

## ONLINE APLIKACIJA ZA EVIDENTIRANJE INCIDENATA UPOSLENIH

## ONLINE APPLICATION FOR DOCUMENTING EMPLOYEE INCIDENTS

**Nevezudin Buzadija**  
Politehnički fakultet Univerziteta u  
Zenici  
Travnik

**Naida Bikić**  
Filozofski fakultet Univerziteta u Zenici  
Zenica

### REZIME

*Aplikacija za evidentiranje incidenata uposlenih jedne organizacije je jako važan segment organizacione strukture u dinamičkom okruženju. Aplikacija je online koja funkcioniše sa tri nivoa hijerarhijske strukture, a ti nivoi obezbjeđuju privatnost podataka i dostupna je svim uposlenim jedne organizacije. Efikasnost i jednostavnost upotrebe je posebno izražena u obrazovnim ustanovama i organizacijama koje upošljavaju veliki broj ljudi. Za izradu aplikacije korištena je MVC C# tehnologija uz povezivanje sa Microsoft SQL bazom podataka. U ovom radu demonstrirana je i brzina izvršavanja zahtjeva menadžera u web okruženju.*

**Ključne riječi:** online aplikacija, MVC C#, SQL, bootstrap metodologija, incidenti

### ABSTRACT

*An application for documenting employee incidents in an organization is a very important segment of the organizational structure in a dynamic environment. The solution is designed as an online application that functions with three levels of hierarchical structure, providing data privacy and accessible to all employees of an organization. Efficiency and ease of use is particularly pronounced in educational institutions and organizations employing a large number of people. The MVC C# technology was used to create the application, relying on the Microsoft SQL database. This paper also demonstrates the speed of execution of manager requests in web environment.*

**Keywords:** online application, MVC C#, SQL, bootstrap methodology, incidents

### 1. UVOD

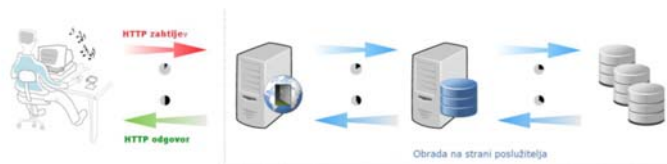
Online aplikacije sve su više zastupljene u praćenju poslovnih procesa. Svrha aplikacije za evidentiranje incidenata uposlenih u organizaciji je da se na vrijeme poduzmu odgovarajuće mjere za otklanjanje problema. Dinamičke aplikacije danas su sve više zastupljene, a jedan od programskih jezika za stvaranje takvih aplikacija je objektni C#. Za razvoj ove aplikacije korišten je model vodopada iz razloga što daje najbolje mogućnosti u pogledu stvaranja integriranog troslojnog pristupa. Kreiranje aplikacije podrazumijeva stvaranje novog programskog rješenja u razvojnom okruženju Microsoft Visual Studio kao ASP.NET MVC web aplikacije.

C# kodom stvaraju se klase, odnosno modeli podataka na osnovu kojih je uz pomoć Object Relational Mapper-a kreirana baza podataka. Suprotan prethodnom postupku je Database-first pristup kod koga se prvo kreira baza podataka (npr. u SQL Serveru), te se na osnovi te baze podataka uz pomoć Object Relational Mapper-a stvaraju potrebni modeli podataka i klasa [6]. Dakle, nakon stvorenih klasa (modela) uz pomoć Entity Frameworka, automatski je kreirana i potrebna lokalna baza podataka. Pri samom povezivanju aplikacije i baze, podaci o njihovoj vezi se spremaju u posebnu datoteku projekta za naknadna pozivanja.

Uzorak Model - View - Controller (MVC) [2], dijeli aplikaciju na tri glavne komponente: model, pregled i kontroler. ASP.NET MVC radni okvir pruža alternative s obzirom na ASP.NET Web Forms za izradu web aplikacija. ASP.NET MVC okvir je provjeren i integriran s postojećim ASP.NET verzijama. Okvir MVC definiran je u sklopu System.Web.Mvc.

Arhitektura aplikacije obuhvata objekte i podatke domena. Zahtjeve za promjene prima od kontrolera i obavještava pogled o promjenama, a zahtjeve za čitanje prima od pogleda. Pogled je implementacija korisničkog interfejsa koji obuhvata sve elemente interfejsa i može neposredno čitati podatke modela. Kontroler je logička komponenta korisničkog interfejsa koji prima obavještenja o aktivnostima korisnika od pogleda i inicira aktivnosti na nivou modela, kao i promjene na nivou pogleda [3].

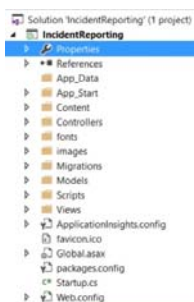
Princip po kome funkcioniraju web aplikacije prilično je jednostavan. Korisnikov web browser šalje zahtjev web serveru, koji podatke prosljeđuje web aplikaciji, koja ih obradi, rezultat vraća serveru, pa napokon i browseru. U toj situaciji browser ne zna da li je korisnik uslužen statičkom stranicom ili je kod koji browser prikazuje generiran od strane Web aplikacije (slika 1). Bitna kvalitativna svojstva web aplikacija i procesa razvoja mogu se povezati sa opštim standardom evaluacije kvaliteta softvera ISO/IEC 9126-1.



Slika 1. Dinamičke web aplikacije i statičke web stranice [4]

## 2. TEST PROJEKAT

Izradu novog projekta treba da prati i kreiranje paralelnog projekta radi lakšeg testiranja. Jedan test projekat može se koristiti za više srodnih projekata. Paralelno kreiranje test projekta je izvršeno zato što početni generisani kod na ovaj način određuje način korištenja ove test biblioteke. Na slici 2. prikazan je Solution Explorer sa svim komponentama od kojih se neke mogu iskoristiti u drugim projektima.



Slika 2. Mapa test projekta - Incidenti

### 3. SPECIFIKACIJA SOFTVERSKIH ZAHTJEVA

Specifikacija softverskih zahtjeva jasno definira uloge i zadatke za svakog učesnika u procesu, odnosno za svaku kategoriju korisnika aplikacije, kako slijedi:

- Naručilac projekta: pomoću ovog dokumenta naručilac će moći da prati razvoj traženog proizvoda. Prilikom isporuke svi zahtjevi navedeni u specifikaciji moraju biti ispunjeni.
- Menadžer projekta: koristi specifikaciju zahtjeva kako bi se uvjerio da su obuhvaćeni svi zahtjevi od strane korisnika i da razvojnom timu omogući opis sistema koji treba razviti, kao i da prati sam razvoj sistema.
- Programer: koristi dokument za razvoj sistema koji će u potpunosti ispuniti očekivanja naručioca projekta i krajnjih korisnika. Svi slučajevi korištenja navedeni u dokumentu moraju biti implementirani i funkcionalni prilikom isporuke konačne verzije proizvoda.
- Web dizajner: na osnovu specifikacije zahtjeva i ograničenja u pogledu dizajna treba da definira izgled korisničkog interfejsa aplikacije.
- Tester: koristi specifikaciju za verifikaciju i validaciju traženih funkcija aplikacije.

### 4. KORISNIČKI INTERFEJS

Interfejs aplikacije je implementiran po ugledu na bootstrap metodologiju. Aplikacija pristupa bazi podataka izvršavanjem različitih upita. Ne postoje zahtjevi i ograničenja u pogledu hardverskog interfejsa. Softver za komunikaciju koristi HTTP protokol. Na slici 3. prikazana je forma za unos incidenata.



Slika 3. Online forma za unos incidenata

### 5. FUNKCIONALNI ZAHTJEVI

Za potrebe interakcije sa korisnikom dodate su funkcije za registrovanje i prijavu korisnika na sistem. Registrovanje korisnika u sistemu dozvoljeno je samo uposlencima firme. Izgled korisničkog interfejsa i dostupne funkcije su različite u zavisnosti od kategorije korisnika koji pristupa sistemu, kao što se vidi na slikama 4 i 5.

```
public ActionResult Create([Bind(Include = "IncidentTypeId,Name")] IncidentType
incidentType)
{
    if (ModelState.IsValid)
    {
        db.IncidentTypes.Add(incidentType);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(incidentType); }
public ActionResult Edit([Bind(Include = "IncidentTypeId,Name")] IncidentType incidentType)
{
    if (ModelState.IsValid)
    {
        db.Entry(incidentType).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(incidentType); }
```

Slika 4. Funkcije forme za kreiranje i za izmjenu tipova incidenata



Slika 5. Online forma za kreiranje tipova incidenata

Kategorije korisnika su administrativni radnik, voditelj poslova i uposleni, a svaka kategorija ima definiranu listu specifikacija, kako slijedi:

Administrativni radnik:

- Logovanje: Sistem omogućava autentifikaciju korisnika koji mu pristupa.
- Evidencija radnika: Sistem omogućava administrativnom radniku da vrši evidenciju svih polaznika.
- Evidencija voditelja: Sistem omogućava administrativnom radniku da vrši evidenciju svih voditelja.
- Evidencija grupa: Sistem omogućava administrativnom radniku da vrši unos i izmjenu korisničkih grupa.
- Evidencija odjela: Sistem omogućava administrativnom radniku da vrši unos i izmjenu odjela.
- Evidencija sektora: Sistem omogućava administrativnom radniku da vrši unos i izmjenu sektora.
- Evidencija incidenata: Sistem omogućava administrativnom radniku da vrši unos i izmjenu incidenata.

Voditelj poslova

- Logovanje: Sistem omogućava autentifikaciju korisnika kao voditelj poslova
- Evidencija prijava incidenata: Sistem omogućava voditelju poslova da vrši pregled svih prijava.
- Unos novih prijava: Sistem omogućava voditelju poslova da vrši evidenciju svih prijava.
- Izmjena i pretraga prijava: Sistem omogućava voditelju poslova da vrši pretragu i izmjenu prijava.
- Izmjena šifre: Sistem omogućava voditelju poslova da vrši promjenu šifre.

Uposleni

- Logovanje: Sistem omogućava autentifikaciju korisnika kao uposlenik.
- Evidencija svojih incidenata: Sistem omogućava uposlenom da vrši pregled svih svojih prijava.
- Unos prijava: Sistem omogućava uposlenom unos novih prijava u sistem.
- Izmjena i pretraga svojih prijava: Sistem omogućava uposlenom da vrši pretragu i izmjenu svojih prijava.
- Izmjena šifre: Sistem omogućava uposlenom da vrši promjenu šifre.

Na slikama 6 i 7 prikazane su forme i funkcije za pristup različitim nivoa u evidentiranju incidenata po pitanju definisanih dozvola.



Slika 6. Nivoi pristupa online aplikaciji

```

public ActionResult Edit(string id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    ApplicationUser applicationUser = db.Users.Find(id);
    if (applicationUser == null)
    {
        return HttpNotFound();
    }
    ViewBag.DepartmentId = new SelectList(db.Departments, "DepartmentId",
        "Name", applicationUser.DepartmentId);
    ViewBag.SectionId = new SelectList(db.Sections, "SectionId", "Name",
        applicationUser.SectionId);
    return View(applicationUser);
}

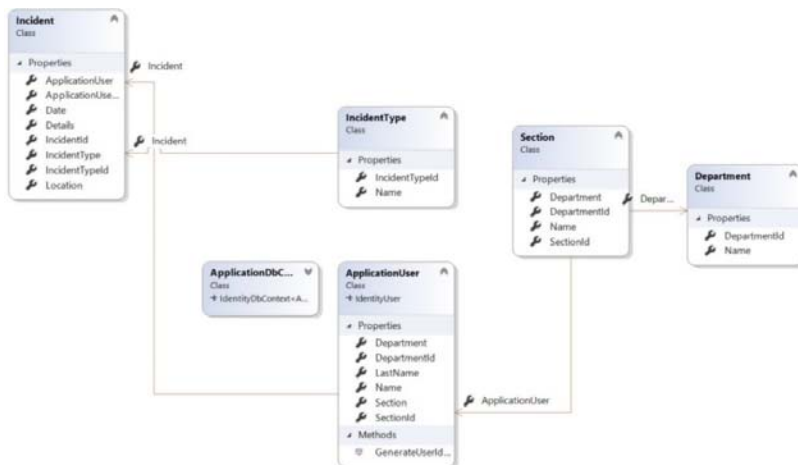
```

Slika 7. Funkcija forme za pristup aplikaciji

## 6. NEFUNKCIONALNI ZAHITJEVI

Web aplikacija postavljena je na server koji omogućava brzu reakciju na korisničke akcije. Podacima u sistemu ne mogu pristupati svi korisnici. IT zaposleni u firmi imaju veća ovlaštenja od ostalih korisnika, a jedino je administrativnim radnicima dozvoljeno ažuriranje podataka bitnih za poslovanje firme. Do iznenadnog prestanka rada sistema može doći jedino kao posljedica grešaka u radu operativnog sistema korisnika ili problema vezanih za funkcionisanje servera.

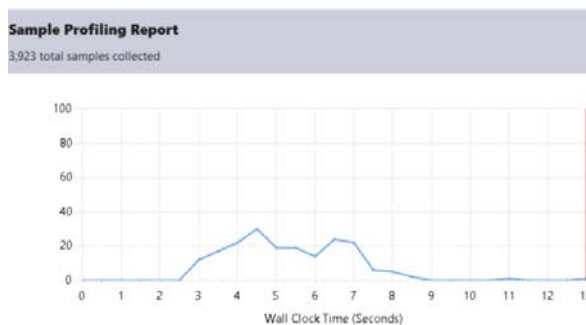
Na slici 8. prikazan je dijagram klasa koji nam govori o fleksibilnosti aplikacije i mogućnosti naknadne nadogradnje u svrhu proširenja funkcionalnosti.



Slika 8. Klasni dijagram aplikacije Incidenti

Interfejs online aplikacije je jednostavan i ne traži visok nivo informatičke pismenosti, što omogućava primjenu u bilo kojoj organizacionoj strukturi. Efikasnost aplikacije za evidentiranje incidenata uposlenih ogleda se u brzom kreiranju izvještaja i povratne veze prema nivoima koji trebaju da reaguju. Efikasnost aplikacije ogleda se i u brzini procesiranja zahtjeva.

Jedna od osnovnih mjera kvaliteta Web aplikacija su performanse. Glavna performansa Web aplikacije je vrijeme odziva, koje prvenstveno ima odraza na kvalitet i upotrebnu vrijednost aplikacije. Vrijeme odziva predstavlja period koji prođe od upućenog zahtjeva korisnika do povratne informacije koju korisnik dobija. Na slici 9. prikazan je izvještaj koji pokazuje brzinu procesiranja aplikacije.



Slika 9. Vrijeme odziva online aplikacije Incidenti

## 7. ZAKLJUČAK

Razvoj softvera je izuzetno zahtjevan posao imajući u vidu da se greška u razvoju, napravljena u bilo kojem trenutku, negativno odražava na određene aspekte razvoja tog istog softvera u budućnosti. Aplikacija je kreirana u dinamičkom okruženju i dostupna je u svakom trenutku putem web browsera. Moguća je i integracija ove aplikacije u postojeći informacijski sistem jedne firme i moguća je njena upotreba u analizi tipova incidenata u firmi. Posebno je značajno da je ovako kreirana online aplikacija podržana na svim platformama i uređajima.

Novija istraživanja [9-11] ukazuju na brojne prednosti ali i na nedostatke korištenja elektronskog praćenja incidenata zaposlenih, a ova studija slučaja može se koristiti za testiranje primjenjivosti u konkretnoj organizaciji. Dalji razvoj ove aplikacije treba da obuhvati povećanje sigurnosti uvijerenjem HTTPS sigurnog protokola za pristup aplikaciji, cloud-baziranih rješenja kao što je Microsoft Azure, te da koristi iskustva prikupljena prilikom korištenja ove aplikacije u konkretnom radnom okruženju u firmama različitih veličina.

## 8. LITERATURA

- [1] Allen K. S., D. Matson, J. Galloway, Wilson B.: Professional ASP.NET MVC 5, Wrox, 2014
- [2] ASP.NET MVC Overview: Microsoft, [https://msdn.microsoft.com/en-us/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx) (24.4.2019)
- [3] Esposito D.: Programming Microsoft ASP.NET MVC (3rd Edition), SAD, Microsoft Press, 2014
- [4] Esposito D., Saltarello A.: Microsoft .NET: Architecting Applications for the Enterprise, Second Edition, Microsoft Press Redmond, 2015
- [5] Freeman, A.: Pro ASP.NET MVC 5. Apress, 2013
- [6] Galloway J., Allen K. S., Matson D., Wilson B.: Professional ASP.NET MVC 4, SAD, John Wiley & Sons Inc, 2012
- [7] Gitlab, SSgs part1 Static vs. Dynamic Websites, <https://about.gitlab.com/2016/06/03/ssg-overview-gitlab-pages-part-1-dynamic-x-static/> (24.4.2019)
- [8] Manger R.: *Softversko inženjerstvo*, Prirodoslovno matematički fakultet u Zagrebu, 2015
- [9] Grispos G., Glisson W.B., Bourrie D., Storer T., Miller S.: Security incident recognition and reporting (SIRR): an industrial perspective. arXiv preprint arXiv:1706.06818., 2017
- [10] Grispos G., Glisson W.B., Bourrie D., Storer T., Miller S.: Security incident recognition and reporting (SIRR): an industrial perspective. arXiv preprint arXiv:1706.06818., 2017
- [11] Burk J.A., Prince G.: Mobile Electronic Devices In The Workplace: A Strategy to Manage the Risk. Professional Safety, 64(4), 40-41., 2019