



Tehnička dokumentacija

ReverseD

TIM:

Karlo Barjaktarić, Tea Jurić,
Naomi Kombol, Lisa Mihelec,
Nina Rubeša

SADRŽAJ:

<u>1. Uvod</u>	<u>3.</u>
1.1. O projektu	3.
1.2. O našem timu	3.
1.3. Programski jezik C++	3.
1.4. Ideja Projekta	6.
<u>2. Deteljan opis rada</u>	<u>7.</u>
2.1. Učitelji.txt	7.
2.2. cApp.cpp	7.
2.3. cApp.h	8.
2.4. cDnevnik.cpp	8.
2.5. cDnevnik.h	10.
2.6. cUcitelj.cpp	11.
2.7. Korisnici.txt	14.
2.8. FormLogin.h	15.
2.9. FormLogin.cpp	15.
2.10. Ostalo	18.
<u>3. Tehnička dokumentacija</u>	<u>19.</u>
3.1. Tehnologije	19.
3.2. Korisničko sučelje	19.
3.3. Pokretanje programa	20.
3.4. Logo programa	
<u>4. Najčešća pitanja</u>	<u>21.</u>
<u>5. Zaključak</u>	<u>22.</u>

1. UVOD

1.1. O projektu

Ovaj projekt je napravljen u sklopu online nastave, a pokrenut je pod vodstvom prof. Gorana Bonete. Radi se o zadatku u kojem su učenici četvrtih razreda matematičkog smjera Gimnazije Andrije Mohorovičića Rijeka, svrstani slučajnim odabirom u tim u kojem su se trebali upoznati sa njima potpuno nepoznatim programskim jezikom. Nadalje, bili su dužni na temelju novostečenog znanja, zajedničkim snagama te raspodjelom rada, napraviti zanimljiv i inovativan program u tome programskom jeziku, dokumentaciju toga programa te naposljetku screencast. Radili su svatko od svoje kuće.

1.2. O našem timu

Tim broj 3 je sastavljen od 5 učenika 4.4 razreda, a to su: Karlo Barjaktarić, Tea Jurić, Naomi Kombol i Lisa Mihelec. Tim3 je raspodijelio svoje poslove ravnomjerno tako da nitko među njima nije ostao izopćen ili napravio sam puno više posla u odnosu na ostale što je rezultiralo kvalitetno obavljenom zadatku.

Zadatak ovog tima je bio upoznati se sa programskim jezikom C++ o kojem će se više saznati u sljedećem ulomku.

1.3. Programski jezik C++

C++ je objektno orijentirani programski jezik kojim su pisani mnogi današnji programi s kojima se možemo susreći u svakodnevnom životu korištenjem računala. C++ je statički tipkan, sastavljen, ima opću namjenu te je osjetljiv na velika i mala slova. Također, ima slobodan oblik koji podržava procesno, objektno orijentirano i generičko programiranje. Jezik je razvijen u Bell Labs ili Bell Telephone Laboratories Inc. tj. u nezavisnom razvojnom laboratoriju američke telekomunikacijske tvrtke United States Bell System pod rukovodstvom danskog računalnog znanstvenika Bjarnea Stroustrupa. On je 1979. godine započeo je razvoj C++ jezika kao proširenje programskom jeziku C pa mu je originalno ime bilo "C with classes" (eng. C s klasama). Zbog velike je potražnje 1998. je postao standardnim jezikom, a do danas je postao najpopularnijim programskim jezikom.

Zašto programeri uopće koriste C++?

1. Programerima pruža visoku razinu kontrole resursa sustava i memorije.
2. Daje jasnu strukturu i omogućava ponovnu upotrebu koda, svodeći troškove razvoja.
3. Prenosiv je i može se koristiti za razvoj aplikacija koje se mogu prilagoditi na više platformi.
4. Blizak je C # i Java jeziku pa je programerima jednostavno prebaciti se na C ++ ili obrnuto.
5. Zabavan je i jednostavan za učenje.
6. Često se sa njime susrećemo u današnjim operativnim sustavima, grafičkim korisničkim sučeljima i ugrađenim sustavima.
7. Podržava razne stilove programiranja. Možete pisati u stilu Fortran, C, Smalltalk itd. Svaki stil može učinkovito postići svoje ciljeve uz održavanje učinkovitosti izvođenja i prostora.

Sastavnice jezika

1. Jezgroviti jezik - daje sve građevne blokove uključujući varijable, tipove podataka i literale itd.
2. Standardna knjižnica C++-a - bogat set funkcija koje manipuliraju datotekama, nizovima itd.
3. Standardna biblioteka predložaka (STL) - bogat skup metoda manipulacije strukturama podataka itd.

Osnovna sintaksa C ++

Kada razmotrimo program C ++, on se može definirati kao zbirka objekata koji komuniciraju pozivanjem metoda jedne s drugima. Važno je definirati:

- Objekte - instanca klase koja ima stanja i ponašanja.
- Klase - predložak ili nacrt koji opisuje ponašanja ili stanja čiju vrstu objekt podržava
- Metode – ponašanju kojem su zapisane logike, manipulira se podacima i izvršavaju se radnje.
- Promjenjive instance - Svaki objekt ima svoj jedinstveni skup varijabli instanci. Stanje objekta stvara se vrijednostima dodijeljenim ovim varijablama instance.

Primjer koda i objašnjenje

```
#include <iostream>
using namespace std;

// main() mjesto gdje započinje izvršavanje programa.
int main() {
    cout << "Hello World"; // ispisuje Hello World
    return 0;
}
```

- Jezik C ++ definira nekoliko zaglavlja sa **#include** koja sadrže informacije koje su ili potrebne ili korisne vašem programu. Za ovaj program potrebno je zaglavlje **<iostream>**.
- Linija pomoću **namespace std;** ona govori prevoditelju, koji se koristi za sastavljanje izvornog koda u konačni izvršni program, da koristi prostor s imenima **std**. Prostori imena su relativno noviji dodatak C ++.
- Sljedeći redak **// main () je mjesto gdje započinje izvršavanje programa** je jednoređni komentar dostupan na C ++. Komentari u jednom retku počinju sa **//** i zaustavljaju se na kraju retka. Postoje i komentari u više redaka koji započinju sa **/*** a završavaju sa ***/**.
- Linija **int main ()** je glavna funkcija u kojoj započinje izvršavanje programa.
- Sljedeći redak **' cout << "Hello World"; '** uzrokuje da se na zaslonu prikaže poruka "Hello World".
- Sljedeći redak vraća 0; prekida main () funkciju i tjera je da vraća vrijednost 0 procesu pozivanja.

Važne osnovne značajke

- U C ++ zarez je krajnji izraz. Odnosno, svaka pojedinačna izjava mora biti završena točkom sa zarezom. Označava kraj jedne logičke cjeline.
- C ++ identifikator je naziv koji se koristi za identificiranje varijable, funkcije, klase, modula ili bilo koje druge stavke definirane od strane korisnika. Identifikator započinje slovom A do Z ili a do z ili podvlakom (**_**), a zatim slijedi nula ili više slova, podvlaka i znamenki (0 do 9).
- C++ je osjetljiv na mala i velika slova

Detaljnije o C++ programskom jeziku će se objasniti kroz iduću cjelinu.

1.4. Ideja projekta

Program „ReverseD“ je osmišljen sa ciljem omogućavanja lakše, brže i bolje komunikacije između profesora i učenika. On predstavlja novi oblik E-dnevnika u kojem bi učenici, ocjenjivanjem rada profesora zajedničkom ocjenom svakog pojedinog razreda te bilješkama po učeniku, slali povratne informacije profesorima o njihovom radu. Stoga cilj ovog projekta je, na temelju vrednovanja rada profesora, osigurati učenicima bolju i kvalitetniju nastavu te bolju komunikaciju između razreda i učenika. Profesori bi dobili informacije o tome kakav oblik nastave i pristup profesora bolje ‘leži’ učenicima, koji su najveći problemi učenicima tijekom učenja pojedinog predmeta, na čemu bi profesor mogao bolje poraditi, postoje li nekakve nejasnoće u samoj nastavi, zašto dolazi do problema sa ocjenama ili nesporazuma među učenicima i profesorima (ako uopće dolazi do njih) itd. Ovaj program bi također predstavio učenicima, koji imaju problema sa sramežljivosti i komunikacijom, lakši način na koji mogu izraziti svoje mišljenje. Sve bilješke od strane učenika bi bile anonimne, ali će obavezno biti pregledavane kako se ne bi nikoga omalovažavalo nego samo uputilo k boljoj nastavi.

2. DETALJAN OPIS RADA

2.1. Učitelji.txt

Ovaj dio programa predstavlja TXT datoteku ili fiksna datoteka. TXT datoteka je vrsta računalne datoteke koja ima oblik niza linija ASCII ili niz znakova nekog drugog kodnog sistema, dok je kraj datoteke obilježen posebnim znakom kraja datoteke tzv. EOF znakom. U njoj se nalazi niz svih profesora koji su zaposleni u Gimnaziji Andrije Mohorovičića Rijeka i rubrike koje učenici mogu ocjenjivati a to su: Odnos prema radu, Susretljivost i Težina ispita u odnosu na predavano. Ocjene su podijeljene po svim prvim razredima zajedno, zatim svim drugim razredima zajedno, i isto tako i za treće i četvrte. Nekim profesorima su dodane već ocjene jer smo program prosljedili 4.4 razredu kako bi imali primjere za rad.

Primjer:

```
27  Milka Dajak, Odnos prema radu, 1R, 2R, 3R, 4R, 1, 1, 4, 2, 2, 1, Helpfulness, 1R, 2R, 3R, 4R, 1, 1, 1, 1, 1, 1, Težina ispita u odnosu na predavano, 1R, 2R, 3R, 4R, 1, 2, 1, 1, 2, 1, Kraj, Kraj
```

1R, 2R – predstavljaju razrede

1, 1, 1, - predstavljaju primjere ocjena

Milka Dajak – profesor Gimnazije Andrije Mohorovičića Rijeka

2.2. cApp.cpp

Ovaj dio programa predstavlja klasu koja sadrži cijeli program tj. povezuje sve dijelove programa u cjelinu. Zadnji dio predstavlja funkciju koja stvara prvi prozor pri paljenju programa (OnInit = On initialisation). Funkcija je prikazana ispod:

```
bool
cApp::OnInit()
{
    m_frame1 = new FormLogin(wxT("Log In"));
    m_frame1->Show();
    return true;
}
```

2.3. cApp.h

U ovom dijelu programa se definira klaca cApp. Tamo joj se dodaje atribut sa wxApp -a koji predstavlja klasu za prozore. Da bismo to uspjeli, trebali smo uključiti dijelove programa: "FormLogin.h" koji smo već prije definirali, "cDnevnik.h" koji će biti definiran u poglavlju kasnije i iz wxWidgets-a "wx/wx.h".

2.4. cDnevnik.cpp

Ovaj dio programa definira tj. stvara i poziva prozor cDnevnik i s njime se poziva njegova funkcija „cDnevnik.h”.

Dalje smo povezali datoteku “Škole.txt” ,u kojoj se nalaze sve škole koje će biti prisutne u našem ReverseD dnevniku, kako bi napravili listu svih škola (iako trenutno je unutra jedna škola Gimnazija Andrije Mohorovičića koja nam služi kao primjer). A funkcija kojom pozivamo datoteku i radimo listu izgleda ovako:

```
wxArrayString
ListaSkola()
{
    wxArrayString skole;
    string Red;
    ifstream Skole("Škole.txt");
    while (getline(Skole, Red)) {
        skole.Add(Red);
    }
    return skole;
}
```

Nadalje provjeravamo jesmo li povezali točnu školu sa točnim profesorima iz datoteke “Učitelji.txt”.

```
wxArrayString
ListaUc(wxString
skola) {
    string StdSkola = skola.ToStdString();
    bool tocnaskola = false;
    wxArrayString ucitelji;
    string Red;
    ifstream Ucitelji("Učitelji.txt");
```



```

        while (getline(Ucitelji, Red)) {
            if (tocraskola && Red != "End") {
                ucitelji.Add(Red.substr(0, Red.find(',')));
            }
            else if (Red == "End" && tocraskola == true) {
                break;
            }
            if (Red == StdSkola) {
                tocraskola = true;
            }
        }
        return ucitelji;
    }
}

```

Dalje smo napravili combobox tj. kombinacijski okvir. Kombinacijski okvir je element grafičkog korisničkog sučelja. Namjena mu je korisniku omogućiti birati između različitih mogućnosti zadanih već prije.

Comboboxeve smo zatim smjestili funkcijom ispod u toolbar tj. alatnu traku. Alatna traka je traka grafički upravljački element na koji se postavljaju zaslonski gumbi, ikone, izbornici ili drugi ulazni ili izlazni elementi nekog programa, no njih ne možemo samo tako otvoriti. Da bi mogli otvoriti ikone moramo napraviti menubar tj. traku izbornika. Traka izbornika je grafički upravljački element koji sadrži padajuće izbornike. Namjena trake izbornika je opskrba zajedničkim kućištem za prozore ili programe specifične za aplikacije koji omogućuju pristup takvim funkcijama kao što su otvaranje datoteka, interakcija s aplikacijom ili prikaz dokumentacije ili priručnika za pomoć. U ovom slučaju, na njoj možemo odabrati školu i profesore za koje želimo upisati neke ocjene ili bilješke. To u programu izgleda ovako:

```

//Napravi
menubar

m_MenuBar = new wxMenuBar();
this->SetMenuBar(m_MenuBar);

//Napravi toolbar u koji idu ComboBoxevi
m_ToolBar = this->CreateToolBar(wxTB_HORIZONTAL, wxID_ANY);

m_box1 = new wxComboBox(m_ToolBar, 10003, wxEmptyString,
wxDefaultPosition, wxSize(300, 50), ListaSkola());

```

2.5. cDnevnik.h

Ovaj dio programa definira klasu cDnevnik i daje joj atribute klase wxFrame. WxFrame je klasa wxWidgets-a koja omogućava stvaranje prozora kada otvorimo program. A kako bi sve bilo na mjestu kad otvorimo prozor moramo naravno u „cDnevnik.h” (u taj isti prozor) dodati atribute kako bi mogli odabirati ono što želimo raditi s programom. Ti atributi su: menubox, toolbar, combobox, listbox tj. popisni okvir (element grafičkog korisničkog sučelja koji je vrlo sličan kombinacijskom okviru, a amjena mu je korisniku omogućiti birati između različitih mogućnosti zadanih već prije) te na kraju naravno gumb (button) kako bi mogli odabrati. Dodavanje atribura u “cDnevnik.h”:

```
class cDnevnik :
public
wxMDIParentFrame
{
public:
    cDnevnik();
    ~cDnevnik();

public:
    //Daje cDnevniku ove atribute
    wxToolBar* m_ToolBar = nullptr;
    wxMenuBar* m_MenuBar = nullptr;
    wxTextCtrl* m_txt3 = nullptr;
    wxTextCtrl* m_txt4 = nullptr;
    wxListBox* m_list2 = nullptr;
    wxButton* m_btn2 = nullptr;
    wxComboBox* m_box1 = nullptr;
    wxComboBox* m_box2 = nullptr;

    void OnSelect(wxCommandEvent& evt);
    void OnSelect2(wxCommandEvent& evt);

    wxDECLARE_EVENT_TABLE();
};
```

2.6. cUcitelj.cpp

To je najveći dio koda koju u prvom dijelu provjerava ocjene profesora koje su unesene. Odobrava samo ocjene od 1 do 5, a ako je unesena kriva ocjena vraća se vrijednost "False". Dalje program nastavlja sa pregledavanjem svih redova tj. svakog odabranog reda koji sadrži podatke nekog određenog profesora čiji red gledamo. Poslije toga se ocjene vade po kriterijima te se između njih tj. između svakog kriterija stavi "end" da se ocjene odvoje. Iz predhodnog arraya sa ocjenama odvojenih po 3 kriterija koje smo odredili i odvojeni po razredima. Poslije toka sakupljanjem svih ocjena se računa prosjek po razredu za svaki određeni kriterij. Kada se ispisuje ocjena dodaje se datum kad je ocjena upisana i kada su napisane bilješke. To se određuje funkcijom:

```
wxDateTime Sad = wxDateTime::Now();  
wxString vrijemeSad = Sad.Format("%d.%m.%Y.", wxDateTime::CET).c_str();
```

a) Dalje je definirano upisivanje bilješki i dopuštanje njihovog upisivanja:

```
void  
StringWriteTo1(string  
unos) {  
  
    FILE* korisnici = fopen("Bilješke.txt", "a");  
    const char* uneseno = unos.c_str();  
    fputs(uneseno, korisnici);  
    fputs("\n", korisnici);  
    fclose(korisnici);  
}
```

b) Nakon toga, svaka bilješka se prvo treba poslati u listu da se nebi odmah ispisala u tablici kako bi je mogli administratori provjeriti. Šalje se u "Bilješke.txt" i tamo se može provjeriti. Stavljaju se zatim sve bilješke u vektor i briše se sve u postavljenom fajlu kako bi se moglo vratiti sve nazad u isti fajl i dodavati red po red kako bi zadnji došao na kraj kod učitelja u kojem želimo staviti bilješku.

```

FILE* Bilješke1 =
fopen("Bilješke.txt",
"w");

fputs("", Bilješke1);

extern wxString ucitelj;

bool uciteljJe = false;

//Stavi red po red natrag u fajl t.k.d. najnoviji
dođe prije "End," pod učitelj kojem je namjenjen
for (int i = 0; i < redovi.size(); i++) {
    if (redovi[i] == ucitelj.ToString()) {
        uciteljJe = true;
    }
    if (uciteljJe) {
        StringWriteTo1(redovi[i]);
        StringWriteTo1(s1);
        uciteljJe = false;
    }
    else {
        StringWriteTo1(redovi[i]);
    }
}

```

c) Ako je bilješka dobro unesena, unositelj dobiva povratnu informaciju o tome je li ona uspješno unesena ili ako nije. Ako je unesena uspješno ispisuje se “Bilješka je uspješno unesena”, a ako nije uspjelo se ispisuje “Neuspjeh. Unesi bilješku”.

d) Dalje se po redu proces za ocjenu:

1. Unese ocjena odnosno o razredu odmah poslje oznake razreda u ocjenama
2. Zatim ide zapisivanje u “Učitelj.txt”
3. Zatim se radi funkcija za unošenje ocjene odmah poslije razreda u “Učitelj.txt”
4. Da bi do tog došlo se navode kriteriji i stave se redovi u vektore, zatim kada se dođe do konkretnog učitelja čiju ocjenu želimo unijeti, moraju se proći svi elementi tog reda koji opisuju pojedinog profesora. Zatim se stvara funkcija za traženje tog profesora. Zatim se prolazi uneseni kriterij za profesora, koji, ako je kriterij dobar ili ispravan, čeka oznaku dobrog razreda sa “True”. Onda se mora provjeriti jeli određeni kriterij koji je unesen

dobro unesen kako bi mogli nastaviti dalje. Dalje se uređeni string vraća natrag u vektor i mora se cijeli unos potvrditi.

5. Zatim se natrag sve zapisuje u tekst file.
6. Ako nije ocjena nakon cijelog procesa dobro unesena ispisuje se “Ocjena nije od 1 do 5 ili format nije točan” (misli se da format pisanja kao broj ili slovo)

e) Poslije toga ide proces za bilješke:

1. Prvo se vraća string ako je u redu duljina bilješke koja je napisana u svakome redu
2. Onda radimo funkciju za odobriti cijelu bilješku ili je odbiti te opet onda prvo moramo staviti sve redove u vektor i obrisati sve u file-u kao i kod ocjena. Kasnije ako se pronađe dobra bilješka koja se poklapa sa onima u tekstnom okviru (textbox) tj. sa unesenom bilješkom onom kojom smo sami unijeli. Ako se pronađe ona se postavlja kao odobrena bilješka. Zatim se mora pronaći kraj u cijelom nizu bilješki i ta zadnje unesena bilješka koju smo unijeli ide na kraj cijelog niza.
3. Zapisuje se zatim sve u bilješke u programu da ih možemo vidjeti
4. Program je nakon toga spreman na novu bilješku.

f) Za kraj nakon cijelog procesa smo morali započeti definirati prozor u kojemu će sve napravljeno moći posloženo stajati i biti vidljivo korisnicima.

g) Da bi to napravio bilješke se moraju sve skupa staviti u rešetku u kojoj će biti odvojene i posložene po datumima. Zatim se traži ime profesora kojemu one pripadaju te kada dođe do određenog profesora bool vraća vrijednost na ”True”, a inače je “False”. Ako je ispravno onda se unose sve bilješke profesoru.

h) Dodatno smo definirali, nakon toga brisanje bilješki i što će program napraviti ako korisnik nije odobren i ako smo došli do kraja.

i) Završno se onda unose u tablicu sve srednje vrijednosti profesora.

```

wxArrayString ukupniPodatci =
SrednjeVrijednosti(Podatci1R(podatciUcitelja),
Podatci2R(podatciUcitelja),
Podatci3R(podatciUcitelja),
Podatci4R(podatciUcitelja));

int f = 0;
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 3; j++)
    {
        m_grid-
>SetCellValue(i, j, ukupniPodatci[f]);
        m_grid-
>SetCellFont(i, j, font);
        m_grid-
>SetReadOnly(i, j);
        f++;
    }
}

```

2.7.Korisnici.txt

Ovaj dio programa je tekstna datoteka poput one “Učitelji.txt” koja je već spomenuta. U ovoj datoteci trenutno se nalazi popis svih učenika 4.4 razreda Gimnazije Andrije Mohorovičića Rijeka, a lista može biti nadopunjena podacima ostalih učenika gimnazije jer naš program nije ograničen na jedan razred. “Korisnici.txt” sastoji se od niza učenika sa sljedećim podacima: korisničko ime, lozinka, razred i uloga (admin, korisnik).

Primjer:

```

13 silvia.petrovic, petrovic2, 4R, Korisnik, Kraj
14 bojan.puvaca, puvaca, 4R, Korisnik, Kraj
8 naomi.kombol, kombol, 4R, Admin, Kraj

```

Redni broj (13), korisničko ime (silvia.petrovic), lozinka (petrovic2), razred (4R), uloga (podcrtana zbog različitog pristupa i mogućnosti u programu između korisnika i admina)

2.8. FormLogin.h

Ovaj program definira klasu FormLogin i daje joj attribute klase wxFrame (omogućava stvaranje prozora). Ovo je header za dodavanje atributa “FormLogin” programu u kojem definiramo osnovne značajke (funkcije, grafičke elemente, gumbove, tekstualne prozore itd.). Nakon što smo uveli (include) potrebne plug-inove za rad s wxWidgets definirat ćemo klasu izvedenu iz wxFrame i kreirat klasu konstruktor i destruktor. Nadalje, definirat ćemo dva tekstualna prozora za upisivanje korisničkog imena i lozinke uz koje će stajati statični istoimeni tekst te tri gumba – “Login“, “Quit“ i “Profesor sam“.

```
private:
    wxStaticText* m_usernameLabel;
    wxStaticText* m_passwordLabel;
    wxTextCtrl* m_usernameEntry;
    wxTextCtrl* m_passwordEntry;
    wxButton* m_buttonLogin;
    wxButton* m_buttonQuit;
    wxButton* m_profesor;
    wxMessageDialog* dlg;
```

Za kraj, ostaje nam definirati funkciju koja će na pritisak gumba obaviti određeni zadatak, u ovom slučaju to su ulazak u program (ReverseD) kao učenik ili profesor i izlazak iz Log in prozora.

```
private:
    void OnQuit(wxCommandEvent& event);
    void OnLogin(wxCommandEvent& event);
    void ProfesorSam(wxCommandEvent& event);
private:
    DECLARE_EVENT_TABLE()
```

2.9. FormLogin.cpp

Ovaj dio programa definira tj. stvara i poziva prozor Log in koji sadrži zaglavnu funkciju “FormLogin.h”. Pošto se naš glavni program nadovezuje na Log in prozor, moramo ga povezati pomoću `#include "ReverseD.xpm"`. Glavna značajka Log in prozora je provjera je li uneseno korisničko ime i lozinka ispravno i usmjeravanje na glavnu stranicu. Stoga za početak moramo iz postojećeg tekstualnog dokumenta “Korisnici.txt” rastaviti pojedine stringove (redove podataka) na komponente, a ključan faktor rastavljanja ima zarez. To ćemo učiniti na sljedeći način:

```

wxArrayString SplitString3(string Red) {
    wxArrayString rastavljen;
    while (Red.find(",") != std::string::npos) {
        rastavljen.Add(Red.substr(0, Red.find(",")));
        Red = Red.substr(Red.find(",") + 2);
    }
    return rastavljen;
}

```

Definirat ćemo i veličinu prozora (340x150) i onemogućiti njegovo maksimiziranje. Nakon toga ćemo stvoriti ploču za dodavanje elemenata koje smo definirali u “FormLogin.h” kao što su tekstni okvir, tekstualni prozor i statični tekst. Uz svaki atribut odredit ćemo njegove koordinate i veličinu. U stvaranju okvira s rasporedom/veličinom dijeljenja koristit ćemo wxBoxSizer.

Primjer:

```

FormLogin::FormLogin(const wxString& title)
    : wxFrame((wxFrame*)NULL, wxID_ANY, title, wxPoint(wxID_ANY, wxID_ANY), wxSize(340, 150),
{
    m_usernameLabel = new wxStaticText(panel, wxID_ANY, wxT("Korisnicko ime: "),
wxDefaultPosition, wxSize(70, -1));
    hbox1->Add(m_usernameLabel, 0);
}

```

Stvaranje gumba s prilagođenim nazivom događaja “BUTTON_Login” odnosi se na “FormLogin.h” i ovaj se kod koristi za rukovatelje događaja “EVT_BUTTON”. To su gumbi “Login”, “Quit” i “Profesor sam”.

```

wxBoxSizer* hbox3 = new wxBoxSizer(wxHORIZONTAL);
m_buttonLogin = new wxButton(panel, BUTTON_Login, wxT("Login")); //Stvaranje gumba
s prilagođenim nazivom događaja BUTTON_Login, to se ime odnosi na FormLogin.h i ovaj
se kod koristi za rukovatelje događaja EVT_BUTTON
hbox3->Add(m_buttonLogin);

m_buttonQuit = new wxButton(panel, wxID_EXIT, ("Zatvori")); //Isti kao i gore
(redak 61), ali za ID koristimo zadani ID iz wxwidgets
hbox3->Add(m_buttonQuit);
vbox->Add(hbox3, 0, wxALIGN_RIGHT | wxTOP | wxRIGHT | wxBOTTOM, 10);

m_profesor = new wxButton(panel, 10010, wxT("Profesor sam")); //Dodaje mogućnost
prijave kao profesor
hbox3->Add(m_profesor);

```


Na samom kraju moramo usporediti unesene podatke s onima u “Korisnicima.txt”. Za početak uključit ćemo `bool` “jeAdmin” čija je vrijednost “false”. Za vrijeme provjere ako se uspostavi da četvrta komponenta u podacima korisnika odgovara riječi “Admin”, `bool` će automatski svoju vrijednost prebaciti na “true” i na taj način omogućiti adminu da pristupi programu s određenim beneficijama (odobravanje bilješki korisnika).

```
bool jeAdmin = false;
string razred;
```

Zatim ćemo pozvati funkciju “OnLogin” koja je zabilježena u “FormLogin.h” i koja obavlja akciju uz pritisak gumba ukoliko je traženi uvjet zadovoljen. Prvo, uzet ćemo vrijednosti upisane u tekstualne prozore (korisničko ime i lozinka) i spremiti u zasebne varijable (username, password). Kada smo to učinili, provjerit ćemo nekoliko stvari: jesu li podaci uopće uneseni i jesu li podaci točni. Ako podaci nisu uopće uneseni program će javiti upozorenje. Ako su uneseni (pohranjeni su u spomenute varijable) ispitat ćemo njihovu valjanost. Ako vrijedi jednakost između varijabli (username, password) i prve i druge komponente iz “Korisnik.txt” korisnik će se uvesti u ReverseD. No, još je ostala jedna važna komponenta koju moramo provjeriti. To je četvrta komponenta tj. uloga. Ukoliko ona glasi “Admin” korisnik će biti preusmjeren u ReverseD i omogućiti mu dodatne mogućnosti.

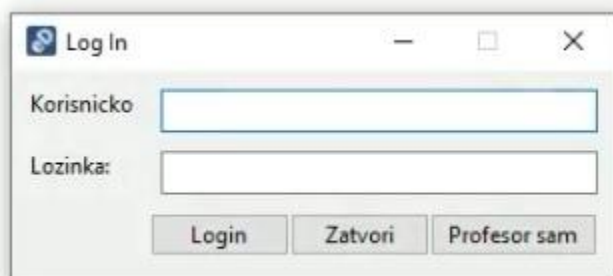
```
while (getline(Korisnici, podatci1)) {
    if (SplitString3(podatci1)[0] == username &&
        SplitString3(podatci1)[1] == password) {
        m_frame3 = new cDnevnik();
        m_frame3->Show();
        razred = SplitString3(podatci1)[2];
        if (SplitString3(podatci1)[3] == "Admin") {
            jeAdmin = true;
        }
    }
}
```

Na pritisak gumba “Login” i zadovoljenja uvjeta otvorit će se prozor ReverseD.

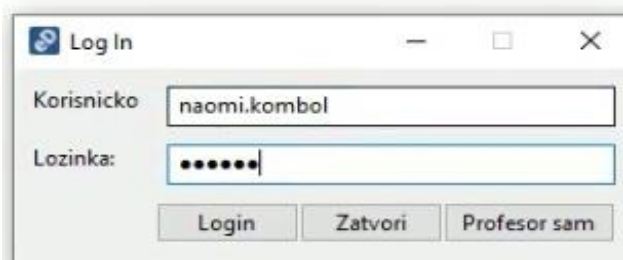
```
void
FormLogin::OnQuit(wxCommandEvent
& event) {
    Close();
}
```

Posljednji kod je definiranje događaja u tablici npr. ako korisnik klikne gumb za prijavu, kod će biti preusmjeren na funkciju “OnLogin”.

```
BEGIN_EVENT_TABLE(FormLogin, wxFrame)
EVT_BUTTON(BUTTON_Login, FormLogin::OnLogin)
EVT_BUTTON(wxID_EXIT, FormLogin::OnQuit)
END_EVENT_TABLE()
```



Izgled prozora “Log in”



Upisivanje podataka korisnika

2.10. Ostali dijelovi

1. "Bilješke.txt" predstavljaju bilješke koje su napisali učenici 4.4 razreda koje smo postavili u TXT datoteku kako bi ih mogli ubaciti u program
2. "Korisnici.txt" predstavljaju popis korisnika koji mogu pristupiti ReverseD e-dnevniku. Tu su navedeni učenici 4.4 razreda. Navedeni su i administratori tj. članovi Tima3 koji mogu odobriti ili ne odobriti bilješke i mijenjati kod.
3. "cUcitelj.h" samo definira kako će izgledati rešetke koje smo namjestili u "cUcitelj.cpp" i poziva funkcije wxWidgets – a.
4. "Proba2.vcxproj", "Proba2.vcxproj.filters", "Proba2.vcxproj.filters.txt", „Proba2.vcxproj.user“ nam služe kako bi mogli koristiti određene funkcije iz uređivača koda u kojima smo radili te kako bi povezali cijeli program u jednu veliku funkcionalnu i skladnu cjelinu.
5. "sample.rc" predstavlja uzorak ili probu
6. "Škole.txt" je definiran u 2.4. cDnevnik.cpp

3. TEHNIČKA DOKUMENTACIJA

3.1. Tehnologije

E-dnevnik za profesore ReverseD je izrađen multiplatformom wxWidgets, a povezivanjem wxWidgets sa Visual Studio Code-om program je bio pokretan i time je dobio ujedno i svoj konačan oblik.

WxWidgets-om je također napravljen i postavljen logo našeg programa.

Za kontrolu verzija programa, njegovo postavljanje, učitavanje te male dodatke korišten je GitHub tj. repozitorij GitHub-a GAM-4-4/ReverseD.

GitHub je globalna tvrtka sa sjedištem u Sjedinjenim Američkim Državama koja pruža hosting za kontrolu verzija softvera za razvoj softvera pomoću Gita. Ona okuplja najveću svjetsku zajednicu programera kako bi otkrili, dijelili i gradili bolji softver.

3.2. Korisničko sučelje

Za razvoj korisničkog sučelja koristili smo se wxWidgetom. WxWidgets je alat za stvaranje grafičkih korisničkih sučelja (GUI) za cross-platformske aplikacije. On također omogućava da se grafičko korisničko sučelje programa sastavi i pokrene na nekoliko računalnih platformi uz minimalne ili nikakve promjene koda. Širok izbor prevodilaca (compilers) i drugih alata koji se koriste s wxWidgets olakšava razvoj sofisticiranih aplikacija. WxWidgets podržava sveobuhvatan raspon popularnih operativnih sustava i grafičkih knjižnica (graphical libraries).

WxWidgets podržava:

- mrežno programiranje
- streamove
- međuspremnik i povlačenje i ispuštanje (drag and drop)
- multithreading
- učitavanje slika i spremanje u različitim formatima
- internetsku pomoć
- pregledavanje HTML-a i ispis itd.

3.3. Pokretanje programa

Za pokretanje koda koristili smo Visual Studio Code. On je lagan, ali moćan uređivač izvornog koda koji se izvodi na radnoj površini i dostupan je za Windows, macOS i Linux. Dolazi s ugrađenom podrškom za JavaScript, TypeScript i Node.js i ima bogat ekosistem proširenja za druge jezike (kao što su C++, C#, Java, Python, PHP, Go) i runtimes (kao što su .NET i Unity).

Visual Studio Code smo najprije pruzeli, a zatim smo pokrenuli instalacijski program (VSCodeUserSetup- {verzija} .exe). Tada je VS Code, prema zadanim postavkama, instaliran pod C: \ users \ {username} \ AppData \ Local \ Programs \ Microsoft VS Code te od tamo možemo pokrenuti kod koji želimo, a isto tako, možemo preuzeti i Zip arhivu, izdvojiti je i pokrenuti Code od tamo kako smo i mi u našem projektu većinom i radili.

Visual Studio Codespaces postavlja:

- Izvorni kod
- izvođenja
- Lintere i uređaei za uklanjanje pogrešaka
- Proširenja

3.4. Logo programa

Logo programa je postavljen preko wxWidgets-a, a pokretanjem programa ReverseD logo će biti prikazan na programskoj traci. Ideja za logo je dobivena od E-dnevnika za učenike koji se koristi u cijelom školskom sustavu.

Izgled:



4. NAJČEŠĆA PITANJA

Imaju li profesori pristup ReverseD e-dnevniku?

Profesori imaju pristup ReverseD samo kako bi mogli pogledati ocjene učenika i njihove bilješke, ali nemaju mogućnost napraviti nikakvu promjenu unutar programa kao što je mijenjanje napisanog, dodavanje ocjena ili bilješki itd. Profesori imaju pristup onakav kakav učenici imaju u školskom E-dnevniku.

Je li ReverseD napravljen samo za Gimnaziju Andrije Mohorovičića Rijeka?

ReverseD nije napravljen samo za GAM. GAM je uzet samo kao početni primjer, a u ReverseD se može dodati bilo koja škola i njeni profesori iz bilo kojeg grada u Republici Hrvatskoj.

Predstavlja li ReverseD izvor omalovažavanja profesora od strane učenika?

Ne, ReverseD je napravljen kao program čiji je cilj uputiti profesore k boljoj i kvalitetnijoj nastavi i boljoj komunikaciji profesora i učenika. Niti jedan oblik omalovažavanja nije dopušten, stoga sve bilješke koje učenici napišu će biti pregledane, a one koje su pretjerane i vode ponižavanju će biti obrisane i neće biti puštene u javnost.

Mogu li profesori znati tko je napisao koju bilješku?

Ne, profesori ne mogu znati tko je napisao bilješku jer je ReverseD napravljen kao program u kojemu je očuvana anonimnost i svi učenici mogu na taj način biti zaštićeni i slobodni reći svoje mišljenje o radu profesora.

Mogu li se bilješke brisati?

Jedino Tim3 tj. admin programa može obrisati bilješke i ima pristup njima, ostali nemaju tu mogućnost.

5. ZAKLJUČAK

Razvojem ReverseD programa cijeli tim je naučio puno novih spoznaja o programiranju općenito i C++ jeziku o kojem prije početka projekta nismo gotovo ništa znali. Sada smo stekli veliko znanje i došli do velikog napretka te smo naučili kako isprogramirati vrlo komplicirane i složene programe u C++ jeziku za što nebi mogli ni zamisliti da ćemo ikada moći. No najvažnije od svega, naučili smo koliko je timski rad bitan u životu te da treba cijeniti druge i tuđi rad i trud.

Naposlijetku htjeli bismo se zahvaliti prof. Goranu Boneti koji nas je smjestio u ovaj tim i time nam sve ovo omogućio.