Slika 10: Procedura OUT_FLOW (lasten vir)

4.7.12 Procedura STAFF_LOGIN

Procedura STAFF_LOGIN vrne ID, ime in priimek blagajnika s pomočjo uporabniškega imena. Procedura se uporablja pri prijavi uporabnika v sistem.

4.7.13 Procedura FIND_STAFF

Procedura FIND_STAFF vrne ID blagajnika oz. uslužbenca. S pomočjo sistemske spremenljivke v bazi CURRENT_USER izpiše ID trenutno prijavljenega uporabnika oz. blagajnika.

```
--find_staff
SET TERM !! ;
CREATE PROCEDURE Find_staff
RETURNS (id VARCHAR(5)) AS
DECLARE VARIABLE uname char(31);
BEGIN
--dobimo username
FOR
SELECT FIRST 1 sec$user_name
FROM SEC$USERS
WHERE sec$user_name=CURRENT_USER
INTO :uname
DO BEGIN
END

--dobimo id
FOR
SELECT FIRST 1 StaffID
FROM Staff
WHERE UPPER(SUBSTRING(S_username FROM 1 FOR 30))=:uname OR SUBSTRING(S_username FROM 1 FOR 30)=:uname
INTO :id
DO BEGIN
| SUSPEND;
END
END!!
SET TERM ; !!
```

Slika 11: Procedura FIND_STAFF (lasten vir)

4.7.14 Procedura SHOW_BASKET

Procedura SHOW_BASKET prikaže košarico udeleženca sejma. V košarici so artikli, ki jih je udeleženec rezerviral in jo mora plačati. V primeru, da je rezerviral svoj artikel se cena postavi na 0. V košarici so prisotni artikli, katerih udeleženec je lastnik in so bili že prodani. V tem primeru je udeležencu potrebno izplačati denar.


```

SET TERM !! ;
CREATE PROCEDURE show_basket(mID VARCHAR(5))
RETURNS(id VARCHAR(5), ime VARCHAR(50), cena float, provizija float, davek float, cas int) AS
BEGIN
    :mID=UPPER(:mID);
    --SELECT ZA ISKANJE izdelkov lastnika ki so bili ze placani
    FOR
        SELECT p.productID, p.P_Name, p.Price*(-1), pt.T_commission, pt.tax
        FROM Member m
        INNER JOIN Product p
        ON p.P_Owner=m.memberID
        INNER JOIN Event e
        ON p.productID=e.productID
        INNER JOIN Ptype pt
        ON p.PtypeID=pt.ptypeID
        WHERE p.P_Owner=:mID AND e.eventtype=3 AND p.productID NOT IN(
            SELECT p.productID
            FROM Member m
            INNER JOIN Product p
            ON p.P_Owner=m.memberID
            INNER JOIN Event e
            ON p.productID=e.productID
            WHERE p.P_Owner=:mID AND e.memberID=:mID AND e.eventtype=3
        )
    INTO :id, :ime, :cena, :provizija, :davek
    DO BEGIN
        SUSPEND;
    END
    --SELECT ZA ISKANJE izdelkov kupca ki so rezervirani
    FOR
        SELECT p.productID, p.P_Name, p.Price, pt.T_commission, pt.tax, datediff(minute from e.time_stamp to CURRENT_TIMESTAMP)
        FROM Member m
        INNER JOIN Event e
        ON e.memberID=m.memberID
        INNER JOIN Product p
        ON e.productID=p.productID
        INNER JOIN Ptype pt
        ON p.PtypeID=pt.ptypeID
        WHERE e.memberID=:mID AND e.eventtype=2 AND p.productID NOT IN(
            SELECT p.productID
            FROM Member m
            INNER JOIN Event e
            ON e.memberID=m.memberID
            INNER JOIN Product p
            ON e.productID=p.productID
            WHERE e.memberID=:mID AND e.eventtype=3
        )
    INTO :id, :ime, :cena, :provizija, :davek, :cas
    DO BEGIN
        SUSPEND;
    END
    --SELECT ZA ISKANJE izdelkov lastnika ki jih хоче prevzeti nazaj v svojo last
    FOR
        SELECT p.productID, p.P_Name, p.Price*(0)
        FROM Member m
        INNER JOIN Product p
        ON p.P_Owner=m.memberID
        INNER JOIN Event e
        ON p.productID=e.productID
        INNER JOIN Ptype pt
        ON p.PtypeID=pt.ptypeID
        WHERE p.P_Owner=:mID AND e.memberID=:mID AND e.eventtype=2 AND p.productID NOT IN(
            SELECT p.productID
            FROM Member m
            INNER JOIN Event e
            ON e.memberID=m.memberID
            INNER JOIN Product p
            ON e.productID=p.productID
            WHERE e.memberID=:mID AND e.eventtype=3
        )
    INTO :id, :ime, :cena
    DO BEGIN
        SUSPEND;
    END
END!!
SET TERM ; !!

```

Slika 12: Procedura SHOW_BASKET (lasten vir)

ID	Ime	Cena	Provizija	Davek	Čas
94039	Elan CARVE X	0,00 €	0,11 %	0,21 %	min
FB94A	testni	100,00 €	0,11 %	0,21 %	19978 min
58362	npb test2	100,00 €	0,11 %	0,21 %	18651 min
94039	Elan CARVE X	0,00 €	0,11 %	0,21 %	18651 min
Vsota: 200,00					
Račun					

Slika 13: Uporaba procedure SHOW_BASKET (lasten vir)

4.8 Sprožilci

Podatkovna baza ima 2 sprožilca. Sprožilci preverjajo na pravilnost vnosov, ter zagotavljajo, da se poslovna pravila pravilno izvajajo.

4.8.1 Sprožilec IP_LOG

Sprožilec IP_LOG, skrbi da se ob vnosu izdelka v tabelo Product izvede procedura INPUT_EVENT, ki v tabelo Event zabeleži dogodek.

```

SET TERM !! ;
CREATE TRIGGER IP_log FOR Product
AFTER INSERT
AS
DECLARE VARIABLE uporabnik_id VARCHAR(5);
DECLARE VARIABLE Izpis CHAR(50);
BEGIN
    --NAJDEMO STAFF_ID
    FOR
        SELECT *
        FROM Find_staff
        INTO :uporabnik_id
    DO BEGIN
        END
    END

    --NAREDIMO LOG Z INPUT_EVENT
    EXECUTE PROCEDURE INPUT_EVENT(1,:uporabnik_id,New.P_Owner,New.ProductID) RETURNING_VALUES(:Izpis);
END!!
SET TERM ; !!

```

Slika 14: Sprožilec IP_LOG (lasten vir)

4.8.2 Sprožilec E_CHK

Sprožilec E_CHK, skrbi za pravilne vnose v tabelo Event in s tem zagotovi pravilno beleženje pomembnih dogodkov.

```

SET TERM !! ;
CREATE TRIGGER E_chk FOR Event
BEFORE INSERT AS
DECLARE VARIABLE flag int;
DECLARE VARIABLE tmp VARCHAR(5);
DECLARE VARIABLE lastnik VARCHAR(5);
BEGIN
--DOBIMO ID LASTNIKA
FOR
    SELECT P_Owner
    FROM Product
    WHERE productID=new.productID
    INTO :lastnik
DO BEGIN
    END

--ET 3 SE ZGODI DVAKRAT (KUPEC KUPI, LASTNIK DOBI)
IF(new.eventtype=3) THEN
BEGIN
    FOR
        SELECT memberID
        FROM Event
        WHERE eventtype=new.eventtype AND productID=new.productID
    INTO :tmp
    DO BEGIN
        IF(:tmp is not null) THEN
        BEGIN
            IF(new.memberid=:lastnik) THEN
            BEGIN
                IF(:tmp=:lastnik) THEN
                BEGIN
                    EXCEPTION EVENT_BUSY;
                END
            END
        END
    END
END
END

--ET 2 SE NE MORE ZGODITI CE JE NAREJEN ZE ET=3
ELSE IF(new.eventtype=2) THEN
BEGIN
    FOR
        SELECT memberID
        FROM Event
        WHERE eventtype IN(2,3) AND productID=new.productID
    INTO :tmp
    DO BEGIN
        IF(:tmp is not null) THEN BEGIN
            EXCEPTION EVENT_BUSY;
        END
    END
END

ELSE BEGIN
    FOR
        SELECT eventid
        FROM Event
        WHERE eventtype=new.eventtype AND productID=new.productID
    INTO :tmp
    DO BEGIN
        IF(tmp is not null) THEN
        BEGIN
            EXCEPTION EVENT_BUSY;
        END
    END
END
END!!
SET TERM ; !!

```

Slika 15: Sprožilec E_CHK (lasten vir)

4.9 Pomožna skripta

Pri implementaciji rezervacije izdelka, sem naletel na manjšo težavo. Izdelke, ki so rezervirani več kot 20 min je potrebno sprostiti nazaj v prodajo. Podatkovna baza, ne omogoča klica neke skripte v obliki sprožilca, funkcije, procedure itd. na neko časovno obdobje. Problem sem rešil tako, da sem napisal .bat skripto, ki se poveže na podatkovno bazo in zažene SQL skripto vsakih 30 sekund.

```
:loop
cd C:\Program Files\Firebird\Firebird_3_0
isql -user sysdba -password masterkey "localhost:c:\baze\sejemv2.fdb" -i "C:\Program Files\Firebird\Firebird_3_0\skripta.sql"
timeout /t 30
goto loop
```

Slika 16: Bat skripta (lasten vir)

5. SPLETNA APLIKACIJA

Spletna aplikacija služi, kot vmesnik med uporabnikom in podatkovno bazo. Spletno aplikacijo sem se odločil napisati v jeziku PHP, saj smo se o njem učili pri predmetu NSA in mi je zato najbolj poznan. Pri pisanju spletne aplikacije sem naletel na težavo pri povezovanju z mojo podatkovno bazo v Firebirdu. Moral sem uporabiti gonilnik in dodati sistemsko spremenljivko v okolje Windows. PHP aplikacijo sem poganjal v Apache strežniku, znotraj orodja Xampp.

5.1 Zakaj spletna aplikacija

Za spletno aplikacijo sem se odločil, ker ima mnogo prednosti pred programom, ki se izvaja na lokalnem računalniku. Spletna aplikacija je neodvisna od operacijskega sistema in nastavitvev operacijskih sistemov. Prav tako zahteva manj administracijskega dela, saj jo lahko enostavno posodobimo.

5.2 Povezovanje z bazo v PHP

Za povezovanje sem uporabil knjižnico ibase. V spremenljivko \$dbh sem shranil parametre s katerimi se bom uspešno povezal na bazo. Uporabniško ime in geslo sem shranil v spremenljivko seje (\$_SESSION), lokacijo strežnika pa v piškotek (\$_COOKIE). V spremenljivko \$stmt sem sestavil in shranil SQL stavek. Nato sem ga izvedel s funkcijo ibase_query(\$dbh, \$stmt) in v while zanki pridobil podatke iz baze. Podatke iz baze prejmem v \$row spremenljivko.

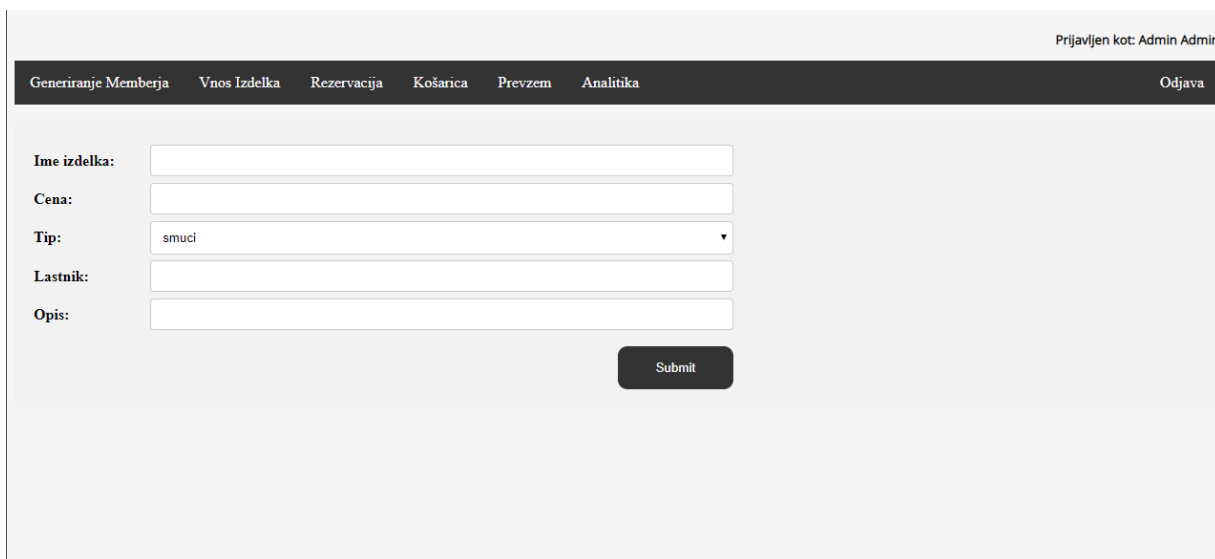
```
$dbh = ibase_connect($_COOKIE['host'], $_SESSION['username'], $_SESSION['pass']);
$stmt = "SELECT StaffID,S_Name,S_Surname,S_Username FROM Staff";

$stmt = ibase_query($dbh, $stmt) or die(ibase_errmsg());
echo '<table><tr><th>ID</th><th>Ime</th><th>Priimek</th><th>Username</th></tr>';
while ($row = ibase_fetch_object($sth)) {
```

Slika 17: Povezava s podatkovno bazo (lasten vir)

5.3 Videz spletne aplikacije

Celotna spletna aplikacija je osnovana na celostni grafični podobi. Njen videz je preprost in minimalističen. Spletna aplikacija je lahka in intuitivna za uporabo.



The screenshot shows a web application interface. At the top right, it says "Prijavljen kot: Admin Admin". Below this is a navigation bar with links: "Generiranje Memberja", "Vnos Izdelka", "Rezervacija", "Košarica", "Prevzem", "Analitika", and "Odjava". The main content area has a form for adding a product. The form includes labels and input fields for "Ime izdelka:", "Cena:", "Tip:" (with a dropdown menu showing "smuci"), "Lastnik:", and "Opis:". A "Submit" button is located at the bottom right of the form.

Slika 18: Forma za vnašanje izdelka (lasten vir)

Prijavljen kot: Admin Admin

Generiranje Memberja
Vnos Izdelka
Rezervacija
Košarica
Prezem
Analitika
Odjava

05b1b

Submit

ID	Ime	Cena	Provizija	Davek	Čas
004F0	Elan SLX	-120,00 €	0,11 %	0,21 %	min
9349C	Atomic redster	200,00 €	0,11 %	0,21 %	3 min
Vsota: 80,00					

Račun

Slika 19: Izpis košarice udeleženca (lasten vir)

5.4 Generiranje dokumentov

Generiranje dokumentov, je bil eden izmed večjih ciljev izdelka. Dokumenti so izjemno pomembni, saj udeleženci z njimi dokazujejo lastništvo nad izdelkom. Dokumente sem generiral s pomočjo PHP-ja in procedur iz katerih sem črpal podatke iz podatkovne baze.

Ko blagajnik vpiše identifikacijsko številko udeleženca, se na ekran izriše njegova košarica, ter gumb za generiranje računa. Ob kliku na gumb se izvede plačilo in izriše račun, kateri se natisne in služi, kot dokaz o plačilu.


Datum Izdaje: 29.03.2020

Sejem d.o.o.
Celovška 113
1000 Ljubljana

RAČUN

ID artikla	Ime artikla	Davek	Provizija	Cena
004F0	Elan SLX	21,00%	11,00%	-106,80€
9349C	Atomic redster	21,00%	11,00%	200,00€
Skupna cena:				93,20€

Slika 20: Račun (lasten vir)

6. UPORABA INFORMACIJSKEGA SISTEMA

Uporabnik spletne aplikacije se prijavi z podanim uporabniškim imenom in geslom v začetnem oknu spletne aplikacije in se tako prijavi v spletno aplikacijo in podatkovno bazo. Pričaka ga spletna stran z navigacijsko ploščo. Ta nato izbere dogodek, ki ga želi opraviti. Odpre se mu forma v katero izpolni ustrezne podatke. Če se pri dogodku generira dokument se mu ta odpre in ga lahko natisne in izda udeležencu sejma. Ko uporabnik konča z delom pritisne gumb, ki ga odjavi.

Navigacijska plošča:

- **Vpis** – če uporabnik želi uporabljati informacijski sistem se potrebuje prijaviti. Okno za vpis je vstopno okno.
- **Generiranje udeleženca** – uporabnik vnese število udeležencev, ki jih želi generirati in izbiro potrdi z gumbom potrdi. Na zaslon se izpišejo identifikacijske številke udeležencev.
- **Vnos izdelka** – uporabnik vnese v formo podatke in potrdi izbiro z gumbom potrdi. Če je bil vnos uspešen se uporabniku nato odpre dokazilo o prevzemu katerega natisne.
- **Rezervacija** – uporabnik vnese v formo identifikacijsko številko kupca in izdelka, ter potrdi izbiro z gumbom potrdi. Na zaslon se izpiše sporočilo, če je bila rezervacija uspešna.
- **Košarica** – Uporabnik vnese identifikacijsko številko udeleženca in izbiro potrdi z gumbom potrdi. Na zaslon se izriše tabela z uporabnikovimi izdelki za katere mu je potrebno izplačati ali pa zaračunati določeno vsoto denarja. Izriše se tudi gumb z imenom račun, kateri ob uporabnikovem kliku opravi transakcijo in uporabniku odpre račun katerega natisne in izda udeležencu.
- **Prevzem** – Uporabnik vnese identifikacijsko številko udeleženca in izbiro potrdi z gumbom potrdi. Na zaslon se mu izpiše seznam izdelkov udeleženca, katere mu je potrebno izdati. Izriše se tudi gumb izdaja, kateri ob uporabnikovem kliku opravi transakcijo in izriše potrdilo o prevzemu.
- **Odjava** – ob kliku na gumb odjava, se uporabnik odjavi. Odpre se vstopno okno za prijavo.

7. MOŽNE IZBOLJŠAVE

Podatkovna baza je zasnovana dobro in deluje optimalno. Procedure in sprožilci prav tako delujejo pravilno in optimalno.

Izboljšati bi bilo potrebno spletno aplikacijo, ki jo bi lahko prilagodili za mobilne telefone ter omogočili generiranje QR kod, s katerimi bi bilo lažje in hitrejšje opravljati transakcije, saj bi se tako izognili tipkanju identifikacijskih števil. Izdelali bi lahko tudi spletno aplikacijo za udeležence, kjer bi lahko beležili svoje rezervacije, nakupe in spremljali njihove artikle.

8. ZAKLJUČEK

S končnim izdelkom sem zadovoljen. Izpolnjuje vse cilje, ki sem si jih zastavil in odgovarja na prvoten problem. Izdelek predstavlja praktičen primer uporabe baze podatkov, ter črpanje uporabnih podatkov iz nje. Uporabil sem veliko programskih jezikov in orodij. Pri načrtovanju in izdelavi sem se veliko zabaval, ter tudi ogromno naučil.. Izdelek, bi lahko nadgradil z spletno aplikacijo za udeležence sejma.

9. VIRI

- PHP dokumentacija: <https://www.php.net/manual/en/language.references.php> (26. 3. 2020)
- HTML dokumentacija: <https://www.w3schools.com/tags/> (24. 3. 2020)
- CSS dokumentacija: <https://www.w3schools.com/cssref/> (26. 3. 2020)
- Firebird dokumentacija:
https://firebirdsql.org/file/documentation/reference_manuals/fblangref25-en/html/fblangref25.html (26. 3. 2020)

9.1 Citati

- John Johnson. (2009) Pridobljeno s strani:
<https://www.defprogramming.com/quotes-by/john-johnson/> (28. 3. 2020)

Izjava o avtorstvu

Izjavljam, da je strokovno poročilo Podporni informacijski sistem v celoti moje avtorsko delo, ki sem ga izdelal samostojno s pomočjo navedene literature in pod vodstvom mentorja.

28. 3. 2020

Nejc Vrčon Zupan