

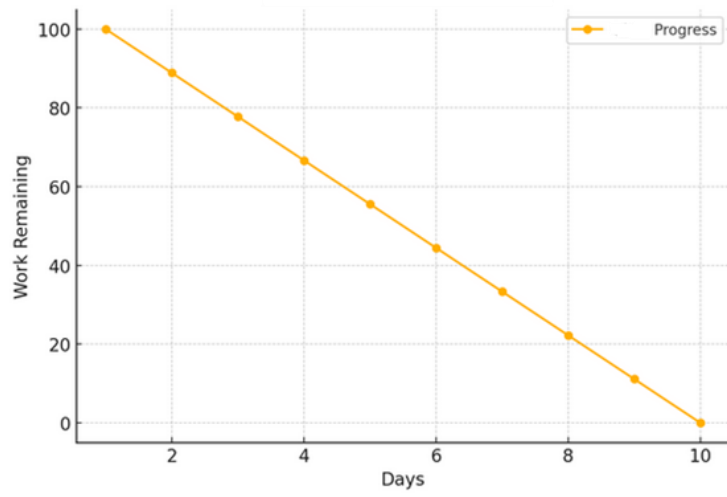
Julski ispitni rok - teorijski deo ispita -

Napomena: Upotreba štampane literature i komunikacionih uređaja je zabranjena i strogo kažnjiva. Dozvoljena je samo upotreba elektronskih materijala sa diska *Materijali*. Ispit traje 165 minuta.

Broj indeksa	Ime i prezime	Broj poena		
		1.	2.	Ukupno

1. [4] Koristeći SCRUM metodologiju, veb sistem muzičkog festivala „Arsenal fest“ u Kragujevcu realizuje tim od petoro programera i testera, koji su čitav projekat podelili u 4 Sprinta, svaki u trajanju od 10 radnih dana. Veb sistem treba da služi posetiocima da se upoznaju sa programom manifestacije, dođu do najvažnijih informacija o izvođačima koji nastupaju i da kupe ulaznice preko elektronskog plaćanja. Grafikoni kako se progres posla ovog softverskog tima izvršavao u svakom Sprintu dat je na sledećim dijagramima. Za dati sistem:
 - a) Opisati u kojim Sprint-ovima postoje periodi bez napretka i sa variranjem količine preostalog posla? Naznačiti u kojim danima se javljaju potencijalni problemi u sva četiri Sprint-a.
 - b) Izračunati brzinu (nagib preostalog posla) za sva četiri Sprinta sa presečnim tačkama u 4. i 7. danu, i u zavisnosti od tih brzina, kog dana bi se Sprint-ovi završili (odnosno da li bi Sprint bio završen ranije, pre roka, ili bi kasnio).

Sprint #1

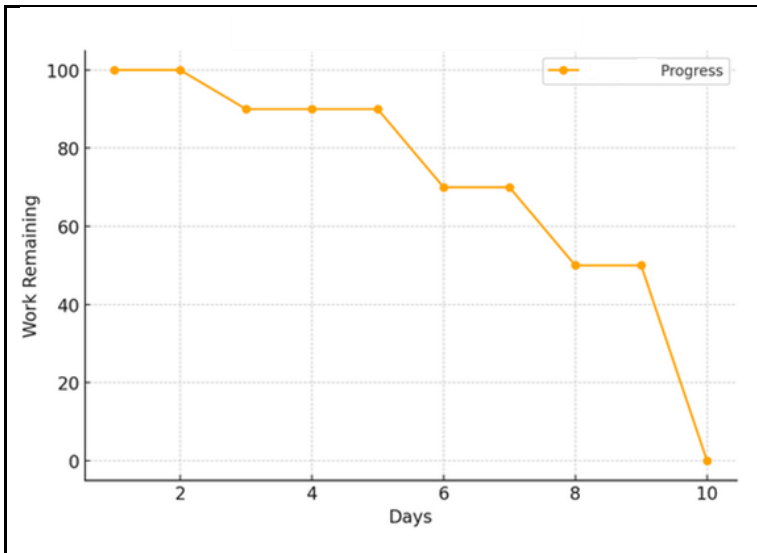


V (4. dan) =

Dan_završetka_sprinta =

V (7. dan) =

Dan_završetka_sprinta =



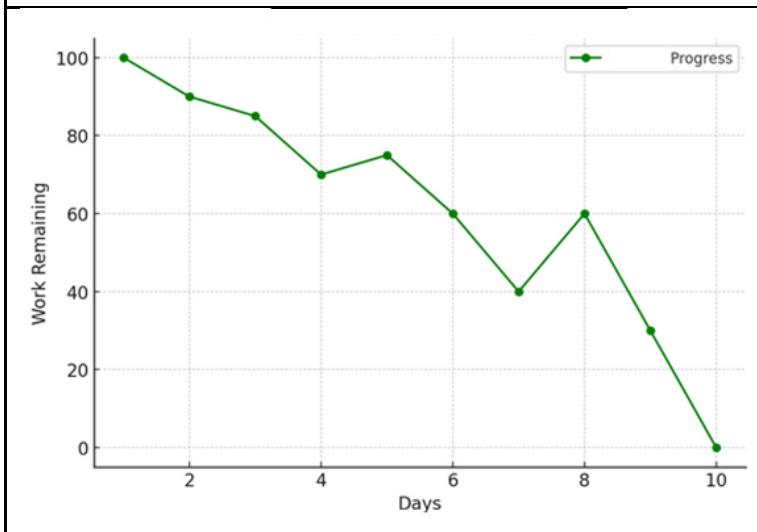
Sprint #2

V (4. dan) =

Dan_završetka_sprinta =

V (7. dan) =

Dan_završetka_sprinta =



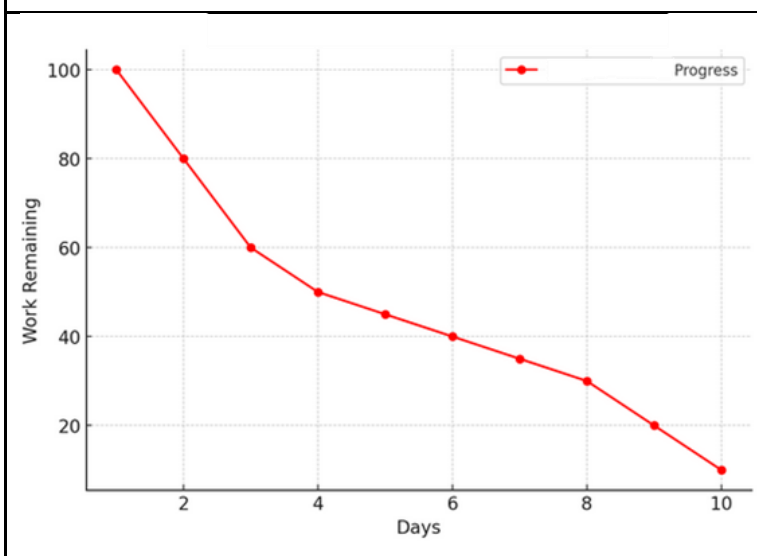
Sprint #3

V (4. dan) =

Dan_završetka_sprinta =

V (7. dan) =

Dan_završetka_sprinta =



Sprint #4

V (4. dan) =

Dan_završetka_sprinta =

V (7. dan) =

Dan_završetka_sprinta =

- c) Opisati koji su glavni događaji i aktivnosti u svakom Sprint-u. Svaki opisati u najviše dve rečenice, naznačiti maksimalno trajanje svakog događaja (u odnosu na definisano trajanje celog Sprint-a) i predložiti šta bi bili glavni zadaci u svakom od Sprint-ova.

2. [4] Razvija se softverski sistem onlajn naručivanja artikala jedne poznate nemačke modne kuće, za teritoriju Jugoistočne Evrope, sa 4 mikroservisa:

Naziv servisa	Cilj servisa	Opis servisa
<i>Order_service</i>	Servis za uslugu naručivanja artikala	Otkazite porudžbinu ako plaćanje ne uspe ili željeni artikal(i) nije/nisu dostupan/ni.
<i>Payment_service</i>	Servis za izvršavanje usluge plaćanja	Povratite uplatu, ako artikal nije dostupan ili isporuka ne uspe.
<i>Article_service</i>	Servis za rezervisanje (čuvanje artikla)	Otpustite rezervaciju artikla, ako isporuka ne uspe.
<i>Shipping_service</i>	Servis za praćenje usluge dostave	Naručilac može otkazati isporuku u svakom trenutku, dok ne dobije paket sa artiklom.

Ukoliko u razvoju koristimo projektni uzorak Saga sa koordinacijom mehanizmom orkestracije, napisati u programskom jeziku Python, klasu Orchestrator, sa najmanje četiri funkcije koje pozivaju svaki od ovih 4 servisa u sekvenci, i još 4 klase (po jednu za svaki od ovih servisa) sa potpisima glavnih funkcija (telo funkcije može da ima ispis samo po jedne poruke, odnosno uloge koju ta funkcija obavlja).

Rešenje:

Julski ispitni rok - praktični deo ispita -

Napomena: Upotreba štampane literature i komunikacionih uređaja je zabranjena i strogo kažnjiva. Dozvoljena je samo upotreba elektronskih materijala sa diska *Materijali*. Ispit traje 165 minuta.

Broj indeksa	Ime i prezime	Broj poena

3. Razvija se aplikacija za pravljenje treninga. Podrazumevano se učitava stranica za prijavljivanje korisnika, osim ako korisnik nije već prijavljen. Omogućeno mu je pritiskom linka za registraciju da pređe na stranicu za registraciju, gde može da kreira svoj nalog. Kada se prijavi, korisnik prelazi na stranicu za prikaz svih treninga koje je kreirao. Svaki trening sadrži niz vežbi, gde svaka vežba ima informaciju o prethodnoj i narednoj u okviru treninga. U nastavku se nalazi spisak stavki koji je potrebno realizovati (a pored svake stavke je naznačeno koliko poena nosi):
- a) [15] Implementirati aplikaciju po opisanoj specifikaciji u *Django* radnom okviru. U nastavku se nalazi raspodela funkcionalnosti po poenima.
- [1] Podignut je server, a na korenoj putanji aplikacije se podrazumevano otvara stranica za prijavu. Ukoliko je korisnik već prijavljen prelazi se na stranicu za prikaz treninga.
 - [1] Sistem je implementiran tako da nije moguće pristupiti ostalim funkcionalnostima ukoliko korisnik nije prijavljen na sistem.
 - [1] Prijavljeni korisnik uvek ima prikazan navigacioni meni, dok su neprijavljenom korisniku dostupne samo stranice za prijavu i registraciju.
 - [1] Prijavljeni korisnik može da se odjavi sa sistema pritiskom na odgovarajući link.
 - [2] Funkcionalnost prijave je implementirana sa svim neuspešnim scenarijima. Za ovu funkcionalnost (kao i registraciju) je potrebno koristiti ugrađenu korisničku klasu omogućenu od strane *Django* radnog okvira. Forma za prijavu treba da radi u skladu sa ugrađenim podrazumevanim ponašanjem formi koje nudi *Django* radni okvir za ovu funkcionalnost.
 - [2] Funkcionalnost registracije, na koju se prelazi preko linka sa stranice sa prijave je realizovana sa svim neuspešnim scenarijima. Forma za registraciju treba da radi u skladu sa ugrađenim podrazumevanim ponašanjem formi koje nudi *Django* radni okvir za ovu funkcionalnost.
 - [3] Nakon uspešne prijave se ispravno izlistavaju svi korisnikovi treninzi u vidu linkova. Pritiskom naziva treninga se odlazi na stranicu na kojoj se izlistavaju detalji treninga i sve vežbe onim redosledom kojim su zamišljene da se rade u treningu.
 - [2] Na stranici za detalje treninga postoji za svaku vežbu mogućnost brisanja u okviru treninga. U tom slučaju je potrebno na adekvatan način ažurirati redosled ostalih vežbi.
 - [2] Na stranici za detalje treninga postoji opcija dodavanja vežbe. Vežba je vezana za konkretan trening. Polje za unos teksta ne sme biti prazno kada se zahteva dodavanje opisa vežbe i predstavlja poslednju vežbu u okviru treninga.
- b) [3] Korišćenjem standardne *Python* bibliotečkog modula *unittest* i podrške za testiranje od strane *Django* radnog okvira, napisati test primere kojim ćete testirati funkcionalnost prijave aplikacije, koja je opisana u tekstu zadatka i implementirana u tački a). Nije dozvoljeno koristiti *Selenium WebDriver*.
- c) [2] Dokumentovati metode i klase aplikacije smisleno u skladu sa konvencijama *Django* radnog okvira. Omogućiti prikaz dokumentacije u administracionom panelu *Django* radnog okvira.

d) [2] Izgenerisati dijagram klasa modela u formatu PNG fajla.

Napomene:

- Poruke i putanje (ukoliko nisu navedene u postavci, uzeti razumnu pretpostavku o sadržaju poruka) treba da budu iste kao u postavci zadatka. Svaka poruka/putanja koja nije u skladu sa postavkom oduzima po 1 poen.
- U slučaju da se zahteva dokumentovanje aplikacije, metode i klase moraju biti smisleno dokumentovane, u suprotnom se dobija maksimalno 50% osvojenih poena za tu stavku.
- Na disku *Materijali* (N:) možete naći početni sadržaj projekta u okviru ZIP arhive pocetni_sadrzaj.zip.
- Predaja se vrši kopiranjem foldera sa projektom (**isključivo bez venv foldera**), i PNG fajla na disk *Rad* (L:). Ukoliko rešenje nije predato na disk za predaju u skladu sa zahtevanim formatom, oduzima se 5 poena.