

Projekt R

Ak. god. 2023./2024.

Pičibangoš sastančenje

Dokumentacija projekta

Sudionici: *Filip Kovač, Filip Barić*

Datum predaje: 26.1.2024.

Nastavnik: *izv. prof. dr. sc. Vladimir Čeperić*

Contents

1	Opis projektnog zadatka	2
1.1	Opis aplikacije	4
2	Specifikacija programske potpore	6
2.1	FastAPI	6
2.2	Streamlit	6
2.3	BeeWare	7
2.4	Programski zahtjevi	7
3	Zaključak i budući rad	9
	Popis literature	10

1. Opis projektnog zadatka

Uvod

Osnova projekta je bilo razviti programsku podršku za *Pičibangoš*, web aplikaciju koja predstavlja online planer i nudi korisnicima opcije ugovaranja sastanaka, stvaranja evenata i timova, i online komunikaciju sa prijateljima pojedinačno ili unutar tima. Također, još jedan od zadataka je bio istražiti mogućnosti koje nudi programski jezik Python sa framework-ovima Streamlit i BeeWare u razvijanju frontend-a web aplikacije i u povezivanju sa backendom, isto tako građenim pomoću pythonovog framework-a FastAPI.

Ciljevi

- Istražiti mogućnosti razvoja programske potpore za web pomoću programskog jezika Python
- Istražiti Pythonove framework-e za razvoj web aplikacije
- Odabir Pythonovih framework-a
- Analiza primjene
- Zaključak na temelju korištenih tehnologija

Python

Python je jedan od najpopularnijih i najučinkovitijih programskih jezika koji sadrži goleme library-e i framework-e za gotovo svaku tehničku domenu. Pythonovi framework-ovi automatiziraju implementaciju mnogih zadataka i daju programerima dobru strukturu za razvoj aplikacija. Svaki framework dolazi s vlastitom kolekcijom modula ili paketa koji značajno skraćuju vrijeme razvoja.

Model razvoja

Životni ciklus razvoja aplikacija daje proces kojim se aplikacije razvijaju, a modeli razvoja aplikacija daju pristup provedbi tog procesa. Ti modeli opisuju kako se sve faze životnog ciklusa procesa razvoja softvera spajaju u softverski projekt. Postoji nekoliko uobičajenih popularnih modela razvoja i stotine, ako ne i tisuće, njihovih iteracija. Model razvoja softvera opisuje što će se raditi ali ostavlja puno prostora načinu kako će se raditi. Potrebno je odabrati odgovarajući model ovisno o veličini projekta. Za male projekte je vodopadni model bez dodatne dokumentacije i bez restrikcija iterativnog pristupa dobar izbor, te smo u ovom projektu odlučili taj i koristiti.

1.1 Opis aplikacije

Uvod

Aplikacija Pičibangoš je osmišljena kao planer gdje korisnik ima opciju ugovarati sastanke i događaje, bilo privatne ili poslovne svrhe, te na taj način voditi evidenciju obaveza i događaja. Također aplikacija nudi opciju i povezivanja, te komuniciranja se sa kolegama, pri čemu korisnik na taj način može vidjeti obaveze kolega s kojima je povezan ili kao prijatelj ili preko tima, te time može lakše upravljati i dogovarati događaje koji se tiču više ljudi. Stoga, cilj aplikacije ja olakšati svakodnevnu komunikaciju i suradnju sa kolegama i prijateljima, te time uštediti vrijeme koje bi se neizbježno izgubilo pri dogovaranju sastanaka kroz raspitivanje o obavezama pojedinaca.

Korištenje aplikacije

Nakon svake uspješne prijave korisnik dobiva pregled svih svojih obaveza koje ima unutar bilo kojeg tima, kao i obaveza koje ima sa pojedinačnim korisnicima na osobnom kalendaru. Iznad kalendara su mu također ponuđene mogućnosti stvaranja novih događaja, timova ili pronalaženja novih prijatelja.

Pri odabiru jednog od timova u koje korisnik uključen, kalendar se prilagođava te preciznije prikazuje obaveze koje korisnik ima unutar tima, kao i obaveze koje imaju njegovi timljani. Na taj način korisnik može preciznije odabrati vrijeme u kojem bi mogao kreirati novi događaj, a kojem bi mogli prisustvovati svi timljani. Također korisnik može predložiti jednog od svojih prijatelja ko novog timljana, što odobrava admin koji je stvorio tim, te na temelju toga se i kalendar ažurira.

Korisnik pri stvaranju novog događaja daje ime događaja, odabire vrijeme u kojem će se spomenuti događaj odvijati i sudionike koje poziva na događaj, bilo pojedinačno navedene ili kroz navođenje timova u kojima su oni prisutni, a korisnik je isto tako sudionik.

Pri stvaranju novih timova, korisnik navodi ime tima, njegov opis te odabire prijatelje sa kojima želi stvoriti navedeni tim.

Ako korisnik želi dodati novog prijatelja, to radi kroz upisivanje imena korisnika u aplikaciji, te kroz slanje zahtjeva prijateljstva, kojeg druga strana mora potvrditi. Na taj način korisnici postaju prijatelji, te mogu stvarati i gledati zajedničke obaveze, kao i imati uvid u obaveze prijatelja.

2. Specifikacija programske potpore

Python framework je zbirka Python modula koja pruža skup zajedničkih funkcija koje se mogu koristiti kao struktura za izgradnju aplikacija bilo koje vrste. Framework-ovi su osmišljeni kako bi pojednostavili razvojni proces dajući općenite smjernice o tome kako bismo trebali graditi softver i apstrahirali neke od složenijih ili ponavljajućih zadataka, što stoga omogućuje jedinstvenu i prilagođenu logiku za aplikacije.

Stoga, nakon istraživanja, odlučili smo koristiti navedene Python framework-e u izgradnji aplikacije:

1. FastAPI
2. Streamlit
3. BeeWare

2.1 FastAPI

FastAPI je moderan web framework koji je kreirao Sebastián Ramírez za izgradnju RESTful API-ja u Pythonu. Ubrzo je stekao popularnost među programerima zbog svoje jednostavne upotrebe, brzine koja je usporediva sa NodeJS-om i Go-om, i robusnosti. Temelj se zasniva na Pydantic-u, te omogućuje validaciju, serijalizaciju i deserijalizaciju podataka. Također automatski generira OpenAPI dokumentaciju za API-je izgrađene s njim.

Upravo radi navedenih razloga, FastApi je postao najbolji odabir za izgradnju backend-a aplikacije.

2.2 Streamlit

Streamlit je Python framework otvorenog koda koja može lako pretvoriti projekte iz podatkovne znanosti i strojnog učenja u web aplikacije. To je izvrstan odabir

za znanstvenike koji se bave podacima ili inženjere strojnog učenja čiji su temeljni skupovi vještina usmjereni na izradu prototipova, eksperimentiranje s modelima i manipuliranje podacima u Pythonu. Streamlit stoga omogućuje stvaranje aplikacije zapanjujućeg izgleda u nekoliko redaka koda, no unatoč navedenim prednostima tu su naravno poneki nedostaci. Najveći utjecaj je imao manjak fleksibilnosti Streamlit-ovih opcija koje su u nekim slučajevima znale otežati razvoj i najjednostavnijih funkcionalnosti.....FILIPE POSERI SE TU

2.3 BeeWare

Python se pokazao kao vrlo sposoban jezik - pristupačan za početnike, ali moćan u rukama stručnjaka. Projekt BeeWare ima za cilj iskoristiti snagu Pythona kao jezika i koristiti ga kako bi korisnicima svih razina vještina omogućio razvoj aplikacija s izvornim korisničkim sučeljima.

Krajnji cilj projekta BeeWare: biti u mogućnosti učiniti za mobilni i stolni korisnički softver isto što je Django napravio za web softver - dati u ruke korisnicima skup alata i biblioteka koji im omogućuju razvoj bogata izvorna korisnička sučelja i implementirati ih na svoje uređaje. Ovo uključuje:

Alati za omogućavanje rada Pythona na različitim uređajima, Alati za pakiranje Python projekta tako da se može izvoditi na tim uređajima, Knjižnice za pristup izvornim widgetima i mogućnostima uređaja, Alati koji pomažu u razvoju, otklanjanju pogrešaka, analizi i implementaciji ovih projekata.

Cilj je da ovaj set alata bude dovoljno jednostavan za korištenje za potpune pridošlice za korištenje u okruženju sličnom Django Girls; ali dovoljno moćni da se mogu koristiti za pokretanje sljedećeg Instagrama, Pinteresta ili Disqus.

No, ne bavimo se samo softverom. Također nam je cilj biti projekt s društvenom sviješću. Cilj nam je razviti i održati raznoliku i inkluzivnu zajednicu i imamo Kodeks ponašanja koji se rigorozno provodi. Također težimo razvoju zdrave i održive zajednice - one koja je svjesna problema mentalnog zdravlja svojih sudionika i koja osigurava resurse za ljude da se počnu angažirati i nastave surađivati sa zajednicom.

2.4 Programski zahtjevi

Neovisno o framework-ovima, aplikacija mora ispunjavati sljedeće programske zahtjeve:

- Sustav nužno mora podržavati rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti sustav
- Sustav treba imati jasno i intuitivno sučelje
- Sustavu se može pristupiti iz javne mreže
- Zaporke korisnika ne smiju se zapisivati u bazu u plain text formatu

3. Zaključak i budući rad

Zadatak nam je bio napraviti implementaciju kartaške igre u kojoj se igrači bore sa kartama koje mogu skupiti na različitim lokacijama i prijavljuju nove lokacije. Kartografi mogu odobravati lokacije, a administrator može upravljati svim korisnicima i zadavati im privremena isključenja. Na kraju nismo ipak uspjeli implementirati sve što je bilo planirano i neki dijelovi ne rade kako bi trebali prema specifikaciji.

Projekt je započeo odlično sa pisanjem dokumentacije i izradom UML dijagrama. Kad je došlo do implementacije napisanog, nastali su neki problemi. Nismo znali točno podijeliti poslove i trebali smo dosta proučavati kako rade radni okviri kojima smo radili front-end i back-end aplikacije. No uz njih smo naučili puno o tim radnim okvirima i vidjeli kakvi se sve problemi mogu riješiti s njima. Za izradu generičkih funkcionalnosti nije bilo prevelikih problema.

Težak problem koji smo trebali riješiti je borba. Kako spojiti igrače i kako izvesti borbu i prolazak kroz stanja borbe. Trebalo nam je dosta vremena da ju implementiramo te nije implementirana na najelegantniji način. Ostatak funkcionalnosti išao je lakše. Bilo je malo problema sa implementacijom karte, ali uspjeli smo je ostvariti. Sa funkcionalnostima koje uključuju jednostavan dohvat podataka nije bilo problema.

Jedino sa čime smo se dosta mučili bez da smo našli rješenje je spremanje i prikaz slika. Osim što slike nisu implementirane, nismo stigli implementirati ni obrasce uporabe za administratora, i nije implementirano skupljanje lokacija tako da se fizički dođe do njih. Igrači karte mogu skupljati samo iz paketića (maknuli smo ograničenje da to budu samo životinje). Obrasci uporabe koji nisu implementirani: UC3, UC9, UC19, UC25, UC27, UC28, UC29, UC30.

Nismo stigli implementirati dosta velik dio aplikacije. Bilo je problema s koordiniranjem zadataka i poslova. Jako puno vremena je otišlo na upoznavanje s alatima. No sada znamo bolje baratati s tim radnim okvirima te bismo mogli projekt napraviti brže.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. FastAPI dokumentacija, <https://fastapi.tiangolo.com/>
5. Streamlit dokumentacija, <https://docs.streamlit.io/>
6. BeeWare dokumentacija, <https://beeware.org/>