

*Prolećni semestar, 2020/21*

*PREDMET: IT370 INTERAKCIJA ČOVEK-RAČUNAR*

Domaći zadatak: **10**

Ime i prezime: **Bogdan Janković**

Broj indeksa: **3920**

Datum izrade: **27.04.2021.**

# Tekst domaćeg zadatka:

Rad u grupama 4 studenta da se umreže putem google kolaborativne platforme. Potrebno je da preko ove platforme podelite uloge i rad na konceptualizaciji mobilne aplikacije za dostavljanje hrane - ili na neku drugu temu koju možete predložiti asistentu na odobrenje.

U toku rada na zadatku potrebno je obratiti pažnju na organizaciju tima i koristiti neku od online platformi za kolaboraciju i komunikaciju što ćete obrazložiti u eseju.

Predlažemo vam dve platforme za kolaboraciju unutar tima - slack i trello.

**Trello**omogućava da se dodele zadaci i podzadaci - taskovi i subtaskovi članovima tima, logički grupišu taskovi, postave rokovi i sl. kao i za razmenu različitih priloga (attachment) i poruka između članova tima. **Slack** je takođe kvalitetan izbor.

U ovom zadatku je potrebno proći kroz sve faze dizajna interfejsa mobilne aplikacije.

* Navesti članove tima (2-4 studenta u timu)
* Eksplikacija teme/ideje (aplikacija za dostavu hrane ili neka druga tema)
* Evaluacija ideje
* Persone – profil korisnika
* Kompetetivna analiza
* Mentalni model
* Task analiza
* Sadržaj aplikacije
* Strategija za osvežavanje (update, feed) sadržaja
* GUI Wireframe prototip -skica
* Testiranje GUIja

# Rešenje zadatka:

Kada je u pitanju alat za praćenje problema ili upravljanje zadacima, možda će biti teško znati odakle početi. Dostupno je toliko alata!

Jedan od najvećih uvida je da ne može biti sve za svakoga. Pruzanje funkcija je previše stvarno odnosno konfuzno i umesto da vaš alat učini vrednijim za veći broj ljudi, brzo može postati manje koristan za sve. Morate da odlučite za koga gradite svoj alat, koje su njihove tačke bola, a zatim smislite kako da im olakšate život.

Dakle, u osnovi, **GitKraken Boards** je dizajniran za programere i timove koji rade sa njima. Da bih vam brzo pokazao kako se GitKraken Boards izdvaja od konkurenata, uporediću ga sa poznatim alatom za praćenje zadataka: Trello. Pokazaću vam 10 načina na koje smo GitKraken ploče dizajnirali posebno imajući na umu razvojne programere, a do kraja ovog članka kladim se da ćete biti spremni da počnete da koristite GitKraken boards umesto Trello-a ili bilo kog drugog sistema za praćenje problema koji imate koristeći danas.

1. ***Sve potrebne informacije u jednom prikazu***

Veliki deo povećanja produktivnosti je brzo obavljanje stvari. Kada pokušavate da se brzo krećete, ne želite da morate da skačete po više ekrana da biste dobili relevantne informacije koje su vam potrebne. Ovde se pojavljuje glavna razlika u korisničkom interfejsu između GitKrakena i Trello-a.

Trello otvara modal i zaključava ekran kada kliknete da biste videli karticu. Možda ne zvuči kao fatalna mana, ali je frustrirajuće u onim trenucima kada samo želite da pregledate gomilu karata, kliknuli ste pogrešnu karticu ili samo želite još jednom da proverite drugu karticu pre nego što izvršite uređivanje.

1. ***Pronađite ono što tražite bržim filtriranjem***

Iako je važno da imate sve što vam treba u jednom prikazu, to takođe znači da morate brzo da filtrirate ono što ne treba da vidite!Uz Trelloovu pretragu prvo ukucate niz za pretragu, a zatim pritisnite Enter da biste pregledali rezultate. Ako želite da koristite filtere za kartice, pronaći ćete ih zakopane u meniju.

3. ***Brzo filtrirajte kartice prema kome su dodeljene***

Kada je reč o upravljanju vašim zadacima, želećete da znate šta vam je posebno određeno. U Trellu, kada kliknete na ikonu korisnika, otvara se modal za prikaz aktivnosti tog korisnika i nekoliko drugih stvari.Ako kliknete na ikonu korisnika na GitKraken pločama, ona će filtrirati vašu tablu tako da prikazuje samo kartice dodeljene izabranom korisniku. A pošto se kartice mogu dodeliti više ljudi, možete kliknuti na više čvorova da biste videli stavke dodeljene dvojcu ili timu.

4. ***Ne plaćajte dodatno za prikaz kalendara***

Većina programera je vrlo naviknuta da koristi kalendar kako bi videla koji predstojeći zadaci su planirani za dan, nedelju i tako dalje; vrlo je korisno za planiranje vremena.

Ako želite prikaz kalendara u Trellu, prvo ćete morati da primenite Uključivanje kalendara na svoj nalog. Ako već koristite jedno besplatno uključivanje, od vas će biti zatraženo da izvršite nadogradnju na Trello Business Class: plaćeni paket. U GitKraken tablama postoji besplatni, ugrađeni prikaz kalendara. Samo kliknite ikonu kalendara da biste videli zadatke sa predstojećim rokovima.

5. ***Prevucite i otpustite da biste promenili rokove***

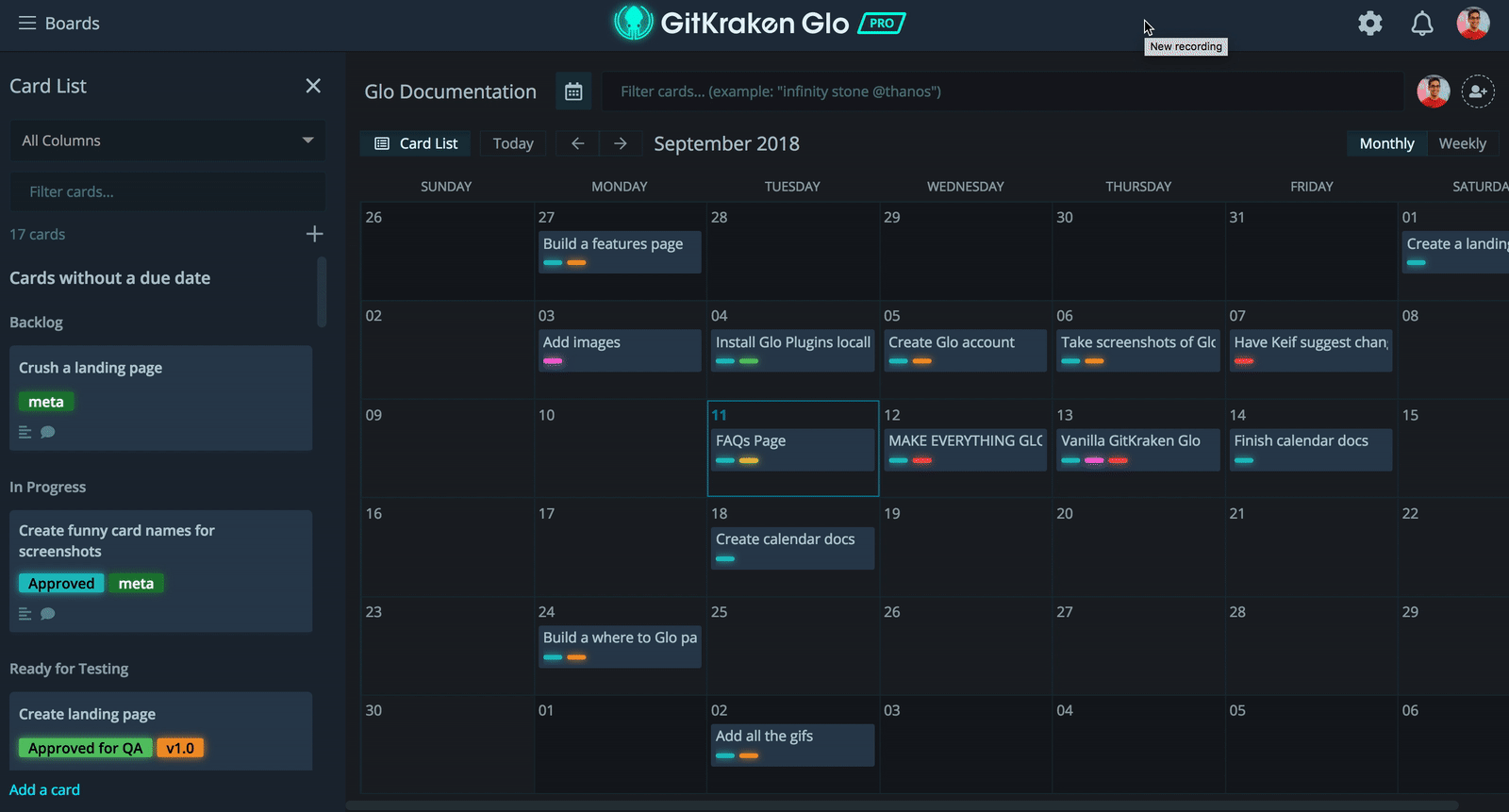
U GitKraken pločama, podrazumevano, leva tabla prikazuje listu kartica koje nemaju rokove dospeća, što olakšava prepoznavanje kartica kojima je potreban rok dospeća. Samo povucite i otpustite da biste dodelili kada treba da dospeju

6. ***Prilagodite izgled i osećaj***

Verovatno ste ovo već primetili: Trello dolazi samo u osnovnoj plavoj boji. Sa GitKraken pločama imate mogućnosti! Ploče GitKraken imaju i tamnu temu, „Glo“ temu, a ako niste spremni da pređete na tamnu stranu - ipak - imamo i svetlu temu.

7. ***Pratite zadatke u VSCode, Atom ili GitKraken Git GUI***

Manje prebacivanje konteksta je izuzetno važno za produktivnost programera.Trello ne nudi dodatke za Atom ili VSCode.Budući da je naš fokus na programerima, od početka smo dali prednost lakom pristupu GitKraken pločama. GitKraken pločama se može pristupiti direktno u GitKraken Git GUI-u, putem pregledača ili pomoću dodataka Atom ili VSCode. Ovo je promena igre za produktivnost, jer programeri mogu pristupiti svojim stavkama u alatima koje već koriste! Jednostavno ažurirajte svoj projekat, a zatim premestite ili ažurirajte svoju GitKraken Boards karticu.

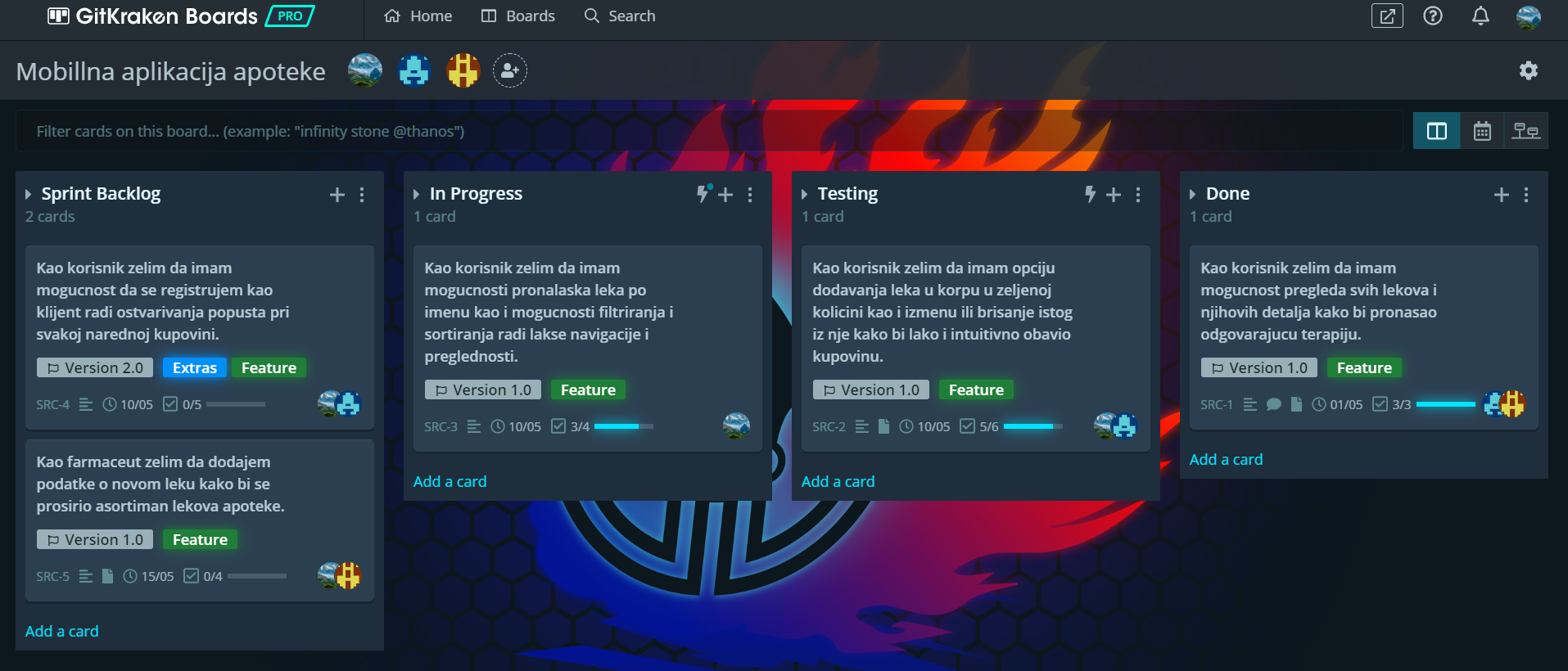


8. ***Iskoristite više GitHub problema sa tokovima posla***

U zajednici GitHub postoji 31 milion programera koji zajedno sarađuju na kodu; to je puno zadataka! GitHub problemi su ono što mnogi od ovih programera koriste za praćenje zadataka, grešaka i karakteristika. Dakle, ima smisla da vaš program za praćenje zadataka može da uvuče ove stavke umesto vas. Trello zvanično ne podržava GitHub Issue Sinc.

Alternativno, GitKraken Boards pružaju elegantnu integraciju, i baš kao i pogled kalendara, uključen je besplatno! Kada sinhronizujete ploču sa GitKraken odbora sa GitHub izdanjima, moći ćete da vizualizujete napredak GitHub izdanja kroz svoj tok posla.

Prikaz GitKraken Board-a korišćenog pri razvoju ove aplikacije:



**Prolazak kroz sve faze dizajna interfejsa mobilne aplikacije:**

* **Navesti članove tima (2-4 studenta u timu)**

Nikola Nikolic 3914

Bogdan Jankovic 3920

Andrej Kitanovic 3937

* **Eksplikacija teme/ideje (aplikacija za dostavu hrane ili neka druga tema)**

**Vizija sistema:**Apoteci je je potreban softver kao deo poslovnog sistema koji bi se koristio za samu prodaju lekova u apoteci ali i za ostale administrativne poslove. Apoteka poseduje magacin u koji svakodnevno pristiže roba dobavljača ali i iz kojeg se uzimaju proizvodi za prodaju u samoj apoteci i isto tako i za online prodaju. Apoteka ima zaposlene na pozicijama magacionera, online farmaceuta i farmaceuta u apoteci. Naši klijenti imaju mogućnost kupovine lekova uz recept ali i uz članske kartice uz koje ostvaruju dodatni popust. Ono što je trenutni cilj apoteke je da automatizuje proces naručivanja robe kod dobavljača kada se dostupna količina spusti ispod određene granice. Sa druge strane našim klijentima želimo da omogućimo online kontakt sa farmaceutom kao i online kupovinu sa upotrebom članske kartice za popust što je do sada jedino bilo moguće u samoj apoteci.

* **Evaluacija ideje**

Idejno rešenje aplikacije ima potencijala i smatram ovako razvijen i predstavljen projekat na temu sistema za upravljanje lancem apoteka veoma primenljivim u farmaceutskoj poslovnoj oblasti a isto tako i ono što je ovaj pristup razvoju omogućio je to da je sistem veoma lako nadogradiv i proširiv u nekom narednom vremenskom periodu.

* **Persone – profil korisnika**

Aplikacija je namenjena za profil korisika male i srednje informatičke pismenosti odnosno starijim korisnicima te i očekivano slabovidim. Za ovakav profil korisnika se mora razviti odgavarajući interfejs. Za ovakav profil korisnika se očekuje i dosta grešaka na koje sistem mora da odreaguje na odgovarajući način te bude i otporan na greške kod ovakvih korisnika.

* **Kompetetivna analiza**

Analizom konkurenata na tržištu je utvrđeno da je ovakav softverski sistem preko potreban u ovom trenutku i da će se vidno isticati u odnosu na konkurente koji nude mnogo manje mogućnosti odnosno funkcionalnosti, jednostavnosti i preglednosti.

* **Mentalni model**

Kako se očekuje da postoji neki osnovni mentalni model onlone kupovine kod korisnika, na početku će im svakako biti predstavljeno uputstvo te i kasnija opcija gledanja demo snimka upotrebe raznih funkcionalnosti.

* **Task analiza**

Task analizama je utvrđen raspored navigacionih elemanata koji shodno prati profile korisnika kojima je aplikacija namenjena te se i kroz zadatke analize vršilo ispitivanje očekivanja korisnika koja su iz dobijenih rezutata i implementirana. Pored toga dobio se uvid i u neke probleme koji su se javljaji kod korisika pri rešavanju zadatih zadataka te su u razvoju te greške otklonjene.

* **Sadržaj aplikacije**

Aplikacija će pored osnvnih karakteristika online prodavnice sadržati i prikaz najbližih apoteka shodno trenutnoj lokaciji korisnika, logovanje, registraciju i različite nivoe detaljnosti prikaza u zavisnosti od uloge korisnika u sistemu.

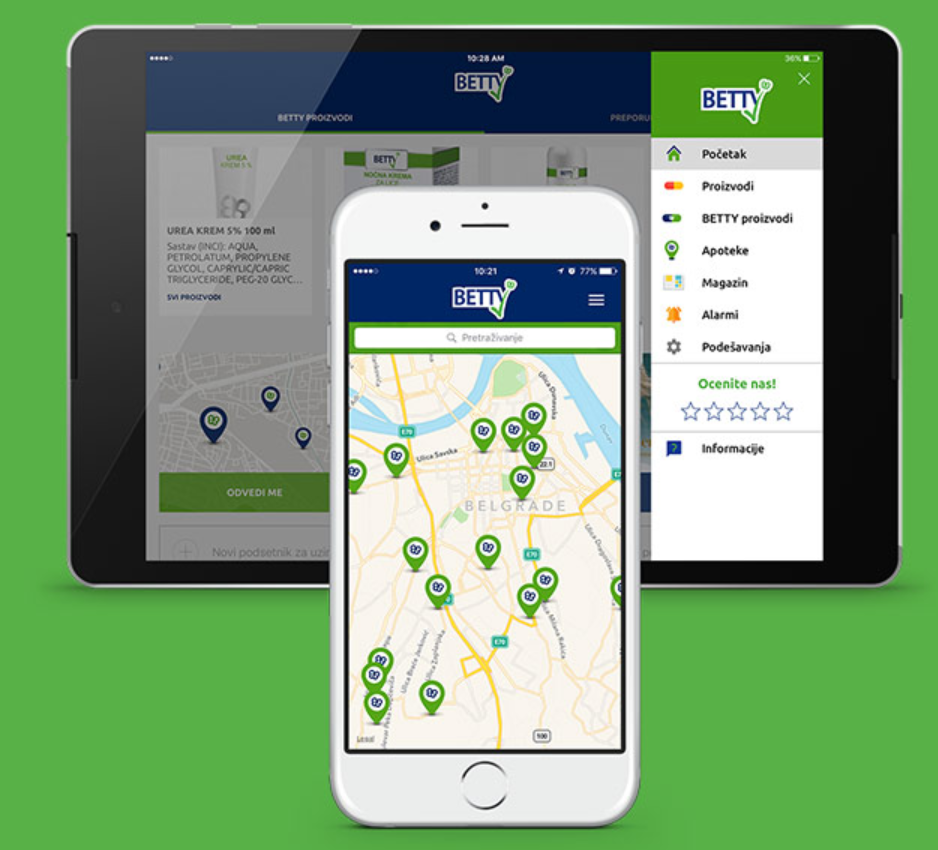
* **Strategija za osvežavanje (update, feed) sadržaja**

**Raspolozivost**

* Sistemi u apoteci i magacinu moraju da budu raspoloživi u toku radnog vremena.
* Online sistem mora biti raspoloživ u bilo kom trenutku izuzev u periodu redovnog održavanja sistema.

**Performanse**

* Sistem mora veoma brzo i često osvežavati informacije koje prikazuje zbog stalnih promena u bazi.
* Sistemu neće biti potrebno duže od 10 sekundi da se ponovo pokrene u slučaju prekida rada.
* **GUI Wireframe prototip –skica**

****

* **Testiranje GUIja**

Za testiranje korisničkog interfejsa možemo koristiti više različitih metoda. Pod

testiranjem korisničkog interfejsa podrazumevamo testiranje korisničkog iskustva a

ne integraciono ili jedinično testiranje same aplikacije tj. njenog programskog dela.

Pre testiranja moramo naravno imati prototip aplikacije koju želimo da testiramo. Na

prototipu kao i na gotovoj aplikaciji moramo izvršiti optimizaciju zadataka odnosno

dubinu svakog individualnog važnijeg zadatka dovesti do najviše tri koraka. Ovaj

postupak takođe smanjuje kognitivno naprezanje o kome smo već govorili. Što se tiče

metoda testiranja možemo razlikovati dva osnovna tipa:

* **Metoda posmatranja**
* **Metoda heurističke analize**

***Neki od testova koje ćemo primenjivati na ovom projektu koji su deo heurističke analize su:***

* **Five Second Test**
* **Click Test (Heatmap)**
* **Navigation Test**

# Korišćena literatura. : (opciono)

<https://www.gitkraken.com/boards>