

VELEUČILIŠTE U RIJECI

Valentina Bognolo, Fani Tomac

APLIKACIJA ZA EVIDENCIJU KEMIČALIJA

Projektna dokumentacija

Rijeka, 2022.

VELEUČILIŠTE U RIJECI

Poslovni odjel

Spec. dipl. struč. studij Informacijske tehnologije u poslovnim sustavima

APLIKACIJA ZA EVIDENCIJU KEMIČALIJA

Projektna dokumentacija

MENTOR

dr. sc. Marin Kaluža, prof. v.š.

STUDENTI

Bognolo Valentina 2422000125/19

Tomac Fani 2422000112/20

Rijeka, lipanj 2022.

Sadržaj

1.	OPIS SUSTAVA.....	5
1.1	Svrha i cilj sustava (Valentina Bognolo)	5
1.2	Specifikacija zahtjeva (Valentina Bognolo).....	5
1.3	Arhitektura sustava (Fani Tomac)	6
1.4	Korištene tehnologije (Fani Tomac)	6
2.	POSTAVLJANJE RAZVOJNOG OKRUŽENJA (Fani Tomac).....	8
3.	USE CASE DIJAGRAM (Valentina Bognolo)	10
4.	DIJAGRAM AKTIVNOSTI	11
4.1	Prijava – administrator (Fani Tomac).....	11
4.2	Odabir prijave – administratora (Fani Tomac).....	12
4.3	Evidencija stanja skladišta – administrator (Fani Tomac)	13
4.4	Registracija korisnika – administrator (Fani Tomac)	14
4.5	Upravljanje obavijestima – administrator (Fani Tomac)	16
4.6	Pregled obavijesti po datumu – administrator (Fani Tomac).....	17
4.7	Prijava – korisnik (Valentina Bognolo)	18
4.8	Odabir laboratorija – korisnik (Valentina Bognolo).....	19
4.9	Pregled obavijesti – korisnik (Valentina Bognolo)	20
4.10	Pregled obavijesti po datumu – korisnik (Valentina Bognolo).....	21
4.11	Uređivanje stanja laboratorija – korisnik (Valentina Bognolo).....	22
5.	DIJAGRAM KLASA (Valentina Bognolo).....	24
6.	SEKVENCIJALNI DIJAGRAM (Fani Tomac)	25
7.	MOCKUP-OVI KORISNIČKIH PROZORA (Fani Tomac)	27
8.	MAPIRANJE (Valentina Bognolo)	31
9.	TESTIRANJE SIGURNOSTI (Fani Tomac)	40
10.	VALIDACIJA SPECIFIKACIJE ZAHTJEVA (Valentina Bognolo).....	44
11.	SUDJELOVANJE NA PROJEKTU	45
	Literatura i izvori	47
	Popis slika.....	48
	Popis tablica.....	50

1. OPIS SUSTAVA

1.1 Svrha i cilj sustava (Valentina Bognolo)

Aplikacija je osmišljena za profesore u srednjoj školi koji koriste kemikalije u svrhu održavanja laboratorijskih vježbi. Cilj aplikacije je poboljšati i ubrzati komunikaciju među korisnicima te olakšati praćenje stanja kemikalija unutar skladišta i laboratorija. Osim za prikaz stanja kemikalija u skladištu i u laboratorijima, aplikacija bi se koristila i za evidenciju uzetih, odnosno donesenih kemikalija iz/u laboratorij/a; za evidenciju prenesenih kemikalija iz skladišta; za prikaz obavijesti te, u slučaju administratora, za unos novih i uređivanje postojećih obavijesti i kemikalija. Sustav obuhvaća dva aktera, administratora i korisnika te mu se pristupa putem prijave nakon što administrator registrira novog korisnika.

1.2 Specifikacija zahtjeva (Valentina Bognolo)

Funkcionalnosti administratora:

- *Prijava* (autentikacija u sustav unosom ispravnog korisničkog imena i lozinke);
- *Odabir prijave* (pristup administratoru);
- *Upravljanje obavijestima* (unos novih obavijesti, uređivanje postojećih obavijesti);
- *Pregled obavijesti po datumu* (odabir datuma za prikaz obavijesti);
- *Evidencija stanja skladišta* (unos novih kemikalija na listu postojećih, uređivanje podataka postojećih kemikalija);
- *Registracija korisnika* (unos novih korisnika).

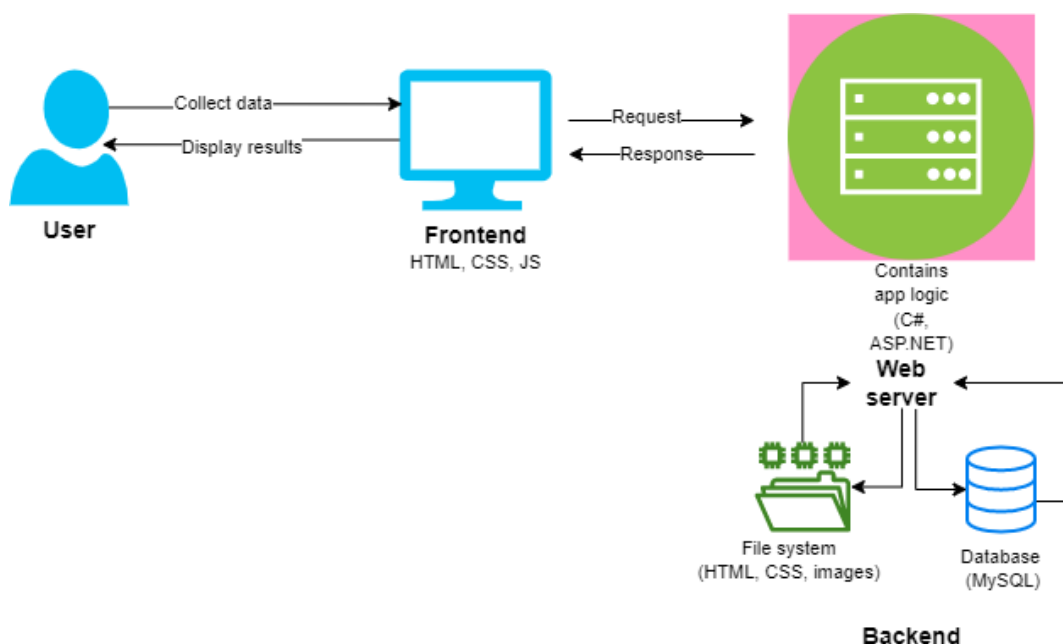
Funkcionalnosti korisnika:

- *Prijava* (autentikacija u sustav unosom ispravnog korisničkog imena i lozinke);
- *Odabir laboratorija* (odabir jednog od ponuđenih laboratorija);
- *Pregled obavijesti* (pregled svih obavijesti);
- *Pregled obavijesti po datumu* (odabir datuma za prikaz obavijesti);
- *Pregled stanja laboratorija* (pregled popisa kemikalija koje se nalaze u laboratoriju te njihove količine);
- *Uređivanje stanja laboratorija* (uređivanje količine kemikalija koje se nalaze u laboratoriju, pregled popisa kemikalija i njihovih količina u skladištu, dodavanje novih kemikalija iz skladišta).

1.3 Arhitektura sustava (Fani Tomac)

Arhitektura se sastoji od korisničkog sučelja za računalo te web i database servera. Aplikaciji se pristupa putem web preglednika. Komunikacija se izvršava na način da korisnik šalje zahtjev web poslužitelju koji mu prikazuje traženi sadržaj, dok se podaci o kemikalijama preuzimaju iz baze podataka koja se nalazi na poslužitelju baze podataka.

Slika 1. Arhitektura sustava



Izvor: obrada autora

1.4 Korištene tehnologije (Fani Tomac)

Za razvoj aplikacije korištene su sljedeće tehnologije: C#, ASP.NET, HTML, CSS i JS za back-end i front-end te MySQL za spajanje na bazu podataka.

C Sharp ili C# je objektno orijentirani programski jezik koji se može koristiti za stvaranje Windows klijentskih aplikacija, XML web usluga, distribuiranih komponenti, klijent – server aplikacija, aplikacija baza podataka te mnogih drugih. C# podržava koncept enkapsulacije, odnosno povezivanje važnih dijelova objekata u cjeline koje nisu vidljive korisniku, dozvoljava jednostavne promjene kod implementacije te sprječava unos nevažeće vrijednosti. Također podržava koncept polimorfizma i nasljeđivanja, odnosno svojstvo jedne klase da može preuzeti svojstva druge a može imati i dodatna svojstva te može prebrisati metode klase koju nasljeđuje.

ASP.NET je besplatan web okvir dizajniran za izradu web stranica i web aplikacija koji koristi HTML, CSS i JavaScript. Razvio ga je Microsoft kako bi omogućio izradu dinamičkih web stranica, web aplikacija i web usluga. ASP.NET nudi tri okvira za izradu web aplikacija: web obrasci, ASP.NET

MVC i ASP.NET web stranice. Svaki okvir cilja na drugačiji razvojni stil. ASP.NET web obrasci su dio ASP.NET okvira za web aplikacije, uključene u Visual Studiu. Pomoću web obrazaca mogu se izgraditi dinamičke web stranice koristeći poznati *drag-and-drop* (model povlačenja i ispuštanja). Nakon toga mogu se jednostavno postaviti svojstva, metode i događaji za kontrole na stranici.

MySQL (*Structured Query Language*) je *open source* platforma baze podataka podržana od strane Oracle Corporation-a. Razvijen je, prvenstveno, za rad s velikim bazama podataka te njegova povezanost, brzina i sigurnost čine MySQL server vrlo pogodnim za pristup bazama podataka na internetu. Također, MySQL baze podataka su relacijskog tipa.

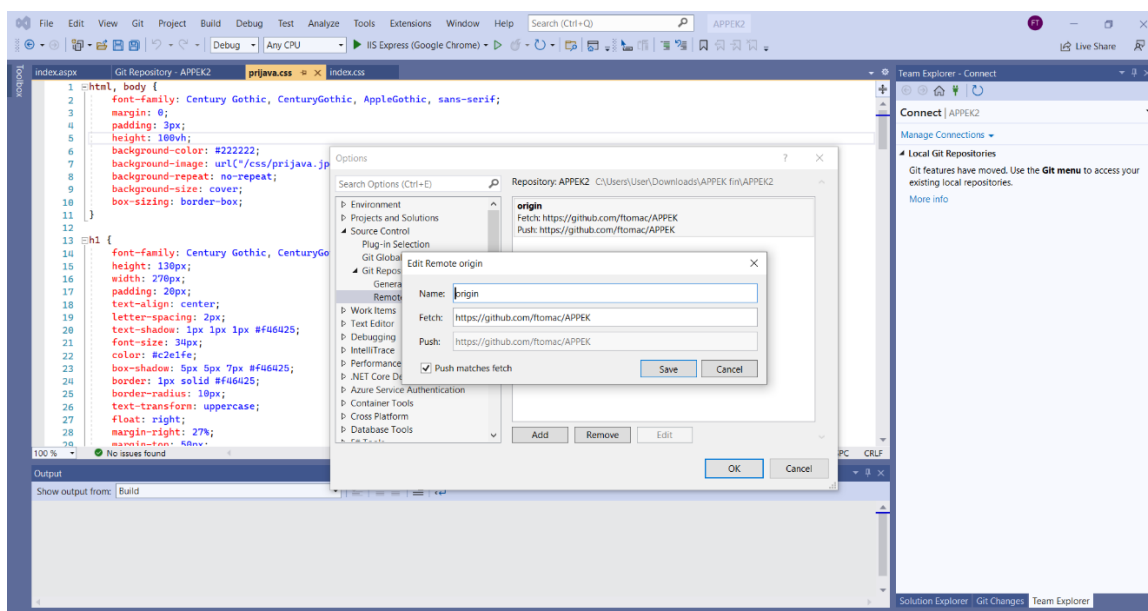
phpMyAdmin je besplatan i *open source* alat napisan u PHP-u, namijenjen upravljanju MySQL-om uz korištenje web preglednika. Može obavljati različite zadatke kao što su stvaranje, izmjena ili brisanje baza podataka, izvršavanje SQL izraza ili upravljanje korisnicima i dozvolama.

2. POSTAVLJANJE RAZVOJNOG OKRUŽENJA (Fani Tomac)

Kod postavljanja razvojnog okruženja potrebno je najprije preuzeti Visual Studio IDE (<https://visualstudio.microsoft.com/vs/>) i Git (<https://git-scm.com/download/win>) sa njihovih službenih stranica. Nadalje, potrebno je registrirati se na službenoj stranici GitHub-a (<https://github.com>). Nakon toga smo pomoću naredbe „push“ unutar Visual Studia, postavile projekt na GitHub. U branchu „master“ nalazi se cijeli projekt. S obzirom da smo imale problema sa spajanjem Visual Studia i GitHub-a, nismo bile u mogućnosti redovito commit-ati promjene, već smo cijelu aplikaciju učitale odjednom.

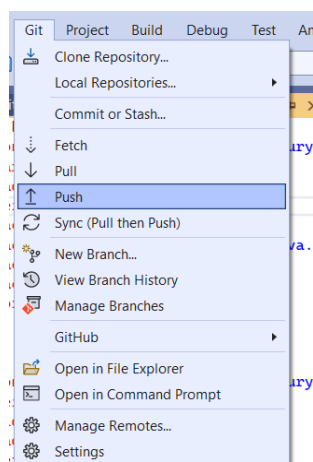
Aplikacija se nalazi na adresi: <https://github.com/ftomac/APPEK>

Slika 2. Prikaz projekta povezanog s Githubom



Izvor: obrada autora

Slika 3. Prikaz opcije za objavljivanje na GitHub



Izvor: obrada autora

Slika 4. Postavljena aplikacija na Githubu

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master had recent pushes 1 minute ago

Compare & pull request

master 2 branches 0 tags

Go to file Add file Code

This branch is 2 commits ahead, 2 commits behind main.

Contribute

ftomac Add project files. 10e93d8 8 minutes ago 2 commits

APPEK2	Add project files.	8 minutes ago
.gitattributes	Add .gitignore and .gitattributes.	8 minutes ago
.gitignore	Add .gitignore and .gitattributes.	8 minutes ago
APPEK2.sln	Add project files.	8 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

About

APPEK

0 stars 1 watching 0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

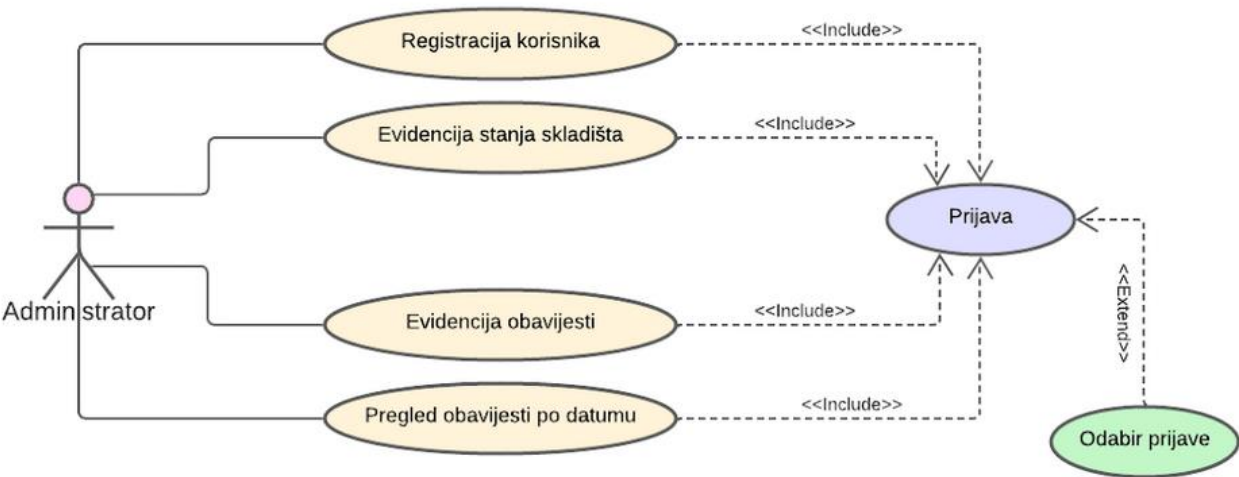
Contributors 2

ftomac vbognolo

Izvor: obrada autora

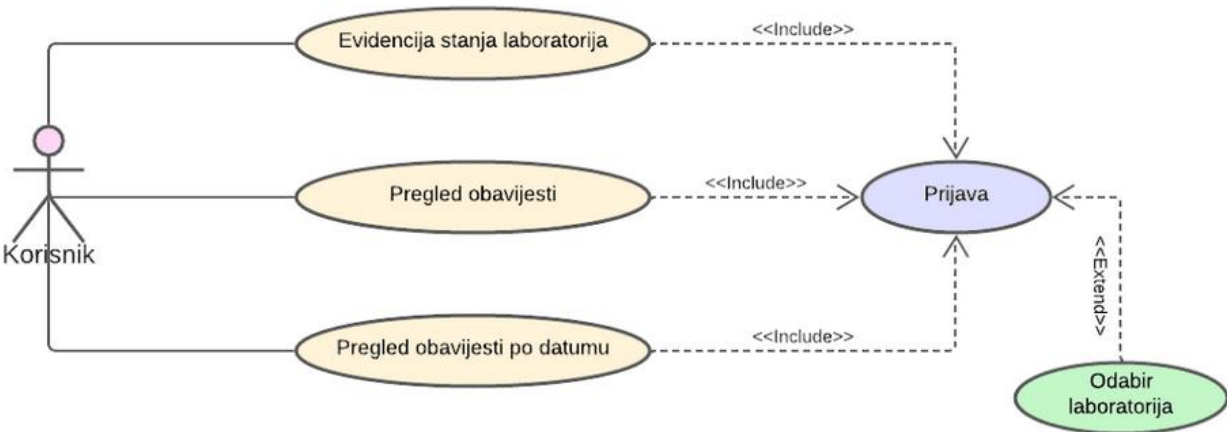
3. USE CASE DIJAGRAM (Valentina Bognolo)

Slika 5. Use case dijagram za administratora



Izvor: obrada autora

Slika 6. Use case dijagram za korisnika



Izvor: obrada autora

4. DIJAGRAM AKTIVNOSTI

4.1 Prijava – administrator (Fani Tomac)

Slika 7. Dijagram aktivnosti za prijavu administratora



Izvor: obrada autora

Identifikacijski sažetak: Administrator se prijavljuje u sustav.

Akteri: Administrator.

Preuvjeti: Registracija administratora u bazi podataka.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Administrator traži ulazak u sustav.
2. Sustav nudi unos korisničkog imena i lozinke.
3. Administrator unosi korisničko ime i lozinku.
4. Administrator potvrđuje unos.
5. Sustav provjerava korisničko ime i lozinku.
6. Sustav prikazuje odabir prijave.

Alternativne sekvence:

A1: nema

Sekvence s greškom:

E1: Neuspješna prijava u sustav – ponovni unos

1. Iz koraka G4 – Sustav javlja administratoru da jedno od podataka nije ispravno.
2. Administrator upisuje ispravne podatke.
3. Administrator potvrđuje unos.

4.2 Odabir prijave – administratora (Fani Tomac)

Slika 8. Dijagram aktivnosti za odabir prijave za administratora



Izvor: obrada autora

Identifikacijski sažetak: Administrator odabire način prijave u sustav.

Akteri: Administrator.

Preduvjeti: Autentikacija administratora u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Administrator traži ulaz u sustav sa administratorskim ovlastima.
2. Administrator potvrđuje odabir.
3. Sustav preusmjerava administratora na naslovnu stranicu.

Alternativne sekvence:

A1: Prijava administratora kao korisnika

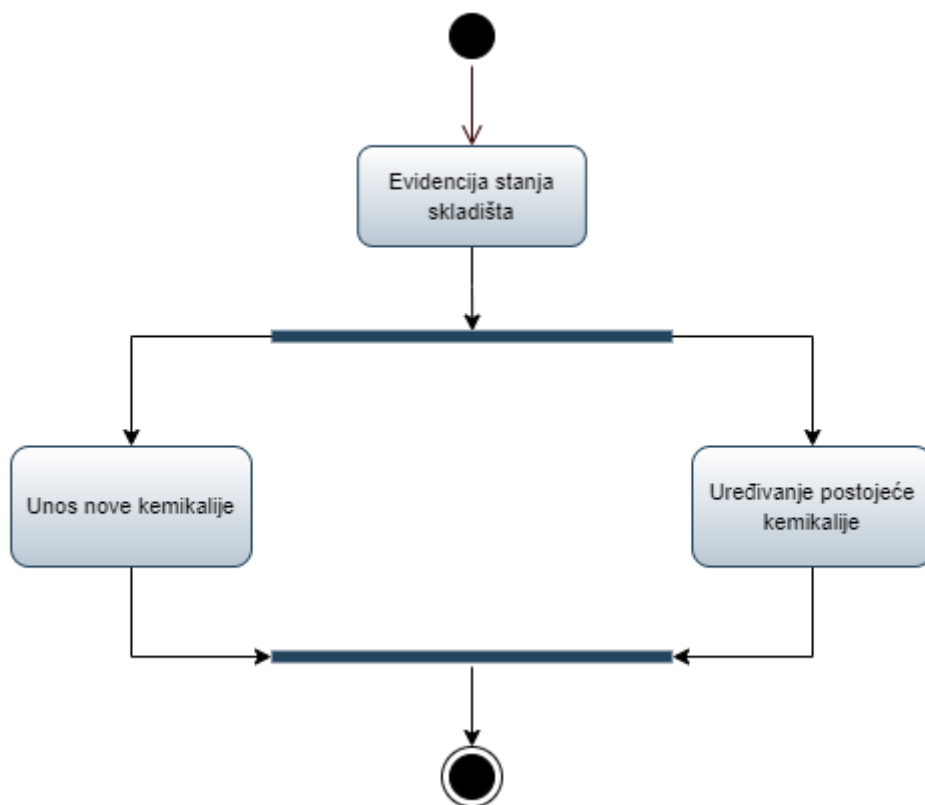
1. Administrator traži ulaz u sustav sa korisničkim ovlastima.
2. Administrator potvrđuje odabir.
3. Sustav preusmjerava korisnika na naslovnu stranicu.

Sekvence s greškom:

E1: nema

4.3 Evidencija stanja skladišta – administrator (Fani Tomac)

Slika 9. Dijagram aktivnosti za evidenciju stanja skladišta za administratora



Izvor: obrada autora

Identifikacijski sažetak: Unos novih kemikalija i uređivanje postojećih kemikalija u skladištu.

Akteri: Administrator.

Preuvjeti: Autentikacija administratora u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Administrator traži ulaz u skladište.
2. Administrator traži unos nove kemikalije.
3. Administrator unosi podatke nove kemikalije u za to predviđena mjesta.
4. Administrator potvrđuje unos.
5. Sustav pohranjuje kemikaliju u skladište.
6. Sustav ažurira pregled kemikalija u skladištu.

Alternativne sekvence:

A1: Uređivanje postojećih kemikalija u skladištu

1. Administrator odabire kemikaliju iz popisa kemikalija.
2. Administrator uređuje podatke postojeće kemikalije unosom novih podataka.

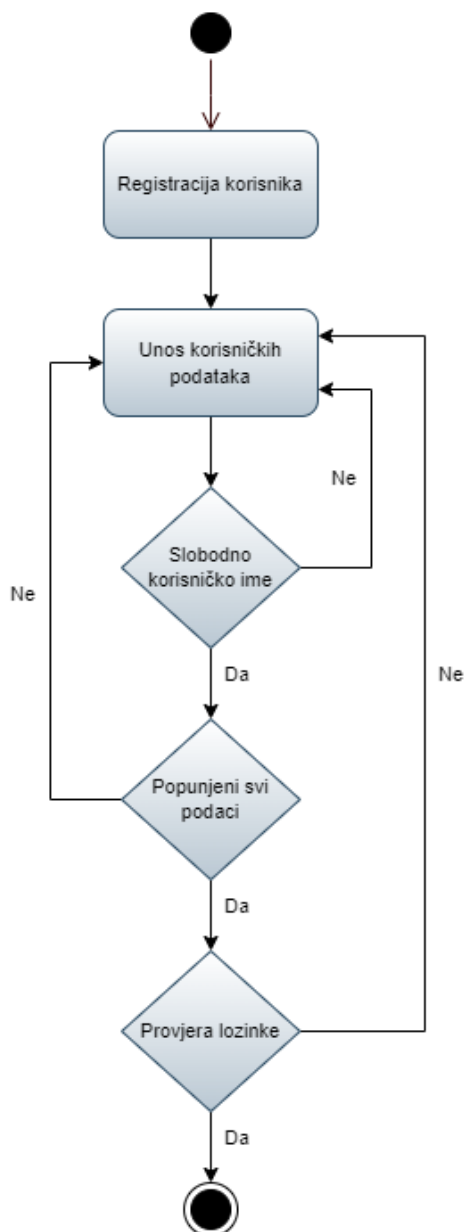
3. Administrator potvrđuje unos.
4. Sustav pohranjuje promjene odabrane kemikalije u skladištu.
5. Sustav ažurira pregled kemikalija u skladištu.

Sekvence s greškom:

E1: nema

4.4 Registracija korisnika – administrator (Fani Tomac)

Slika 10. Dijagram aktivnosti za registraciju korisnika za administratora



Izvor: obrada autora

Identifikacijski sažetak: Administrator registrira novog korisnika.

Akteri: Administrator.

Preduvjeti: Autentikacija administratora u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Administrator traži ulaz u registraciju korisnika.
2. Administrator unosi podatke novog korisnika.
3. Administrator potvrđuje unos.
4. Sustav radi provjeru unesenih podataka.
5. Sustav pohranjuje novog korisnika.

Alternativne sekvence:

A1: nema

Sekvence s greškom:

E1: Neuspješna registracija u sustav – nepopunjeni podaci

1. Iz koraka G3 – Sustav javlja administratoru da jedno od podataka nije popunjeno.
2. Administrator unosi sve tražene podatke.
3. Administrator potvrđuje unos.
4. Sustav pohranjuje novog korisnika.

E2: Neuspješna registracija u sustav – netočna potvrda lozinke

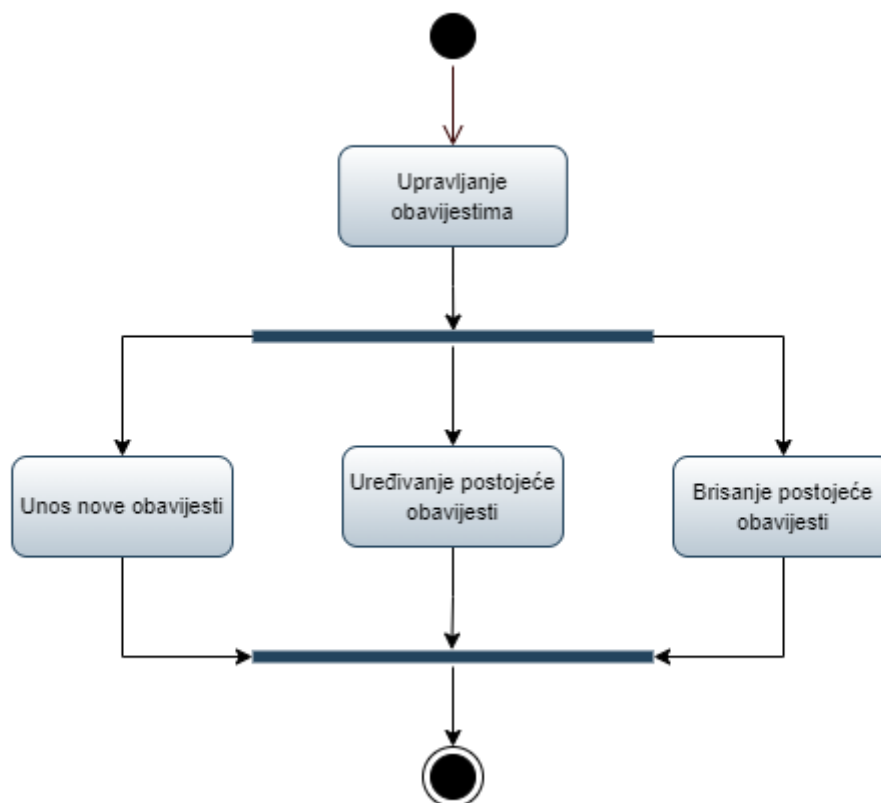
1. Iz koraka G3 – Sustav javlja administratoru da se lozinke ne podudaraju.
2. Administrator upisuje ispravne lozinke.
3. Administrator potvrđuje unos.
4. Sustav pohranjuje novog korisnika.

E3: Neuspješna registracija u sustav – postojeći korisnik

1. Iz koraka G4 – Sustav javlja administratoru da korisničko ime postoji.
2. Administrator unosi nepostojeće korisničko ime.
3. Administrator potvrđuje unos.
4. Sustav pohranjuje novog korisnika.

4.5 Upravljanje obavijestima – administrator (Fani Tomac)

Slika 11. Dijagram aktivnosti za upravljanje obavijestima za administratora



Izvor: obrada autora

Identifikacijski sažetak: Dodavanje novih i uređivanje postojećih obavijesti.

Akteri: Administrator.

Preduvjeti: Autentikacija administratora u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Administrator unosi novu obavijest.
2. Administrator potvrđuje unos.
3. Sustav pohranjuje obavijest.
4. Sustav prikazuje obavijest u ažuriranom *dashboard*-u.

Alternativne sekvence:

A1: Uređivanje postojećih obavijesti

1. Administrator odabire obavijest iz *dashboard*-a.
2. Administrator uređuje postojeću obavijest unosom nove obavijesti.
3. Administrator potvrđuje unos.
4. Sustav pohranjuje promjene odabrane obavijesti.

5. Sustav ažurira pregled *dashboard*-a.

A2: Brisanje postojećih obavijesti

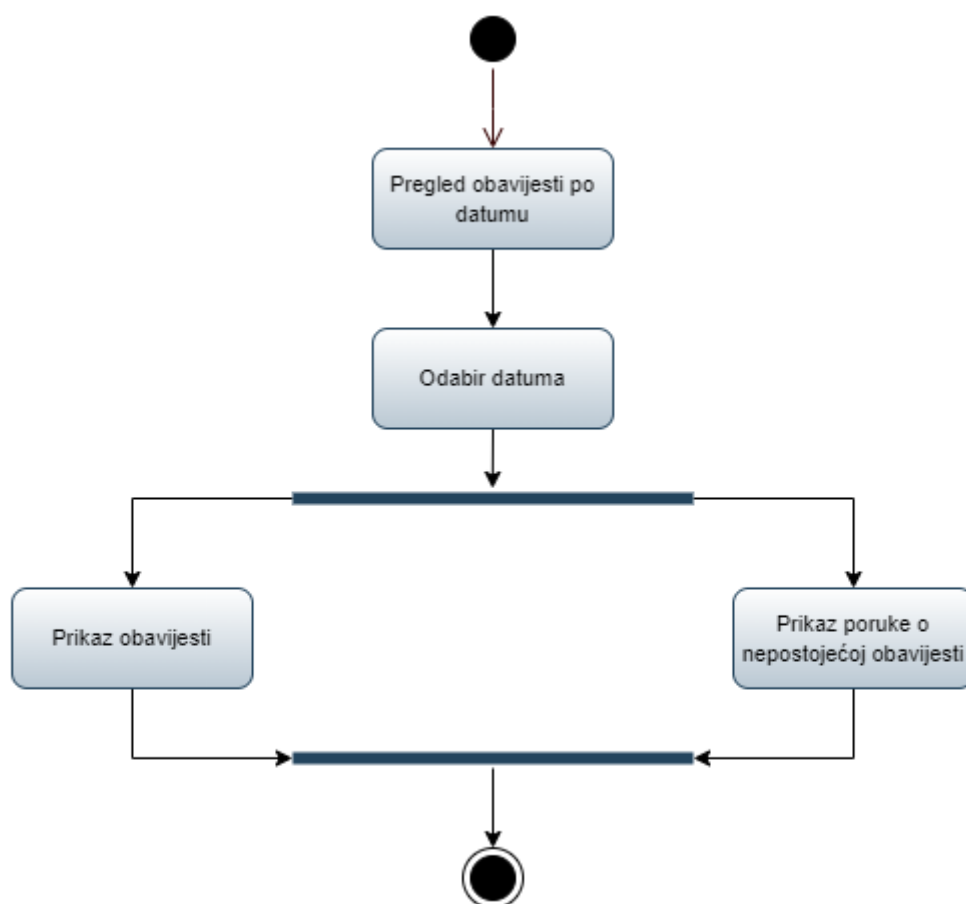
1. Iz koraka A1.1 - Administrator briše postojeću obavijest.
2. Administrator potvrđuje unos.
3. Sustav pohranjuje promjene.
4. Sustav briše odabranu obavijest.
5. Sustav ažurira pregled *dashboard*-a.

Sekvence s greškom:

E1: nema

4.6 Pregled obavijesti po datumu – administrator (Fani Tomac)

Slika 12. Dijagram aktivnosti za pregled obavijesti po datumu za administratora



Izvor: obrada autora

Identifikacijski sažetak: Administrator odabire datum za prikaz obavijesti.

Akteri: Administrator.

Preuvjeti: Autentikacija administratora u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Administrator odabire datum u kalendaru.
2. Sustav ispisuje sve obavijesti odabranog datuma.

Alternativne sekvence:

A1: Ispis nepostojeće obavijesti

1. Iz koraka G1 – Sustav ispisuje poruku nepostojećih obavijesti odabranog datuma.

Sekvence s greškom:

E1: nema

4.7 Prijava – korisnik (Valentina Bognolo)

Slika 13. Dijagram aktivnosti za prijavu korisnika



Izvor: obrada autora

Identifikacijski sažetak: Korisnik se prijavljuje u sustav.

Akteri: Korisnik.

Preduvjeti: Registracija korisnika u bazi podataka.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Korisnik traži ulazak u sustav.
2. Sustav nudi unos korisničkog imena i lozinke.
3. Korisnik unosi korisničko ime i lozinku.
4. Korisnik potvrđuje unos.
5. Sustav provjerava korisničko ime i lozinku.
6. Sustav prikazuje odabir laboratorija.

Alternativne sekvence:

A1: nema

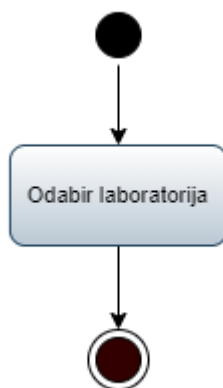
Sekvence s greškom:

E1: Neuspješna prijava u sustav – ponovni unos

1. Iz koraka G4 – Sustav javlja korisniku da jedno od podataka nije ispravno.
2. Korisnik upisuje ispravne podatke.
3. Korisnik potvrđuje unos.
4. Sustav prikazuje odabir laboratorija.

4.8 Odabir laboratorija – korisnik (Valentina Bognolo)

Slika 14. Dijagram aktivnosti za odabir laboratorija za korisnika



Izvor: obrada autora

Identifikacijski sažetak: Korisnik odabire jedan od ponuđenih laboratorija.

Akteri: Korisnik.

Preduvjeti: Autentikacija korisnika u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Sustav prikazuje laboratorije.
2. Korisnik odabire jedan laboratorij.
3. Korisnik potvrđuje odabir.
4. Sustav dodjeljuje laboratorij korisniku i preusmjerava na naslovnu stranicu.

Alternativne sekvence:

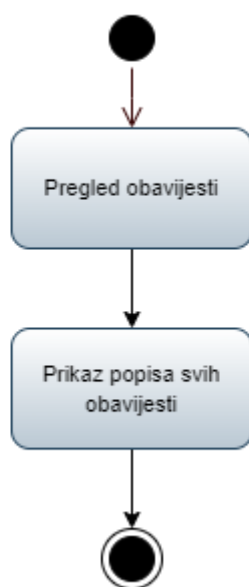
A1: nema

Sekvence s greškom:

E1: nema

4.9 Pregled obavijesti – korisnik (Valentina Bognolo)

Slika 15. Dijagram aktivnosti za pregled obavijesti za korisnika



Izvor: obrada autora

Identifikacijski sažetak: Pregled svih obavijesti.

Akteri: Korisnik.

Preduvjeti: Autentikacija korisnika u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Korisniku se prikazuje naslovna stranica sa pregledom svih obavijesti.

Alternativne sekvence:

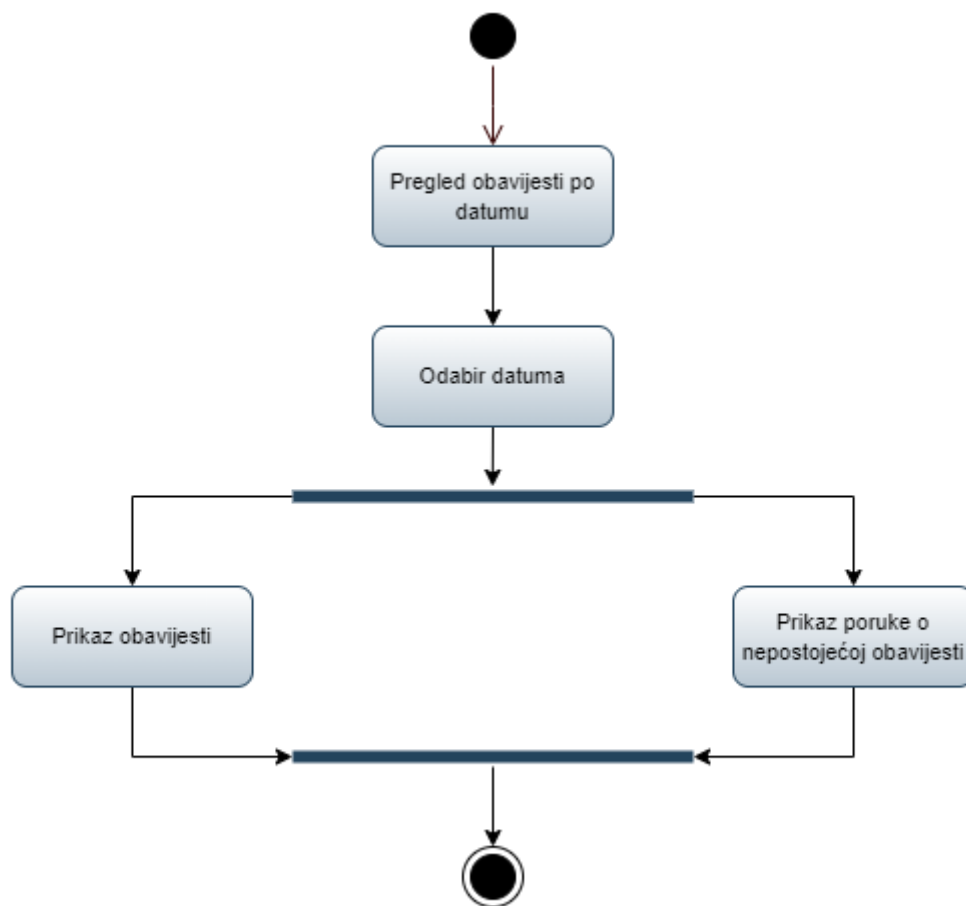
A1: nema

Sekvence s greškom:

E1: nema

4.10 Pregled obavijesti po datumu – korisnik (Valentina Bognolo)

Slika 16. Dijagram aktivnosti za pregled obavijesti po datumu za korisnika



Izvor: obrada autora

Identifikacijski sažetak: Korisnik odabire datum za prikaz obavijesti.

Akteri: Korisnik.

Preduvjeti: Autentikacija korisnika u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Korisnik odabire datum u kalendaru.
2. Sustav ispisuje sve obavijesti odabranog datuma.

Alternativne sekvence:

A1: Ispis nepostojeće obavijesti

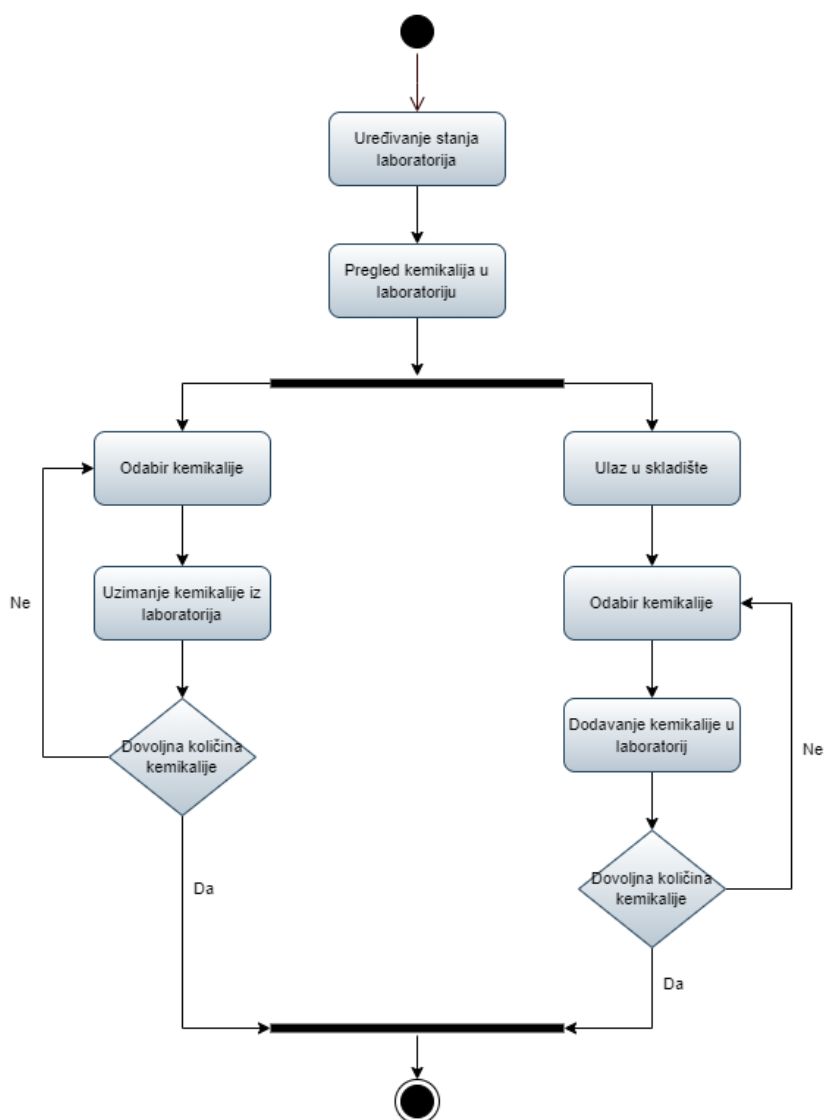
1. Iz koraka G1 – Sustav ispisuje poruku nepostojećih obavijesti odabranog datuma.

Sekvence s greškom:

E1: nema

4.11 Uređivanje stanja laboratorija – korisnik (Valentina Bognolo)

Slika 17. Dijagram aktivnosti za uređivanje stanja laboratorija za korisnika



Izvor: obrada autora

Identifikacijski sažetak: Korisnik uređuje količine kemikalija u laboratoriju i dodaje nove kemikalije iz skladišta.

Akteri: Korisnik.

Preduvjeti: Autentikacija korisnika u sustavu.

Tijek događaja:

Glavni uspješni scenarij (G):

1. Korisnik traži ulaz u laboratorij.
2. Sustav prikazuje popis kemikalija i njihovo stanje u laboratoriju.
3. Korisnik odabire kemikaliju sa popisa.
4. Korisnik unosi količinu kemikalije koju želi dodati ili oduzeti iz laboratorija.
5. Korisnik potvrđuje unos.
6. Sustav pohranjuje promjene.
7. Sustav ažurira pregled stanja laboratorija.
8. Sustav prikazuje nastale promjene u *dashboard*-u na naslovnoj stranici.

Alternativne sekvence:

A1: Dodavanje novih kemikalije iz skladišta

1. Iz koraka G2 – Korisnik traži ulaz u skladište.
2. Sustav prikazuje stanje skladišta.
3. Korisnik odabire kemikaliju sa popisa.
4. Korisnik unosi količinu kemikalije koju želi dodati u laboratorij.
5. Korisnik potvrđuje unos.
6. Sustav pohranjuje nastale promjene.
7. Sustav ažurira pregled stanja laboratorija i skladišta.
8. Sustav prikazuje nastale promjene u *dashboard*-u na naslovnoj stranici.

Sekvence s greškom:

E1: Nedovoljna količina kemikalije – uzimanje kemikalije iz laboratorija

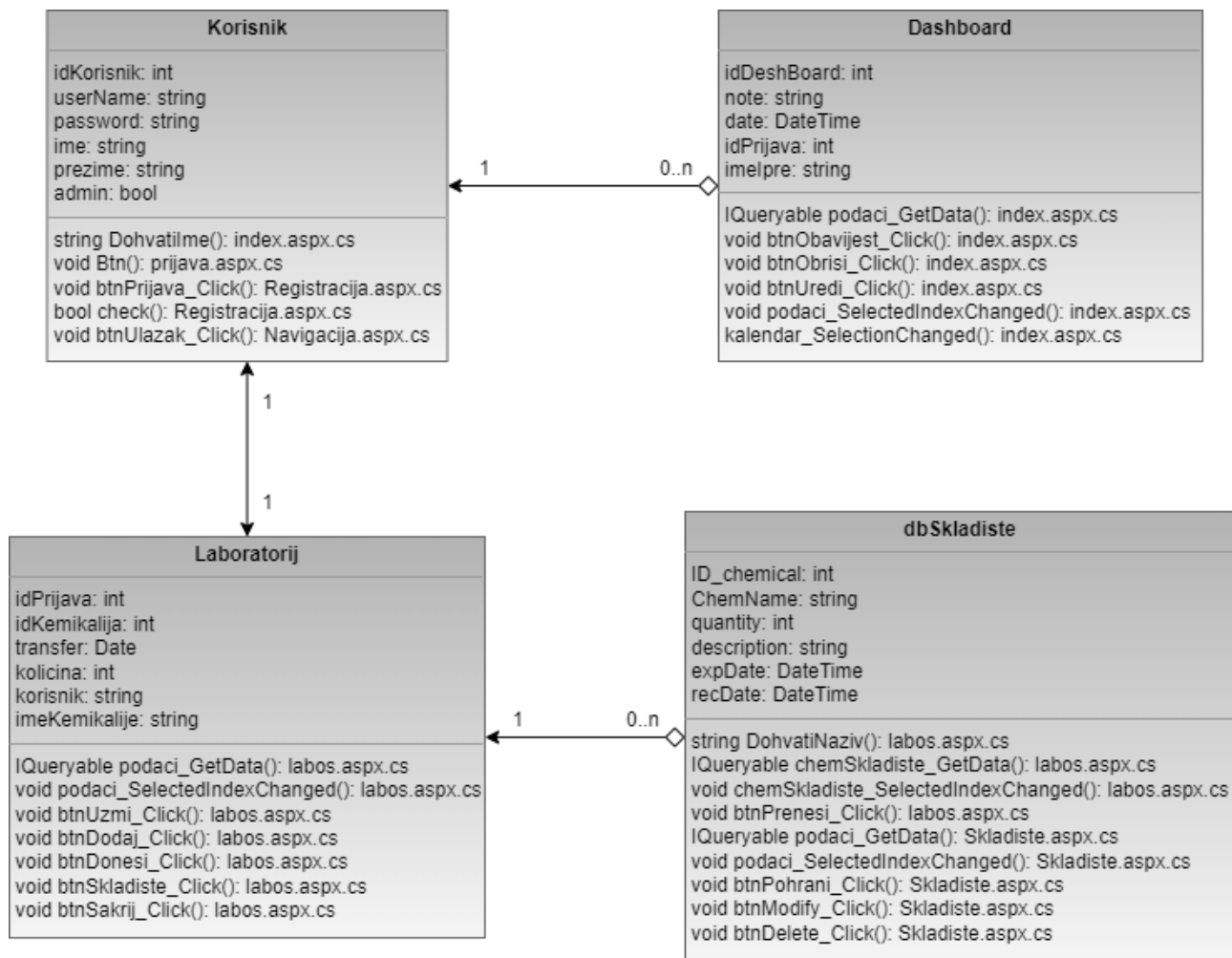
1. Iz koraka G5 – Sustav ispisuje poruku o nedovoljnoj količini odabrane kemikalije.

E2: Nedovoljna količina kemikalije – uzimanje kemikalije iz skladišta

1. Iz koraka A5 – Sustav ispisuje poruku o nedovoljnoj količini odabrane kemikalije.

5. DIJAGRAM KLASA (Valentina Bognolo)

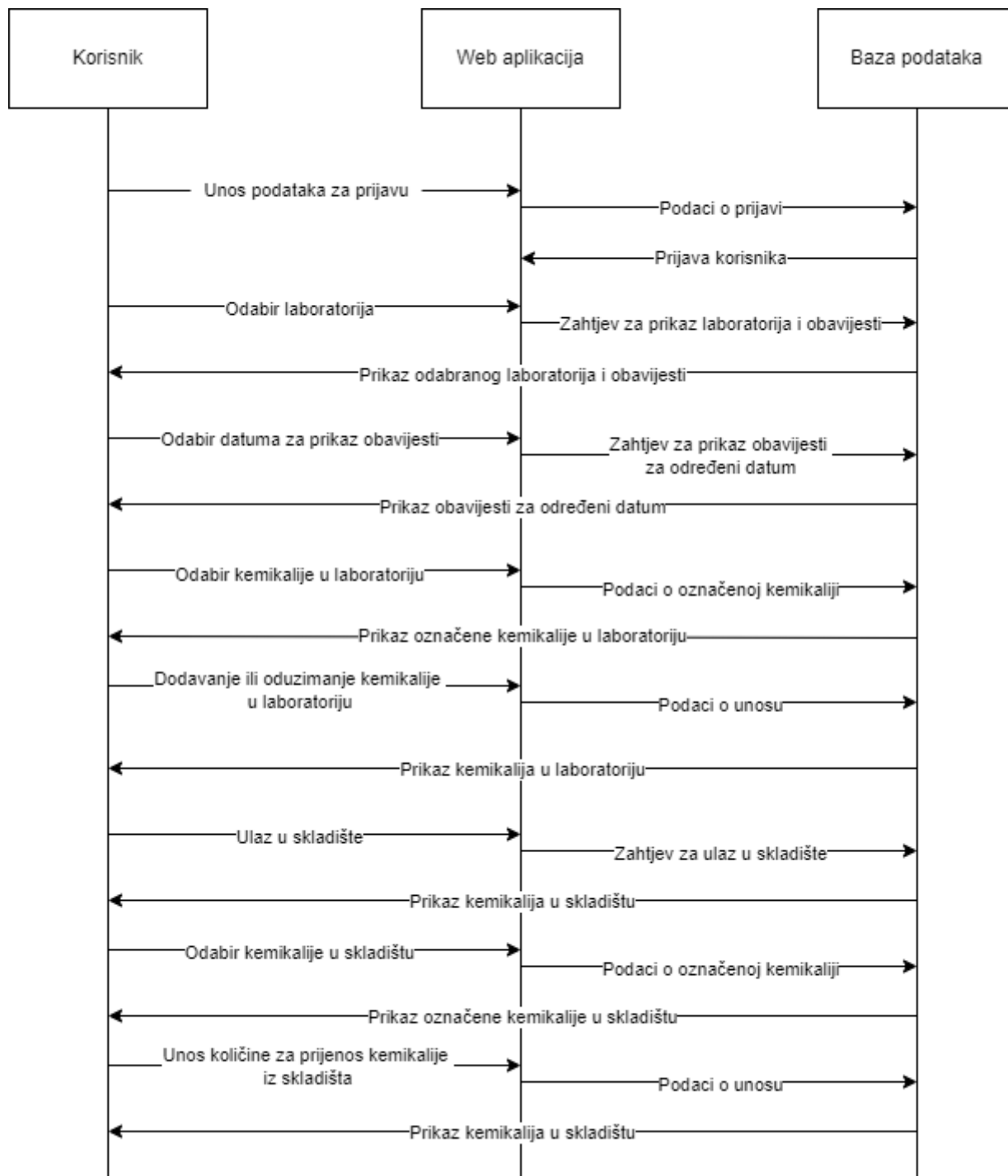
Slika 18. Dijagram klasa



Izvor: obrada autora

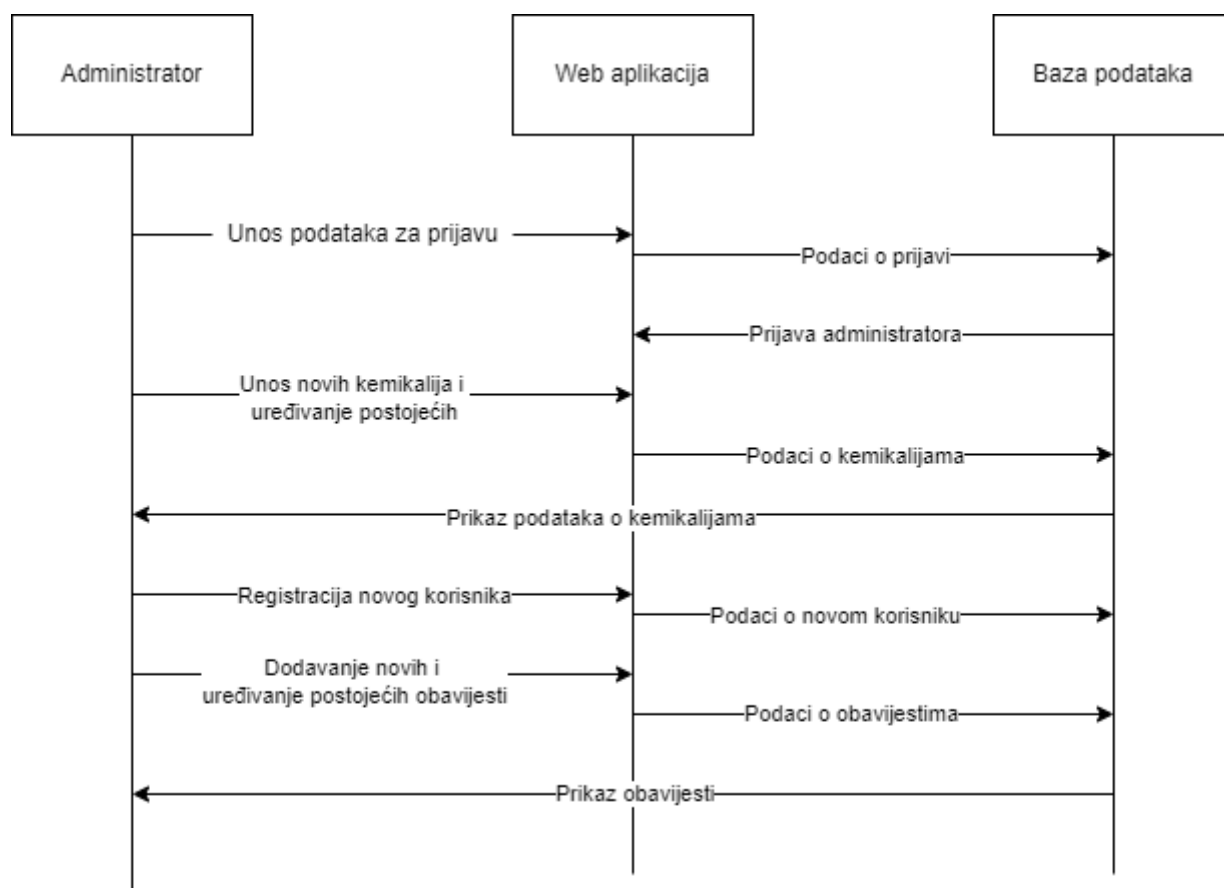
6. SEKVENCIJALNI DIJAGRAM (Fani Tomac)

Slika 19. Sekvencijalni dijagram za korisnika



Izvor: obrada autora

Slika 20. Sekvencijalni dijagram za administratora



Izvor: obrada autora

7. MOCKUP-OVI KORISNIČKIH PROZORA (Fani Tomac)

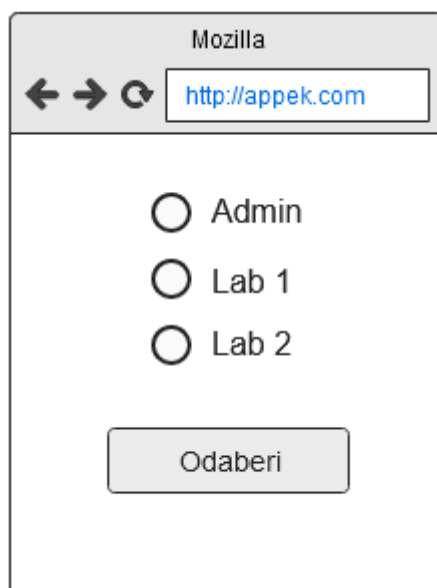
Slika 21. Mockup prozora prijave korisnika



The mockup shows a web browser window titled "Mozilla". The address bar contains the URL "http://apek.com". The main content area has the heading "Prijava" (Login). Below the heading are two input fields: the first is labeled "Korisničko ime" (Username) and the second is labeled "Lozinka" (Password). At the bottom of the form is a button labeled "Prijava" (Login).

Izvor: obrada autora

Slika 22. Mockup prozora za odabir prijave ili laboratorija



The mockup shows a web browser window titled "Mozilla". The address bar contains the URL "http://apek.com". The main content area contains three radio button options: "Admin", "Lab 1", and "Lab 2". At the bottom of the form is a button labeled "Odaberi" (Select).

Izvor: obrada autora

Slika 23. Mockup prozora za pregled i unos obavijesti

Mozilla

←
→
↻

Naslovna
Skladište
Registracija
Odjava

▼ Obavijest	▼ Datum	▼ Korisnik	
Nova poruka	20.04.2022.	Korisnik	<input type="checkbox"/>
Nova poruka	20.04.2022.	Korisnik	<input checked="" type="checkbox"/>
Nova poruka	20.04.2022.	Korisnik	<input type="checkbox"/>
Nova poruka	20.04.2022.	Korisnik	<input type="checkbox"/>

May 05, 2022

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Prikaz obavijesti odabranog datuma

Izvor: obrada autora

Slika 24. Mockup prozora za registraciju korisnika

Mozilla

←
→
↻

Registracija

Korisničko ime

Lozinka

Potvrda lozinke

Ime

Prezime

Izvor: obrada autora

Slika 25. Mockup prozora za skladište i unos nove kemikalije

Mozilla

← → ↻
http://apek.com

Naslovna	Skladište	Registracija	Odjava
----------	-----------	--------------	--------

▼ Kemikalija	▼ Količina	▼ Opis	▼ Datum isteka	▼ Datum zaprimanja	
Kemikalija 1	5	Opis 1	15.10.2040.	20.04.2022.	<input type="checkbox"/>
Kemikalija 2	5	Opis 2	15.10.2040.	20.04.2022.	<input checked="" type="checkbox"/>
Kemikalija 3	5	Opis 3	15.10.2040.	20.04.2022.	<input type="checkbox"/>
Kemikalija 4	5	Opis 4	15.10.2040.	20.04.2022.	<input type="checkbox"/>

Nova kemikalija

Naziv

Količina

Opis

Rok trajanja

Datum zaprimanja

Izvor: obrada autora

Slika 26. Mockup prozora za evidenciju stanja laboratorija

Mozilla

← → ↻
http://apek.com

Naslovna	Laboratorij	Odjava
----------	-------------	--------

▼ Kemikalija	▼ Količina	▼ Korisnik	
Kemikalija 1	2	Korisnik 1	<input type="checkbox"/>
Kemikalija 2	1	Korisnik 2	<input checked="" type="checkbox"/>
Kemikalija 3	1	Korisnik 1	<input type="checkbox"/>
Kemikalija 4	1	Korisnik 1	<input type="checkbox"/>

Izvor: obrada autora

Slika 27. Mockup prozora za pristup skladištu iz laboratorija

Mozilla

← → ↻

<http://apek.com>

Skladište

Izlaz

▼ Kemikalija	▼ Količina	▼ Opis	▼ Datum isteka	▼ Datum zaprimanja	
Kemikalija 1	5	Opis 1	15.10.2040.	20.04.2022.	<input type="checkbox"/>
Kemikalija 2	5	Opis 2	15.10.2040.	20.04.2022.	<input checked="" type="checkbox"/>
Kemikalija 3	5	Opis 3	15.10.2040.	20.04.2022.	<input type="checkbox"/>
Kemikalija 4	5	Opis 4	15.10.2040.	20.04.2022.	<input type="checkbox"/>

Količina označene kemikalije

Prenesi

Izvor: obrada autora

8. MAPIRANJE (Valentina Bognolo)

Struktura aplikacije sastoji se od više datoteka od kojih je važno izdvojiti datoteku za korištenu bazu podataka, mapu koja sadrži korištene klase te datoteke za naslovnu stranicu, prijavu, navigaciju, skladište, registraciju i laboratorij.

Komponenta **DBConnect.cs** omogućuje povezivanje i rad s bazom podataka te sadrži metode *Initialize()*, *OpenConnection()*, *CloseConnection()*, *Insert()*, *Update()*, *Delete()* i *Select()*.

Slika 28. Prikaz koda za inicijalizaciju baze podataka

```
private void Initialize()
{
    server = "ucka.veleri.hr";
    database = "vbognolo";
    uid = "vbognolo";
    password = "11";

    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
        database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    connection = new MySqlConnection(connectionString);
}
```

Izvor: obrada autora

Slika 29. Prikaz koda za otvaranje konekcije prema bazi podataka

```
private string OpenConnection()
{
    try
    {
        connection.Open();
        return "";
    }
    catch (MySqlException ex)
    {
        switch (ex.Number)
        {
            case 0:
                return "Nije moguće spojiti se na server.";
            case 1045:
                return "Neispravno korisničko ime ili lozinka.";
        }
        return "Greška";
    }
}
```

Izvor: obrada autora

Slika 30. Prikaz koda za zatvaranje konekcije prema bazi podataka

```
public bool CloseConnection()
{
    try
    {
        connection.Close();
        return true;
    }
    catch (MySqlException ex)
    {
        return false;
    }
}
```

Izvor: obrada autora

Slika 31. Prikaz koda za funkciju Insert()

```
public void Insert(string Sql)
{
    if (this.OpenConnection() == "")
    {
        MySqlCommand mySql = new MySqlCommand(Sql, connection);
        mySql.ExecuteNonQuery();
        this.CloseConnection();
    }
}
```

Izvor: obrada autora

Slika 32. Prikaz koda za funkciju Update()

```
public void Update(string sql)
{
    if (this.OpenConnection() == "")
    {
        MySqlCommand cmd = new MySqlCommand(sql, connection);
        cmd.ExecuteNonQuery();
        this.CloseConnection();
    }
}
```

Izvor: obrada autora

Slika 33. Prikaz koda za funkciju Delete()

```
public void Delete(string sql)
{
    if (this.OpenConnection() == "")
    {
        MySqlCommand cmd = new MySqlCommand(sql, connection);
        cmd.ExecuteNonQuery();
        this.CloseConnection();
    }
}
```

Izvor: obrada autora

Slika 34. Prikaz koda za funkciju Select()

```
public MySqlDataReader Select(string sql)
{
    if(this.OpenConnection() == "")
    {
        MySqlCommand cmd = new MySqlCommand(sql, connection);
        MySqlDataReader data = cmd.ExecuteReader();
        return data;
    }

    return null;
}
```

Izvor: obrada autora

Komponenta **index.aspx.cs** predstavlja naslovnu stranicu aplikacije i sadrži metode *DohvatiIme()*, *podaci_GetData()*, *btnObavijest_Click()*, *btnObrisi_Click()*, *btnUredi_Click()*, *podaci_SelectedIndexChanged()* i *kalendar_SelectionChanged()*.

Slika 35. DohvatiIme() - Dohvaćanje imena i prezimena korisnika

```
public string DohvatiIme(int serial)
{
    DBConnect connect = new DBConnect();

    MySqlDataReader imePre = connect.Select("SELECT ime, prezime FROM Prijava where ID_prijava = " + serial);

    string naziv;
    if (imePre.Read())
    {
        naziv = String.Format("{0} {1}", imePre["ime"], imePre["prezime"]);
        connect.CloseConnection();
    }
    else
        naziv = "";

    return naziv;
}
```

Izvor: obrada autora

Slika 36. Metode za unos novih i upravljanje postojećim obavijestima

```
protected void btnObavijest_Click(object sender, EventArgs e)
{
    dbConnect.Insert(String.Format("INSERT INTO Dashboard (note, date, ID_prijava) " +
        "VALUES ('{0}', '{1}', {2})",
        txtObavijest.Text.Trim(),
        FormatDateForMySQL(DateTime.Now),
        korisnik.IdKorisnik));

    Response.Redirect("index.aspx", false);
}

0 references
protected void btnObrisi_Click(object sender, EventArgs e)
{
    string id2 = Session["idObavijesti"].ToString();

    dbConnect.Delete("DELETE FROM Dashboard WHERE ID_dashboard = " + id2);

    Response.Redirect("index.aspx", false);
}

0 references
protected void btnUredi_Click(object sender, EventArgs e)
{
    dbConnect.Update(String.Format("UPDATE Dashboard " +
        "SET note = '{0}' " +
        "WHERE ID_dashboard = {1}",
        txtObavijest.Text,
        Session["idObavijesti"].ToString()));

    Response.Redirect("index.aspx", false);
}
```

Izvor: obrada autora

Slika 37. `kalendar_SelectionChanged()` - Prikaz obavijesti po datumu

```
protected void kalendar_SelectionChanged(object sender, EventArgs e)
{
    DateTime date = kalendar.SelectedDate;
    Session["datum"] = date;
    listBoard = podaci_GetData().ToList<Dashboard>();
    string obavijest = "";

    foreach (Dashboard dashboard in listBoard)
        if (dashboard.Date.Equals(date))
            obavijest += dashboard.ImeIpri + ": " + dashboard.Note + "\n";

    if (!obavijest.Equals(String.Empty))
        txtObavDatum.Text = obavijest;
    else
        txtObavDatum.Text = "Nije pronađena obavijest pod tim datumom.";
}
```

Izvor: obrada autora

Komponenta *prijava.aspx.cs* predstavlja stranicu prijave korisnika i sadrži metodu *Btn()*.

Slika 38. *Btn()* - Prijava korisnika

```
protected void Btn(object sender, EventArgs e)
{
    string user = txtKorIme.Text;
    string pass = txtLozinka.Text;

    MySqlDataReader data = baza.Select("SELECT * FROM Prijava where username = '"+user+"' ");

    if(data.Read())
    {
        if (data["password"].ToString() == pass)
        {
            korisnik.IdKorisnik = int.Parse(data["ID_prijava"].ToString());
            korisnik.UserName = data["username"].ToString();
            korisnik.Password = data["password"].ToString();
            korisnik.Ime = data["ime"].ToString();
            korisnik.Prezime = data["prezime"].ToString();
            korisnik.Admin = bool.Parse(data["admin"].ToString());

            baza.CloseConnection();

            Session["user"] = korisnik;
            Response.Redirect("Navigacija.aspx");
        }
        else
            lblLozinka.Text = "Lozinka nije ispravna";
    }
    else
        lblKorisnik.Text = "Korisnik nije u bazi";
}
```

Izvor: obrada autora

Komponenta *Navigacija.aspx.cs* predstavlja stranicu odabira prijave za administratora i odabira laboratorija za korisnika te sadrži funkciju *btnUlazak_Click()*.

Slika 39. *btnUlazak_Click()* - Odabir prijave ili laboratorija

```
protected void btnUlazak_Click(object sender, EventArgs e)
{
    if (rbNavigacija.SelectedIndex == 0)
    {
        Session["admin"] = true;
        Response.Redirect("index.aspx");
    }
    else if (rbNavigacija.SelectedIndex == 1)
    {
        Session["admin"] = false;
        Session["odabraniLab"] = "Lab1";

        Response.Redirect("index.aspx");
    }
    else
    {
        Session["admin"] = false;
        Session["odabraniLab"] = "Lab2";

        Response.Redirect("index.aspx");
    }
}
```

Izvor: obrada autora

Komponenta **Skladiste.aspx.cs** predstavlja stranicu skladišta i sadrži metode *podaci_GetData()*, *podaci_SelectedIndexChanged()*, *btnPohrani_Click()*, *btnModify_Click()* i *btnDelete_Click()*.

Slika 40. *podaci_GetData()* - Dohvaćanje podataka skladišta iz baze podataka

```
public IQueryable<APPEK2.Models.dbSkladiste> podaci_GetData()
{
    MySqlDataReader data = dbConnect.Select("SELECT * FROM Skladiste");

    while (data.Read())
    {
        skladiste = new dbSkladiste();

        skladiste.ID_chemical1 = int.Parse(data["ID_chemical"].ToString());
        skladiste.ChemName1 = data["chemName"].ToString();
        skladiste.Quantity = int.Parse(data["quantity"].ToString());
        skladiste.Description = data["description"].ToString();
        skladiste.ExpDate = DateTime.Parse(Date.Parse(data["expDate"].ToString()).ToString());
        skladiste.RecDate = DateTime.Parse(Date.Parse(data["recDate"].ToString()).ToString());

        listSkladiste.Add(skladiste);
    }

    dbConnect.CloseConnection();

    return listSkladiste.AsQueryable();
}
```

Izvor: obrada autora

Slika 41. *podaci_SelectedIndexChanged()* - Funkcija za dohvaćanje indeksa reda kemikalija

```
protected void podaci_SelectedIndexChanged(object sender, EventArgs e)
{
    int id = podaci.SelectedIndex;
    listSkladiste = podaci_GetData().ToList<dbSkladiste>();
    skladiste = listSkladiste[id];
    Session["kemikalija"] = skladiste;

    txtNaziv.Text = skladiste.ChemName1;
    txtKolicina.Text = skladiste.Quantity.ToString();
    txtOpis.Text = skladiste.Description;
    dateExp.SelectedDate = skladiste.ExpDate;
    dateRec.SelectedDate = skladiste.RecDate;
}
```

Izvor: obrada autora

Slika 42. Metode za unos novih i upravljanje postojećih kemikalija

```
protected void btnPohrani_Click(object sender, EventArgs e)
{
    dbConnect.Insert(String.Format("INSERT INTO Skladiste (chemName, quantity, description, expDate, recDate)" +
        "VALUES ('{0}', {1}, '{2}', '{3}', '{4}')" +
        txtNaziv.Text,
        txtKolicina.Text,
        txtOpis.Text,
        FormatDateForMySQL((DateTime)Session["dateExp"]),
        FormatDateForMySQL((DateTime)Session["dateRec"])));

    Response.Redirect("Skladiste.aspx", false);
}

References
protected void btnModify_Click(object sender, EventArgs e)
{
    skladiste = (dbSkladiste)Session["kemikalija"];

    dbConnect.Update(String.Format("UPDATE Skladiste SET chemName = '{0}', quantity = {1}, description = {2}, expDate = '{3}', recDate = '{4}' WHERE ID_chemical = {5}",
        txtNaziv.Text,
        txtKolicina.Text,
        txtOpis.Text,
        FormatDateForMySQL((DateTime)Session["dateExp"]),
        FormatDateForMySQL((DateTime)Session["dateRec"]),
        skladiste.ID_chemical));

    Response.Redirect("Skladiste.aspx", false);
}

References
protected void btnDelete_Click(object sender, EventArgs e)
{
    skladiste = (dbSkladiste)Session["kemikalija"];

    dbConnect.Delete("DELETE FROM Skladiste WHERE ID_chemical = " + skladiste.ID_chemical);

    Response.Redirect("Skladiste.aspx", false);
}
```

Izvor: obrada autora

Komponenta **Registracija.aspx.cs** predstavlja stranicu registracije korisnika i sadrži metode *btnPrijava_Click()* i *check()*.

Slika 43. *btnPrijava_Click()* - Registracija korisnika

```
protected void btnPrijava_Click(object sender, EventArgs e)
{
    MySqlDataReader data = baza.Select("SELECT * FROM Prijava where username = '" + txtKorIme.Text + "'");

    if (data.Read())
    {
        lblkorisnik.Text = "korisnik postoji";
        lblneisloz.Text = "";
    }
    else
    {
        baza.CloseConnection();

        if (check())
        {
            if (txtLozinka.Text.Equals(txtponovi.Text))
            {
                baza.Insert("INSERT into Prijava(username, password, ime, prezime) VALUES ('" + txtKorIme.Text + "', '" + txtLozinka.Text + "', '" + txtIme.Text + "', '" + txtPrezime.Text + "')");
                MessageBox.Show("Uspješan unos");
            }
            else
            {
                lblneisloz.Text = "Lozinke se moraju podudarati";
            }
        }
        else
        {
            MessageBox.Show("Popunite sve podatke");
        }
    }
}
```

Izvor: obrada autora

Komponenta **Labos.aspx.cs** predstavlja stranicu laboratorija i sadrži metode *podaci_GetData()*, *DohvatiKorisnika()*, *DohvatiNaziv()*, *btnUzmi_Click()*, *btnDodaj_Click()*, *btnPrenesi_Click()*, *btnSkladiste_Click()*, *btnSakrij_Click()*, *podaci_SelectedIndexChanged()*, *chemSkladiste_GetData()*, *chemSkladiste_SelectedIndexChanged()*.

Slika 44. Metode za dohvaćanje imena korisnika i naziva kemikalije

```
public string DohvatiKorisnika(int serial)
{
    DBConnect connect = new DBConnect();

    MySqlDataReader imePre = connect.Select("SELECT ime, prezime FROM Prijava WHERE ID_prijava = " + serial);

    string naziv;
    if (imePre.Read())
        naziv = String.Format("{0} {1}", imePre["ime"], imePre["prezime"]);
    else
        naziv = "";

    connect.CloseConnection();
    return naziv;
}

1reference
public string DohvatiNaziv(int serial)
{
    DBConnect connectNaziv = new DBConnect();
    string naziv;

    MySqlDataReader kemikalija = connectNaziv.Select("SELECT chemName FROM Skladiste WHERE ID_chemical = " + serial);

    if (kemikalija.Read())
        naziv = kemikalija["chemName"].ToString();
    else
        naziv = "";

    connectNaziv.CloseConnection();
    return naziv;
}
```

Izvor: obrada autora

Slika 45. btnUzmi_Click() - Uzimanje kemikalije iz laboratorija

```
protected void btnUzmi_Click(object sender, EventArgs e)
{
    int novoStanje = int.Parse(Session["quantity"].ToString()) - int.Parse(txtKom.Text);

    if (novoStanje < 0)
        MessageBox.Show("U labosu nema dovoljno resursa!");

    else
    {
        DBConnect chemName = new DBConnect();

        MySqlDataReader kemikalija = chemName.Select("SELECT chemName FROM Skladiste WHERE ID_chemical = " + int.Parse(Session["idChem"].ToString()));

        if (kemikalija.Read())
        {
            dbConnect.Update(String.Format("UPDATE {0} SET quantity = {1} WHERE ID_chemical = {2}", Session["odabraniLab"].ToString(), novoStanje, int.Parse(Session["idChem"].ToString())));
            dbConnect.Insert(String.Format("INSERT INTO Dashboard (note, date, ID_prijava) VALUES ('{0} je uzeo/la {1} kom. {2} iz {3}', '{4}', {5})",
                korisnik.Ime + " " + korisnik.Prezime,
                int.Parse(Session["quantity"].ToString()),
                kemikalija["chemName"].ToString(),
                Session["odabraniLab"].ToString(),
                FormatDateForMySQL(DateTime.Now),
                korisnik.IdKorisnik));
        }

        chemName.CloseConnection();
        Response.Redirect("Labos.aspx", false);
    }
}
```

Izvor: obrada autora

Slika 46. btnDodaj_Click() - Dodavanje kemikalije u laboratorij

```
protected void btnDodaj_Click(object sender, EventArgs e)
{
    int novoStanje = int.Parse(Session["quantity"].ToString()) + int.Parse(txtKom.Text);
    MySqlDataReader provjeriSkladiste = dbConnect.Select(String.Format("SELECT * FROM Skladiste where ID_chemical = {0} ", int.Parse(Session["idChem"].ToString())));

    if (provjeriSkladiste.Read())
    if (int.Parse(txtKom.Text) > int.Parse(provjeriSkladiste["quantity"].ToString()))
        MessageBox.Show("U skladištu nema dovoljno zaliha!");

    else
    {
        int novoStanjeSkaldiste = int.Parse(provjeriSkladiste["quantity"].ToString()) - int.Parse(txtKom.Text);
        DBConnect connect = new DBConnect();
        connect.Update(String.Format("UPDATE {0} SET quantity = {1} WHERE ID_chemical = {2}", Session["odabranilab"].ToString(), novoStanje, int.Parse(Session["idChem"].ToString())));
        connect.Update(String.Format("UPDATE Skladiste SET quantity = {0} WHERE ID_chemical = {1}",
            novoStanjeSkaldiste,
            int.Parse(Session["idChem"].ToString())));
        connect.Insert(String.Format("INSERT INTO Dashboard (note, date, ID_prijava) VALUES ('Dodano je novih {0} kom. {1} u {2}', '{3}', {4})",
            Session["quantity"].ToString(),
            provjeriSkladiste["chemName"].ToString(),
            Session["odabranilab"].ToString(),
            FormatDateForMySQL(DateTime.Now),
            korisnik.IdKorisnik));
        MessageBox.Show("Zalihe laboratorija su obnovljene!");
        dbConnect.CloseConnection();
    }

    Response.Redirect("Labos.aspx", false);
}
```

Izvor: obrada autora

Slika 47. btnPrenesi_Click() - Prijenos kemikalije iz skladišta u laboratorij

```
protected void btnPrenesi_Click(object sender, EventArgs e)
{
    int idChemSk1 = int.Parse(Session["idChemSkaldiste"].ToString());
    int quantitySk1 = int.Parse(Session["quantitySkaldiste"].ToString());
    int quantity = int.Parse(txtPrenesi.Text);

    if (quantitySk1 - quantity < 0)
        MessageBox.Show("Nema dovoljno zaliha u skladištu");

    else
    {
        dbConnect.Update(String.Format("UPDATE Skladiste SET quantity = {0} WHERE ID_chemical = {1}", quantitySk1 - quantity, idChemSk1));

        listLabos = podaci_GetData().ToList<Laboratorij>();
        bool postoji = false;
        int quantityLab = 0;
        string nazivChem = "";

        foreach (Laboratorij lab in listLabos)
        if (lab.IdKemikalija.Equals(idChemSk1))
        {
            postoji = true;
            quantityLab = lab.Kolicina;
            nazivChem = lab.ImeKemikalije;
        }

        if (postoji)
        {
            dbConnect.Update(String.Format("UPDATE {0} SET quantity = {1} WHERE ID_chemical = {2}",
                Session["odabranilab"].ToString(), quantityLab + quantity, idChemSk1));
            MessageBox.Show("Dodano je još " + quantity + " mjernih jedinica " + nazivChem);

            dbConnect.Insert(String.Format("INSERT INTO Dashboard (note, date, ID_prijava) VALUES ('{0}', '{1}', {2})",
                "Preneseno je " + quantity + " komada " + nazivChem + " iz skladišta u " + Session["Odabranilab"] + " ",
                FormatDateForMySQL(DateTime.Now),
                korisnik.IdKorisnik));
        }
        else
        {
            dbConnect.Insert(String.Format("INSERT INTO {0} VALUES ({1}, {2}, '{3}', {4})",
                Session["odabranilab"].ToString(), korisnik.IdKorisnik, idChemSk1,
                FormatDateForMySQL(DateTime.Now), quantity));
            MessageBox.Show("U labos je pohranjena nova kemikalija");

            dbConnect.Insert(String.Format("INSERT INTO Dashboard (note, date, ID_prijava) VALUES ('{0}', '{1}', {2})",
                "Donesena je nova kemikalija, " + quantity + " komada " + nazivChem + " u " + Session["Odabranilab"],
                FormatDateForMySQL(DateTime.Now),
                korisnik.IdKorisnik));
        }
    }

    Response.Redirect("Labos.aspx");
}
```

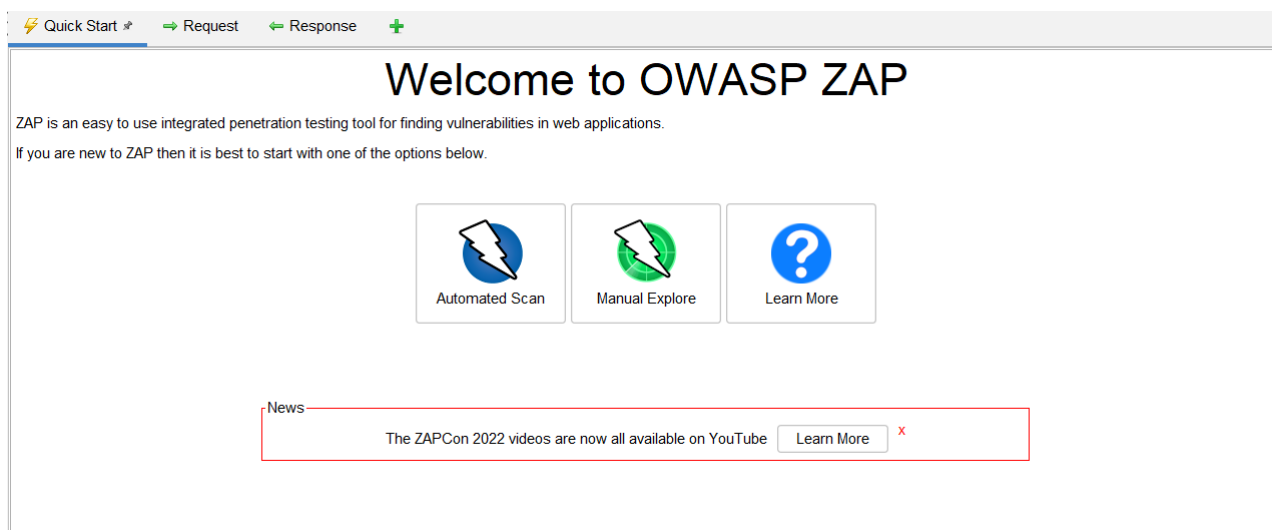
Izvor: obrada autora

9. TESTIRANJE SIGURNOSTI (Fani Tomac)

Za test sigurnosti odabrale smo alat OWASP ZAP (skraćenica od Zed Attack Proxy). To je sigurnosni skener web aplikacija otvorenog koda. Namijenjen je i onima koji su novi u sigurnosti aplikacija, ali i profesionalnim testerima penetracije. Izabrale smo ovaj alat zato što je pristupačan za nove korisnike i jednostavno se koristi.

Nakon uspješnog preuzimanja i jednostavne instalacije programa s njihove službene stranice (<https://owasp.org/www-project-zap/>), ponuđene su dvije opcije izvršavanja testiranja – ručno testiranje (engl. *Manual Explore*) i automatsko testiranje (engl. *Automated Scan*). Mi smo odabrale opciju automatskog testiranja jer smatramo da ćemo tako dobiti kvalitetnije rezultate.

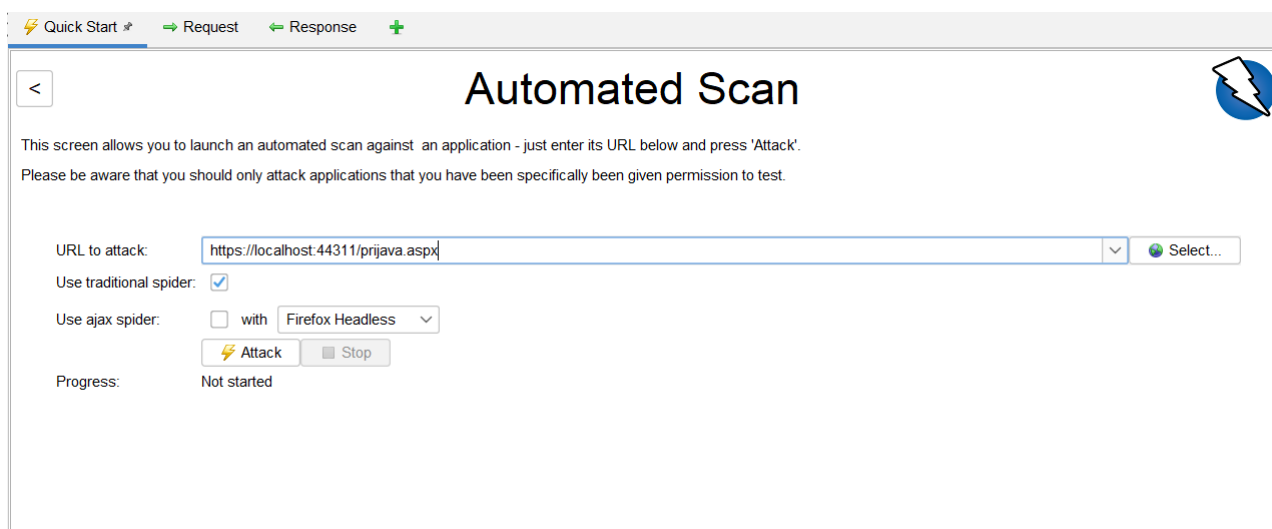
Slika 48. Naslovna stranica ZAP alata



Izvor: obrada autora

Nakon odabira načina testiranja, u sljedećem izborniku, bilo je potrebno dodati poveznicu na našu web aplikaciju. Aplikaciju smo pokrenule preko VS-a na localhost-u, nakon toga smo unijele poveznicu na našu web aplikaciju i odabrale opciju *Attack*.

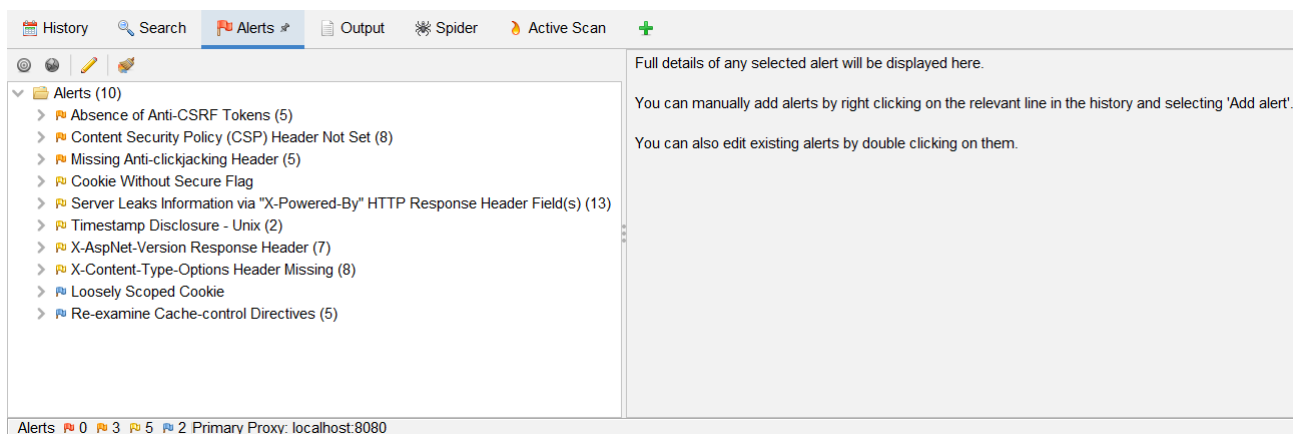
Slika 49. Dodavanje poveznice web aplikacije za automatsko skeniranje



Izvor: obrada autora

Nakon što je testiranje završeno, pojave se rezultati testiranja. Ukupno su skenirane ranjivosti s tri različita stupnja upozorenja, a boje zastavice označavaju razinu ranjivosti. U našem slučaju pojavilo se 10 prijetnji, od kojih su 3 srednjeg stupnja ranjivosti (označeno jačom narančastom bojom), 5 su niskog stupnja ranjivosti (označeno žutom bojom), a 2 su informacijske pojave rizika (označene plavom zastavicom) koje prema procjeni skenera ne utječu na sigurnost aplikacije.

Slika 50. Prikaz rezultata testiranja



Izvor: obrada autora

Prijetnje srednje razine obuhvaćaju:

- *Absence of Anti-CSRF Tokens* - upozorenje koje nam javlja da ne koristimo provjerenu knjižicu ili okvir koji ne dopušta pojavu ove slabosti.
 - Rješenje je upotreba anti-CSRF paketa (kao što je OWASP CSRFGurd) koji ne dopuštaju prijetnji da izvrši napad na sustav.

- *Content Security Policy (CSP) Header Not Set* - politika sigurnosti sadržaja (CSP) dodatni je sloj sigurnosti koji pomaže u otkrivanju i ublažavanju određenih vrsta napada, uključujući skriptiranje na više web-mjesta i napade ubrizgavanjem podataka. Ti se napadi koriste za sve, od krađe podataka do uništavanja stranice ili distribucije zlonamjernog softvera. CSP pruža skup standardnih HTTP zaglavlja koji vlasnicima web-mjesta omogućuju da deklariraju odobrene izvore sadržaja koje bi preglednicima trebalo dopustiti učitavanje na tu stranicu - pokrivene vrste su JavaScript, CSS, HTML okviri, fontovi, slike i objekti koji se mogu ugraditi kao što su Java apleti, ActiveX, audio i video datoteke.
- *Cross-Domain Misconfiguration* - označava prijetnju koja upozorava da nemamo instaliran nikakav sustav obrane protiv CSRF napada koji može biti vrlo opasan za korisnika.
- *Missing Anti-clickjacking Header* - odgovor ne uključuje niti Content-Security-Policy s direktivom 'frame-ancestors' niti X-Frame-Options za zaštitu od 'ClickJacking' napada.
 - Moderni web preglednici podržavaju HTTP zaglavlja Content-Security-Policy i X-Frame-Options. Potrebno je provjeriti je li jedan od njih postavljen na svim web-stranicama koje vraća web-lokacija/aplikacija.

Prijetnje niske razine obuhvaćaju:

- *Cookie Without Secure Flag* - kolačić je postavljen bez sigurnosne zastavice, što znači da se kolačiću može pristupiti putem nekriptiranih veza.
 - Rješenje je postaviti sigurnosnu zastavicu za kolačiće.
- *Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)* - također niski rizik koji govori kako bi mogli imati proboj informacija iz baze podataka.
- *Timestamp Disclosure – Unix* – web server pruža timestamp, odnosno trenutno vrijeme po Unix standardu.
- *X-AspNet-Version Response Header* - poslužitelj propušta informacije putem "X-AspNet-Version"/"X-AspNetMvc-Version" polja HTTP zaglavlja odgovora.
 - Moguće rješenje je konfigurirati server kako ne bi vraćao ta zaglavlja.
- *X-Content-Type-Options Header Missing*- greška niske razine koja ukazuje da ne postoji nikakav x-content (sprječava neželjeno „njuškanje“ od strane potencijalnog napadača).

Prijetnje informacijske razine obuhvaćaju:

- *Loosely Scoped Cookie* - kolačići se mogu odrediti po domeni ili stazi. Ova provjera se odnosi samo na opseg domene. Opseg domene primijenjen na kolačić određuje koje domene mogu pristupiti.

- *Re-examine Cache-control Directives* - zaglavlje kontrole predmemorije nije ispravno postavljeno ili nedostaje, što omogućuje pregledniku i proxy-jima da spremaju predmemoriju sadržaja. Za statične materijale kao što su .css, .js ili slikovne datoteke to bi moglo biti namijenjeno, međutim, resurse bi trebalo pregledati kako bi se osiguralo da se osjetljivi sadržaj neće spremati u predmemoriju.

Napominjemo da prijetnje identificirane ZAP-om u ovoj verziji aplikacije nisu riješene.

10. VALIDACIJA SPECIFIKACIJE ZAHTJEVA (Valentina Bognolo)

Tablica 1. Validacija funkcionalnosti

Funkcionalnost	Akteri	Dovršeno	U tijeku	Nedovršeno
Prijava	Administrator, korisnik			
Odabir prijave	Administrator, korisnik			
Upravljanje obavijestima	Administrator			
Pregled obavijesti	Administrator, korisnik			
Pregled obavijesti po datumu	Administrator, korisnik			
Evidencija stanja skladišta	Administrator			
Registracija korisnika	Administrator			
Pregled stanja laboratorija	Korisnik			
Uređivanje stanja laboratorija	Korisnik			

Izvor: obrada autora

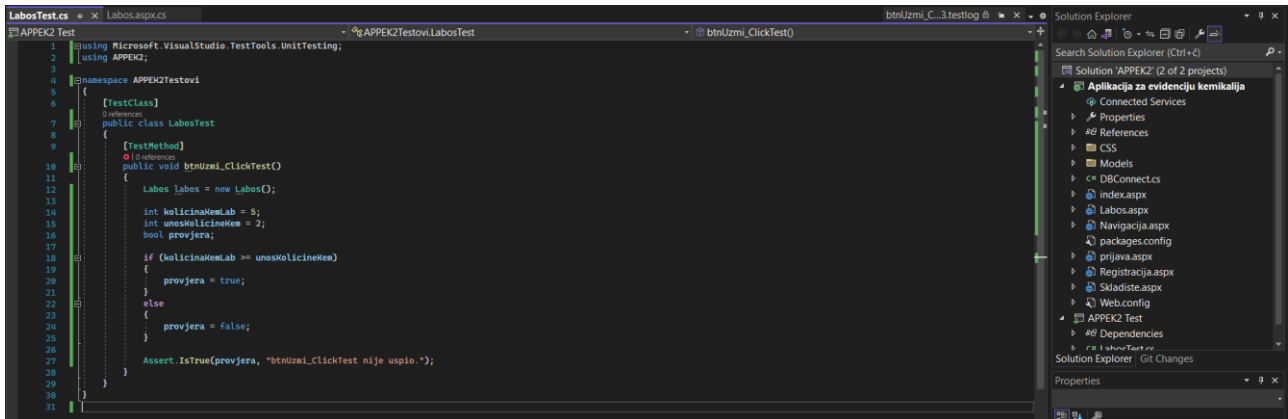
11. SUDJELOVANJE NA PROJEKTU

Tim za izgradnju projekta sastoji se od dva člana, Bognolo Valentine i Tomac Fani. Izrada projekta podijelila se međusobnim dogovorom u 3 kategorije:

1. Izgradnja baze podataka na phpmyadmin-u (Bognolo) i postavljanje GitHub-a (Tomac)
2. Izgradnja programskog rješenja
 - a. Modeli - Dashboard.cs, dbSkladiste.cs, Korisnik.cs, Laboratorij.cs (Bognolo)
 - b. Klasa i metode za bazu podataka - DBConnect.cs (Tomac)
 - c. Front-end naslovne stranice - index.aspx (Tomac)
 - d. Back-end naslovne stranice - index.aspx.cs (Bognolo)
 - e. Front-end stranice laboratorija - Labos.aspx (Bognolo)
 - f. Back-end stranice laboratorija - Labos.aspx.cs (Bognolo)
 - g. Front-end stranice odabira prijave - Navigacija.aspx (Bognolo)
 - h. Back-end stranice odabira prijave - Navigacija.aspx.cs (Tomac)
 - i. Front-end stranice prijave - prijava.aspx (Bognolo)
 - j. Back-end stranice prijave - prijava.aspx.cs (Bognolo)
 - k. Front-end stranice registracije - Registracija.aspx (Tomac)
 - l. Back-end stranice registracije - Registracija.aspx.cs (Tomac)
 - m. Front-end stranice skladišta - Skladiste.aspx (Tomac)
 - n. Back-end stranice skladišta - Skladiste.aspx.cs (Tomac)
 - o. CSS - index.css, labos.css, navigacija.css, prijava.css, registracija.css, skladiste.css (Bognolo, Tomac)
 - p. Testiranje sigurnosti (Bognolo, Tomac)
 - q. Testiranje komponenti (Bognolo, Tomac) - nedovršeno
3. Izrada dokumentacije (Bognolo, Tomac)

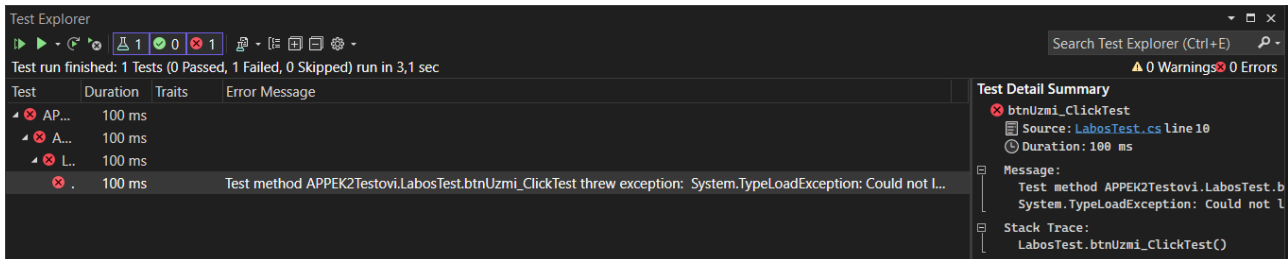
Iako se na početku napravila podjela zadataka za izgradnju programskog rješenja, realizacija je napravljena međusobnim sudjelovanjem u svim dijelovima programskog koda. Greške i problemi nastali kroz razvoj aplikacije, savladani su zajedničkim istraživanjem i isprobavanjem mogućih rješenja. Pokušalo se napraviti i testiranje komponenti implementiranjem programskog koda za testiranje klasa i metoda međutim, isprobani test nije bio uspješan te se nije uspjelo implementirati odgovarajuće rješenje.

Slika 51. Test metode btnUzmi_ClickTest()



Izvor: obrada autora

Slika 52. Neuspješno izvedeno testiranje



Izvor: obrada autora

Literatura i izvori

- <https://stackoverflow.com/>
- <https://www.w3schools.com/>
- https://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- <https://stackify.com/iis-web-server/>
- <https://docs.microsoft.com/en-us/dotnet/csharp/>
- <https://en.wikipedia.org/wiki/ASP.NET>
- <https://visualstudio.microsoft.com/vs/>
- <https://github.com/>
- <https://www.zaproxy.org/>
- <https://www.outsystems.com/forums/>

Popis slika

Slika 1. Arhitektura sustava	6
Slika 2. Prikaz projekta povezanog s Githubom.....	8
Slika 3. Prikaz opcije za objavljivanje na GitHub	8
Slika 4. Postavljena aplikacija na Githubu	9
Slika 5. Use case dijagram za administratora.....	10
Slika 6. Use case dijagram za korisnika	10
Slika 7. Dijagram aktivnosti za prijavu administratora.....	11
Slika 8. Dijagram aktivnosti za odabir prijave za administratora.....	12
Slika 9. Dijagram aktivnosti za evidenciju stanja skladišta za administratora	13
Slika 10. Dijagram aktivnosti za registraciju korisnika za administratora.....	14
Slika 11. Dijagram aktivnosti za upravljanje obavijestima za administratora	16
Slika 12. Dijagram aktivnosti za pregled obavijesti po datumu za administratora	17
Slika 13. Dijagram aktivnosti za prijavu korisnika.....	18
Slika 14. Dijagram aktivnosti za odabir laboratorija za korisnika.....	19
Slika 15. Dijagram aktivnosti za pregled obavijesti za korisnika.....	20
Slika 16. Dijagram aktivnosti za pregled obavijesti po datumu za korisnika	21
Slika 17. Dijagram aktivnosti za uređivanje stanja laboratorija za korisnika	22
Slika 18. Dijagram klasa	24
Slika 19. Sekvencijalni dijagram za korisnika.....	25
Slika 20. Sekvencijalni dijagram za administratora.....	26
Slika 21. Mockup prozora prijave korisnika	27
Slika 22. Mockup prozora za odabir prijave ili laboratorija.....	27
Slika 23. Mockup prozora za pregled i unos obavijesti	28
Slika 24. Mockup prozora za registraciju korisnika	28
Slika 25. Mockup prozora za skladište i unos nove kemikalije.....	29
Slika 26. Mockup prozora za evidenciju stanja laboratorija	29
Slika 27. Mockup prozora za pristup skladištu iz laboratorija	30
Slika 28. Prikaz koda za inicijalizaciju baze podataka	31
Slika 29. Prikaz koda za otvaranje konekcije prema bazi podataka	31
Slika 30. Prikaz koda za zatvaranje konekcije prema bazi podataka.....	32
Slika 31. Prikaz koda za funkciju Insert()	32
Slika 32. Prikaz koda za funkciju Update().....	32
Slika 33. Prikaz koda za funkciju Delete().....	33
Slika 34. Prikaz koda za funkciju Select()	33
Slika 35. DohvatiIme() - Dohvaćanje imena i prezimena korisnika.....	33
Slika 36. Metode za unos novih i upravljanje postojećim obavijestima	34
Slika 37. kalendar_SelectionChanged() - Prikaz obavijesti po datumu	34
Slika 38. Btn() - Prijava korisnika	35
Slika 39. btnUlazak_Click() - Odabir prijave ili laboratorija.....	35
Slika 40. podaci_GetData() - Dohvaćanje podataka skladišta iz baze podataka.....	36
Slika 41. podaci_SelectedIndexChanged() - Funkcija za dohvaćanje indeksa reda kemikalija.....	36
Slika 42. Metode za unos novih i upravljanje postojećih kemikalija	37
Slika 43. btnPrijava_Click() - Registracija korisnika	37
Slika 44. Metode za dohvaćanje imena korisnika i naziva kemikalije	38
Slika 45. btnUzmi_Click() - Uzimanje kemikalije iz laboratorija.....	38
Slika 46. btnDodaj_Click() - Dodavanje kemikalije u laboratorij.....	39
Slika 47. btnPrenesi_Click() - Prijenos kemikalije iz skladišta u laboratorij.....	39
Slika 48. Naslovna stranica ZAP alata	40
Slika 49. Dodavanje poveznice web aplikacije za automatsko skeniranje	41
Slika 50. Prikaz rezultata testiranja.....	41

Slika 51. Test metode btnUzmi_ClickTest().....	46
Slika 52. Neuspješno izvedeno testiranje	46

Popis tablica

Tablica 1. Validacija funkcionalnosti 44