

## 1. Uvod

Programski sustav za praćenje SARS-CoV-2 virusa, bolesnika, kao i kompletne situacije vezane oko odluka kriznih stožera, sastojat će se od namjenske aplikacije načinjene po načelu objektivno orijentiranog modeliranja. Namjenski program biti će pisan u programskom jeziku C# uz pomoć korištenja Visual Studio okruženja. Upravljanje bazom podataka bit će korišten SQL Server, dok samo sučelje, biti će prikazano pomoću ReactJS programskog okruženja.

### 1.1. Doseg projekta

Projekt za cilj ima stvoriti cjelokupnu sliku trenutne pandemije koja vlada, koji su kapaciteti bolesnika po bolnicama, koliko pozitivnih na virus trenutno ima kao i samo stanovništvo, njihovo ponašanje i reakcija kritičnog stožera na situaciju. Vidljivo rješenje aplikacije bi trebali biti svi ti podaci, sistematski svrstani, na jednom mjestu i vizualno pregledni za korisnika.

### 1.2. Glavne funkcije programske opreme

Funkcije programske opreme su: **UnesiOsobu** – pohranjujemo osobe koji na neki način su dio pandemije COVID -19, bili to pacijenti ili stručno osoblje križnih stožera, **UnesiSimptome** – ukoliko unijeta osoba ima pozitivan nalaz na virus, moramo detaljno zabilježiti trenutno stanje, **UnesiPregled** – ukoliko unijeta osoba ima dogovorene buduće preglede zbog situacije vezane sa COVID -19 pandemijom, korisnik je obavezan to zabilježiti, **TragStanja** – opisuje trenutno stanje sa vladajućom pandemijom, koje su crvene zone kretanja, kakva je statistika u tim područjima, koje se mjere preporučaju, **DodajStožer** – mogućnost dodavanja križnih stožera vezanih za situaciju, kao i popis njihovih članova, **DodajOdluke** – dodajemo trenutna upozorenja i naputke od samih križnih stožera kojih se korisnik treba pridržavati, **DodajInstitucije** – popis svih nadležnih institucija, njihovi kontakti kao i radna vremena kako bi korisnik mogao doći do dodatnih informacija.

### 1.3. Zahtjevi za performansama i ponašanje programske opreme

Programska oprema mora imati intuitivno korisničko sučelje sa što manje izbora jer se pretpostavlja da će se njome koristiti korisnici s malim informatičkim znanjem kao i oni koji će biti u velikoj žurbi zbog novih priljeva pacijenata koje će morati popisivati. Zbog razvijenosti ove situacije, sami korisnici neće imati vremena za poduku o radu na sustavu.

### 1.4. Upravljanje i tehnička ograničenja

Ograničenje ovog projekta jest da rješenje mora biti gotovo što je prije moguće zbog razvoja trenutne situacije i potrebe klijenta da nadgleda stanje.

## 2. Procjena projekta

Na projektu se planira rad 5 ljudi. Procjenjuje se da će izrada projekta trajati 25 dana.

### 2.1. Povijesni podaci korišteni za procjenu

Procijenjeno vrijeme potrebno za izradu projekta temeljeno je na vremenu koje je bilo potrebno za izradu prijašnjih, sličnih projekata uz pretpostavku da ne treba ponovno trošiti vrijeme na ovladavanje tehnikama koje su svladane na prijašnjim projektima.

## 2.2. Procjena primijenjenih tehnika i rezultata

Aktivnost	Trajanje /č x h	Čovjek /zadatku
Doseg projekta	42	2
Analiza (prikupljanje informacija, prijedlog izvedbe sustava)	70	1
Dizajn	44	3
Izrada (kodiranje funkcija sustava, izrada korisničkog sučelja)	264	4
Izrada baze podataka	2	1
Testiranje	102	1
Pisanje dokumentacija sustava	23	1
Uvođenje sustava	23	1
	590	

### 2.2.1. Procjena kodiranja funkcija, izrade korisničkog sučelja i izvještaja

Aktivnost	Trajanje /č x h
Izrada formi korisničkog sučelja, provjera unesenih podataka	17
Funkcije za pohranu podataka	50
Funkcije za ispis izvještaja	65
Izrada izvještaja	41
Funkcije izračuna	50
Razvojno testiranje (debugiranje)	41
	264

## 2.3. Usklađivanje procjena

Na projektu se planira rad 5 ljudi. Procjenjuje se da će izrada projekta trajati 25 dana.

## 2.4. Resursi projekta

U izvršenju projekta sudjelovat će voditeljica Petra Kos koja je ujedno i sistem-analitičar, 3 programera Lovre Gusar, Marko Maglić i Magda Milički koji kodiraju, testiraju i pišu dokumentaciju, administrator baze podataka i sistem-administrator Goran Horvat koji uvodi program. Potrebno je 5 računala za članove ekipe, na kojima treba biti instaliran Visual Studio, SQL Server, te MS Office.

## 3. Upravljanje rizicima

Glavni rizici su da projekt neće biti gotov do roka, da neće biti u skladu s očekivanjima te da bi se zahtjevi mogli promijeniti.

### 3.1. Rizici projekta

S obzirom da se programska oprema mora početi koristiti od početka kalendarske godine mora do tada biti završena.

Moglo bi se dogoditi da projekt ne bude u skladu s očekivanjima zato što je ovo prvi programski sustav za poslovnu primjenu na kojem ova ekipa radi.  
Loše komentiran kod može dovesti do otežanog održavanja programa.

### 3.2. Tablica rizika

Ime rizika	Vjerojatnost	Utjecaj
Kašnjenje s isporukom	30%	1
Proizvod neće biti u skladu s očekivanjima	30%	2
Promjene u zahtjevima	20%	2
Loše komentiran kod	10%	4

Vrijednosti utjecaja

- 1 – katastrofalno
- 2 – kritično
- 3 – marginalno
- 4 – zanemarivo

### 3.3. Pregled ublažavanja, praćenja i upravljanja rizicima

#### Kašnjenje s isporukom

- Ublažavanje rizika  
Ovaj rizik pokušava se ublažiti početkom izrade projekta na vrijeme.
- Praćenje rizika  
Radi se raspored za nadgledanje statusa projekta koji se mora slijediti tijekom svih faza izade.
- Upravljanje rizikom  
Ukoliko se primjeti da se projekt neće uspjeti dovršiti na vrijeme, on se prekida, jer od projekta koji zakasni neće biti koristi.

#### Proizvod neće biti u skladu s očekivanjima

- Ublažavanje rizika  
Ovaj rizik se pokušava ublažiti pažljivom analizom korisnikovih potreba.
- Praćenje rizika  
Tijekom cijelog razvoja komunicira se s korisnikom.
- Upravljanje rizikom  
Ukoliko se uoči nezadovoljstvo korisnika, provest će se racionalna (stvarno moguća) prilagodba prema korisničkim zahtjevima.

#### Promjene u zahtjevima

- Ublažavanje rizika  
Redovno će se odvijati sastanci sa korisnicima da bi se osiguralo da su projekt i zahtjevi korisnika usklađeni.
- Praćenje rizika  
Sastanci s korisnikom trebaju osigurati razumijevanje između izvoditelja i korisnika.
- Upravljanje rizikom  
Čim se uoče eventualne razlike između očekivanja i razvijenog proizvoda, korisnik će biti obaviješten i bit će raspravljene mogućnosti daljnje akcije.

#### Loše komentiran kod

- Ublažavanje rizika  
Loše komentiran kod može se izbjeći postavljanjem standarda za izradu programa.
- Praćenje rizika  
Tijekom cijelog razvoja provjeravati će se da li je kod komentiran sukladno postavljenom standardu..
- Upravljanje rizikom  
Ukoliko se uoči da se ne slijede postavljeni standardi, nedostaci će se popraviti.

## 4. Raspored projekta

### 4.1. Skup projektnih zadataka

Faze procesa su određivanje teme projekta i glavne informacije o projektu te upoznavanje tima, analiza zahtjeva za aplikaciju, Stvaranje konceptualnog i fizičkog modela, specifikacija dizajna i zahtjeva (funkcionalnih i nefunkcionalnih), Izrada beta verzije web aplikacije, izrada konačne verzije web aplikacije i uvođenje sustava te posljednja faza, pisanje programske i korisničke dokumentacije.

### 4.2. Funkcionalna dekompozicija

Početna faza projekta sastoji se od upoznavanja tima, te uvođenje u samu temu projekta. Nakon toga, u drugoj fazi, treba zapisati korisničke zahtjeve te napraviti plan projekta. Slijedi stvaranje konceptualnog i fizičkog modela, a nakon tog specifikacija dizajna i specifikacija određenih zahtjeva (funkcionalnih i nefunkcionalnih). Izrada beta verzije aplikacije se sastoji od izrade baze i povezivanja baze, pisanja programskog koda u C#, povezivanja *frontend*-a i *backend*-a i dizanja aplikacije na prije osposobljeni server. Na kraju projekta se piše dokumentacija (programska i korisnička).

## 4.3. Mrežni plan

### Gantogram



#### 4.4. Tablica vremenskih intervala projekta

Projektni zadaci:

Redni broj	Zadatak	Početak	Kraj
1.	Određivanje teme projekta i glavne informacije o projektu te upoznavanje tima	13.03.2020	20.03.2020
2.	Analiza zahtjeva za aplikaciju	20.03.2020	22.03.2020
3.	Stvaranje konceptualnog i fizičkog modela	22.03.2020	6.04.2020
4.	Specifikacija dizajna i zahtjeva (funkcionalnih i nefunkcionalnih)	6.04.2020	10.04.2020
6.	Izrada beta verzije web aplikacije o COVID-19 pandemiji	10.4.2020	15.05.2020
7.	Izrada konačne verzije web aplikacije i uvođenje sustava	15.05.2020	09.06.2020
8.	Pisanje programske i korisničke dokumentacije	09.06.2020	15.06.2020

### 5. Organizacija sudionika projekta

Na projektu sudjeluje voditelj tima, tri programera, administrator baze podataka / sistem-administrator.

#### 5.1. Struktura tima

Voditelj tima zadužen je za definiranje projekta i analizu zahtjeva te upravljanje projektom. Programeri su zaduženi za izradu korisničkog sučelja, izvještaja, kodiranje funkcija, testiranje, poduku korisnika i pisanje dokumentacije. Administrator baze podataka zadužen je za izradu baze podataka. Sistem-administrator zadužen je za uvođenje u rad.

#### 5.2. Izvještavanje i komunikacija

Na projektu radi samo jedan tim, pa komunikacija između timova nije potrebna. Komunikacija unutar tima je verbalna i pismena putem elektroničke pošte.

### 6. Mehanizmi praćenja i kontrole

S obzirom da se radi o kratkotrajnom projektu, napredak će se pratiti na kraju svakog radnog dana. Ukoliko dođe do neusklađenosti s predviđenim, vremenski plan će se revidirati.

#### 6.1. Osiguranje kvalitete

##### SQA Plan

- **Namjena**

Ovim planom pokrivena je izrada Programskog sustava za praćenje SARS-CoV-2 virusa. Plan sadrži faze analize, oblikovanja, izrade programskog sustava, testiranja, pisanje dokumentacije te uvođenja u primjenu.

- **Zaduženja**

Voditelj projekta zadužen je za analizu i oblikovanje. Programeri i administrator baze podataka zaduženi su za izradu programskog sustava, testiranje i pisanje dokumentacije te poduku korisnika.

Sistem-administrator zadužen je za uvođenje u primjenu.

- **Dokumentacija**

Ovaj projekt izrađuje se na temelju Specifikacije zahtjeva.

Korisnička dokumentacija sadržavati će opis rada programa i upute za korištenje.

Tehnička dokumentacija sadržavat će tehničku specifikaciju sustava i upute za održavanje.

- **Alati, tehnike i metodologije**

Programska oprema bit će napisana objektno-orientiranim pristupom u jeziku C#. Koristit će se Visual Studio. Za upravljanje bazom podataka koristit će se SQL Server. Sučelje će biti prikazano pomoću ReactJS programskog okruženja.

- **Kontrola programskog koda**

Za pohranu i kontrolu pristupa kodu koristit će se Visual Studio Team Foundation Source Control.

## **6.2. Upravljanje promjenama**

Upravljanje izmjenama provoditi će voditelj projekta.