SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Bašić

Ivan Šantalab

Ljiljana Pintarić

MusicMap

projekt iz kolegija analiza i razvoj programa

Varaždin, 2014.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Bašić

Studij: Informacijsko i programsko inženjerstvo

Ivan Šantalab

Studij: Informacijsko i programsko inženjerstvo

Ljiljana Pintarić

Studij: Baze podataka i baze znanja

MusicMap

projekt iz kolegija analiza i razvoj programa

Mentor:

Prof.dr.sc. Neven Vrček

Varaždin, studeni 2014.

Sadržaj

[1. Uvod 1](#_Toc403551958)

[2. Metodologija 3](#_Toc403551959)

[2.1. Artefakti 5](#_Toc403551960)

[3. Plan projekta 8](#_Toc403551961)

[3.1. Tehnologije i alati 9](#_Toc403551962)

[3.2. Product backlog 9](#_Toc403551963)

[3.3. Sprint backlog 10](#_Toc403551964)

[4. ERA I UML 12](#_Toc403551965)

[4.1. ERA 12](#_Toc403551966)

[4.1.1. Relacijski model 13](#_Toc403551967)

[4.1.2. Opis baze podataka 13](#_Toc403551968)

[4.2. UML 14](#_Toc403551969)

[4.2.1. Dijagram slučajeva korištenja 14](#_Toc403551970)

[4.2.2. Sekvencijalni dijagram – registracija 15](#_Toc403551971)

[4.2.3. Sekvencijalni dijagram – prijava 16](#_Toc403551972)

[4.2.4. Sekvencijalni dijagram – pregled događaja 17](#_Toc403551973)

[4.2.5. Sekvencijalni dijagram – pretraživanje 18](#_Toc403551974)

[4.2.6. Sekvencijalni dijagram – ažuriranje lokalne baze 19](#_Toc403551975)

[5. Literatura 20](#_Toc403551976)

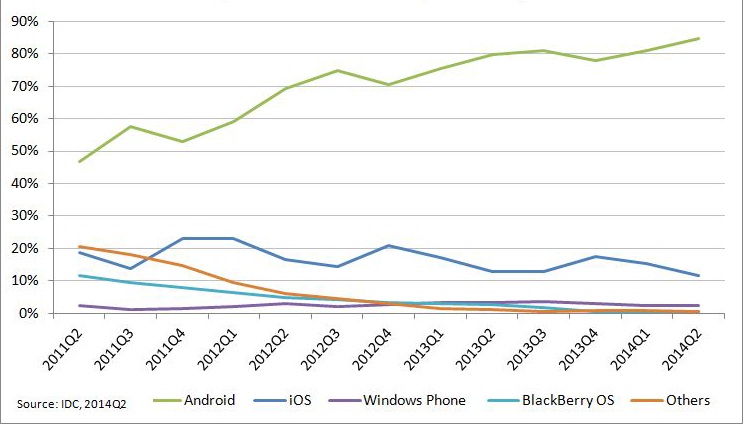
1. Uvod

MusicMap je interaktivna glazbena karta koja daje iscrpan pregled potencijalno zanimljivih i atraktivnih glazbenih događanja uz pregršt popratnih informacija te jednostavnu navigaciju. Uz opis i kartografski prikaz lokacija pojedinih događaja, ovaj vodič zahvaljujući integraciji sa raznim web servisima omogućava i mnoštvo multimedijalnih dodataka u obliku audio i video sadržaja te fotografija o određenom glazbeniku, bandu, festivalu ili klubu.

Aplikaciju ćemo razvijati u suradnji s tvrtkom Neurolab. Ona se bavi dizajnom korisničkih sučelja, WEB razvojem i programiranjem, multimedijom, videom i live streamingom, društvenim mrežama i marketingom, te cloud hostingom. Dobitnik je različitih nagrada iz područja kojima se bavi, te je suradnik nekih svjetski poznatih imena, kao što su Amazon Web Services, Limelight Networks, Rackspace, Altus IT i druge. Za detaljnije informacije posjetite [web stranicu tvrtke](http://www.neuralab.net/).

Kada se radi o operativnom sustavu za koji ćemo razvijati aplikaciju morali smo se odlučiti između iOS-a ili Androida. Windows Phone nismo uzimali u obzir zbog svima poznate njegove jako male zastupljenosti među korisnicima smartphone uređaja. Iako nismo mogli pogriješiti u odabiru između iOS-a ili Androida svejedno smo proučili koji operativni sustav je zastupljeniji, a samim time i „isplativiji“.

U drugom kvartalu 2014. godine Android je zauzimao 84.7% udjela na tržištu, a iOS se nalazio na drugom mjestu sa 11.7% udjela. Razlog tome je što Android pokriva niži segment tržišta.



Slika 1- Udio smartphone uređaja na globalnom tržištu  
[http://www.idc.com/prodserv/smartphone-os-market-share.jsp]

Na slici 1 se jasno vidi udio pojedinih smartphone operativnih sustava na tržištu. Razna istraživanja i trenutno stanje potvrđuju da je Android zastupljeniji operativni sustav te se očekuje da će taj udio zadržati i u budućnosti. U prilog tome idu i predviđanja IDC-a koja govore da će do 2018. godine udio iOS-a neznatno pasti, dok će za to vrijeme Windows Phone udio neznatno porasti, a Android će zadržati svoju poziciju na tržištu.  
(Izvor: Smartphone Market Share, Q2 2014.)

Sumiranjem svih ovih podataka koji se odnose na sadašnje stanje, ali govore dosta i o budućnosti mobilnih operativnih sustava, odlučili smo se na Android operativni sustav te ćemo za njega razviti aplikaciju.

1. Metodologija

Prilikom izrade ovog projekta koristiti ćemo SCRUM metodologiju. SCRUM je okvir, razvijen još u ranim devedesetima, unutar kojeg je moguće rješavati kompleksne adaptivne probleme korištenjem različitih procesa i tehnika, te produktivno i kreativno razvijati kvalitetna rješenja.

SCRUM okvir se sastoji od timova uz koje su povezane uloge, događaji, artefakti I pravila. Svaka od ovih komponenata ima neku namjenu te je neizostavan dio SCRUM-a.

Pravila povezuju događaje, uloge I artefakte, upravljajući vezama I interakcijom između navedenih.

SCRUM je empirijski orijentiran, odnosno temeljen na iskustvu. Također je iterativni, inkrementalni pristup fokusiran na kontrolu rizika I optimizaciji predvidljivosti, a kao I svaka druga implementacija empirijske kontrole procesa, temelji se na transparentnosti – važni aspekti procesa moraju biti vidljivi onima koji su odgovorni za rješenje, kontroli – SCRUM korisnici moraju često provjeravati artefakte I napredak s obzirom na cilj trenutnog sprinta, te je li došlo do kakvih neželjenih promjena, I adaptaciji – ukoliko dođe do devijacije nekih aspekata procesa, što bi moglo rezultirati neprihvatljivim proizvodom, potrebna je adaptacija.

SCRUM tim sastoji se od vlasnika proizvoda (eng. Product Owner), tima za razvoj (eng. Development Team), I SCRUM master-a.

Timovi se sami organiziraju te samostalno određuju kako najbolje završiti posao, I izvršavaju više različitih funkcija, budući da imaju sve potrebne vještine da završe posao bez da ovise o drugima koji nisu dio tima. Vidljivo je da je ovaj model dizajniran kako bi optimizirao fleksibilnost, kreativnost I produktivnost.

Vlasnik proizvoda je odgovoran za maksimizaciju vrijednosti proizvoda kojeg razvije tim. Također je odgovoran za upravljenje backlogom proizvoda koji uključuje popis svojstava proizvoda, njihov poredak koji bi trebao biti optimalan kako bi se postigli svi ciljevi, optimizaciju vrijednosti proizvoda tima, osiguranu transparetnost I vidljivost backloga, idući korak u razvoju, objašnjenja svih stavaka na potrebnoj razini detaljnosti. Budući da je vlasnik odgovoran za rješenje, cijela organizacija mora poštovati njegove odluke.

SCRUM master je odgovoran za svoj tim, odnosno za njegovo pridržavanje SCRUM teorije, pravila I prakse. On pomađe onima izvan tima shvatiti koje su interakcije sa timom prihvatljive I korisne, a koje nisu. Također, zadužen je za pomaganje vlasnika proizvoda na način da, primjerice, pronađe odgovarajuće tehnike za efektivno upravljanje proizvodnim backlogom, pomaže timu da shvati stavke backloga, da razumije planiranje proizvoda u empirijskom okruženju I slično, dok timu pomaže na način da uklanja prepreke napretku tima, savjetuje ih kako se organizirati I riješiti različite probleme I slično.

SCRUM događaji su, za razliku od sprinta, fleksibilni, te su sami po sebi prilika za kontrolu I adaptaciju.

Sprint je bit SCRUM-a, a označava vremenski period tijekom kojeg se razvijaju inkrementi rješenja. Sprintovi obično slijede jedan za drugim za vrijeme trajanja cijelog projekta. Svaki sprint mora imati definirano sljedeće: što se razvija, dizajn I plan razvoja, razvoj I proizvod razvoja. Sastoji se od: planiranja sprinta (eng. Sprint Planning), dnevnog SCRUM-a (eng. Daily Scrum), pregleda sprinta (eng. Sprint Review), I retrospektive (eng. Sprint Retrospective).

Planiranje sprinta događa se jednom u periodu vremena jednog sprinta, obično na početku, u najviše osam sati trajanja, pri čemu se SCRUM master brine da svaki događaj dobije svoje vrijeme I mjesto, te da svi sudionici razumiju ideju događaja. Planiranje odgovara na pitanja: što može biti završeno u ovom sprintu, kako će taj posao biti izvršen te koji su ciljevi sprinta.

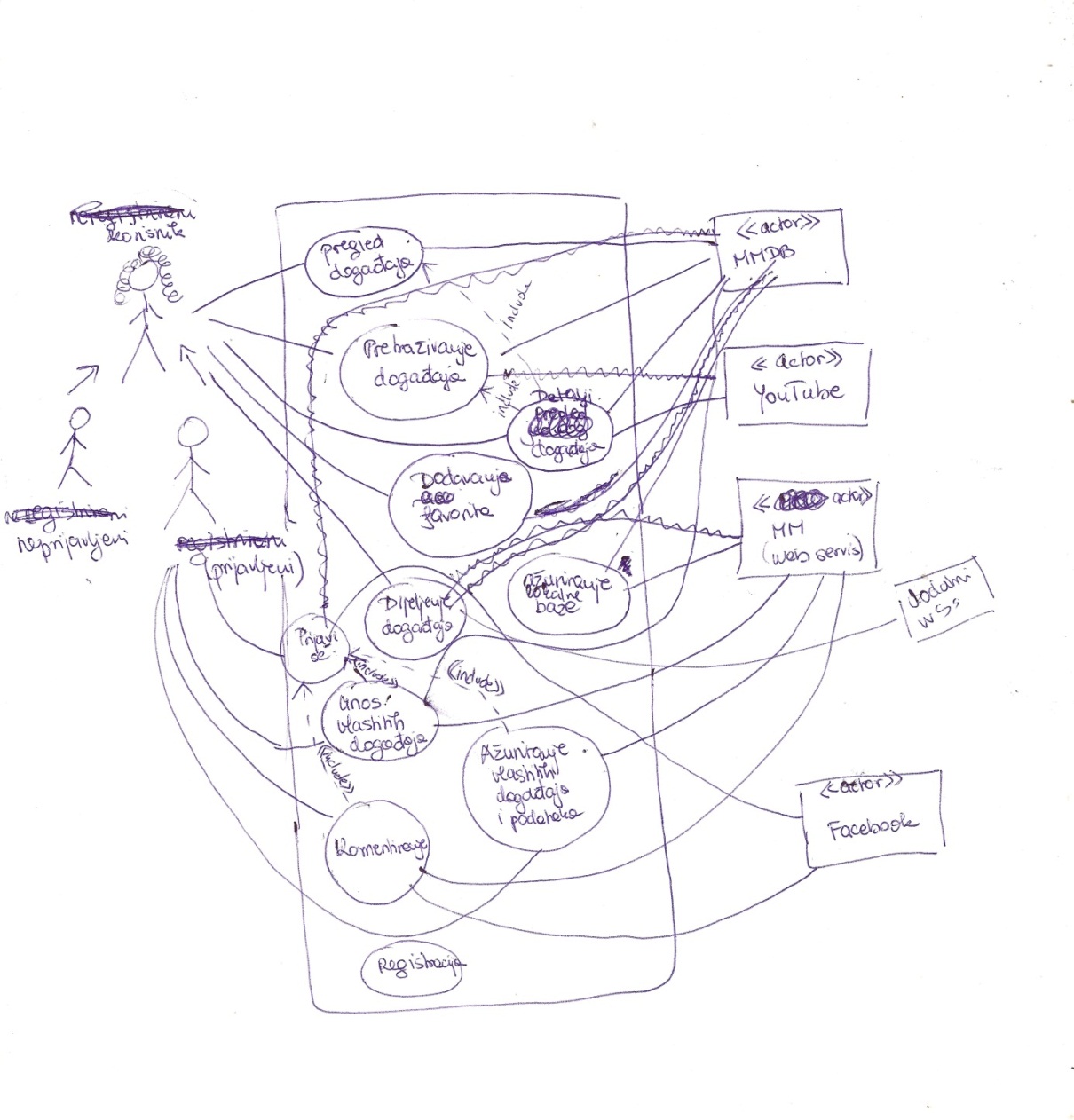
Dnevni SCRUM je 15-minutni dnevni sastanak tima na kojem se razgovara o stvarima koje su obavljene, koje tek trebaju biti obavljanje, sinkroniziraju se aktivnosti I radi plan za iduća 24 sata.

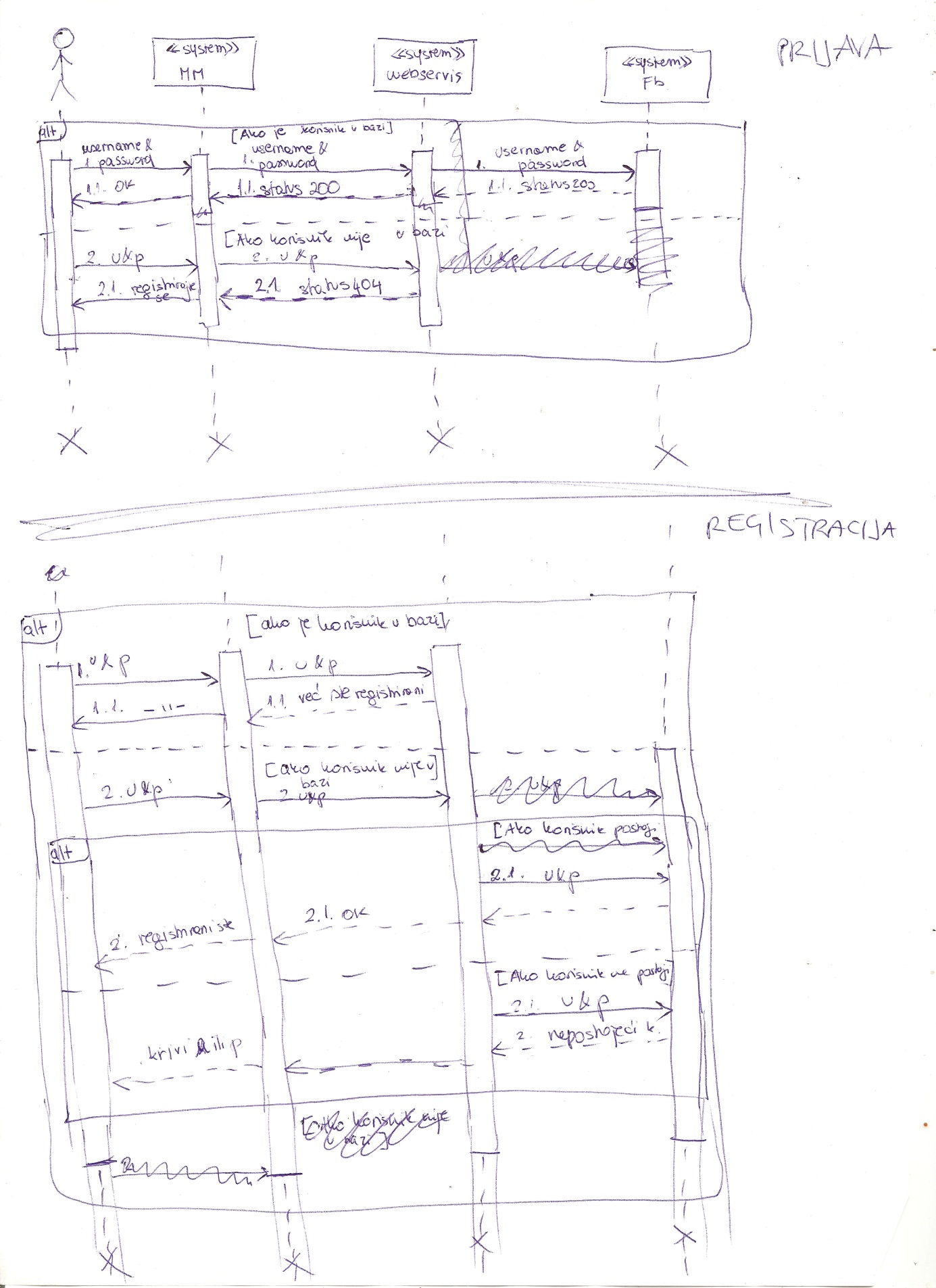
Pregled sprinta dolazi na kraju sprinta, I odnosi se na kontrolu inkrementa I adaptaciju backloga proizvoda ukoliko je to potrebno. Tijekom pregleda, tim I stakeholder-i kolaboriraju obavljen postao u sprintu, te sljedeće što je potrebno obaviti I kako optimizirati vrijednost.

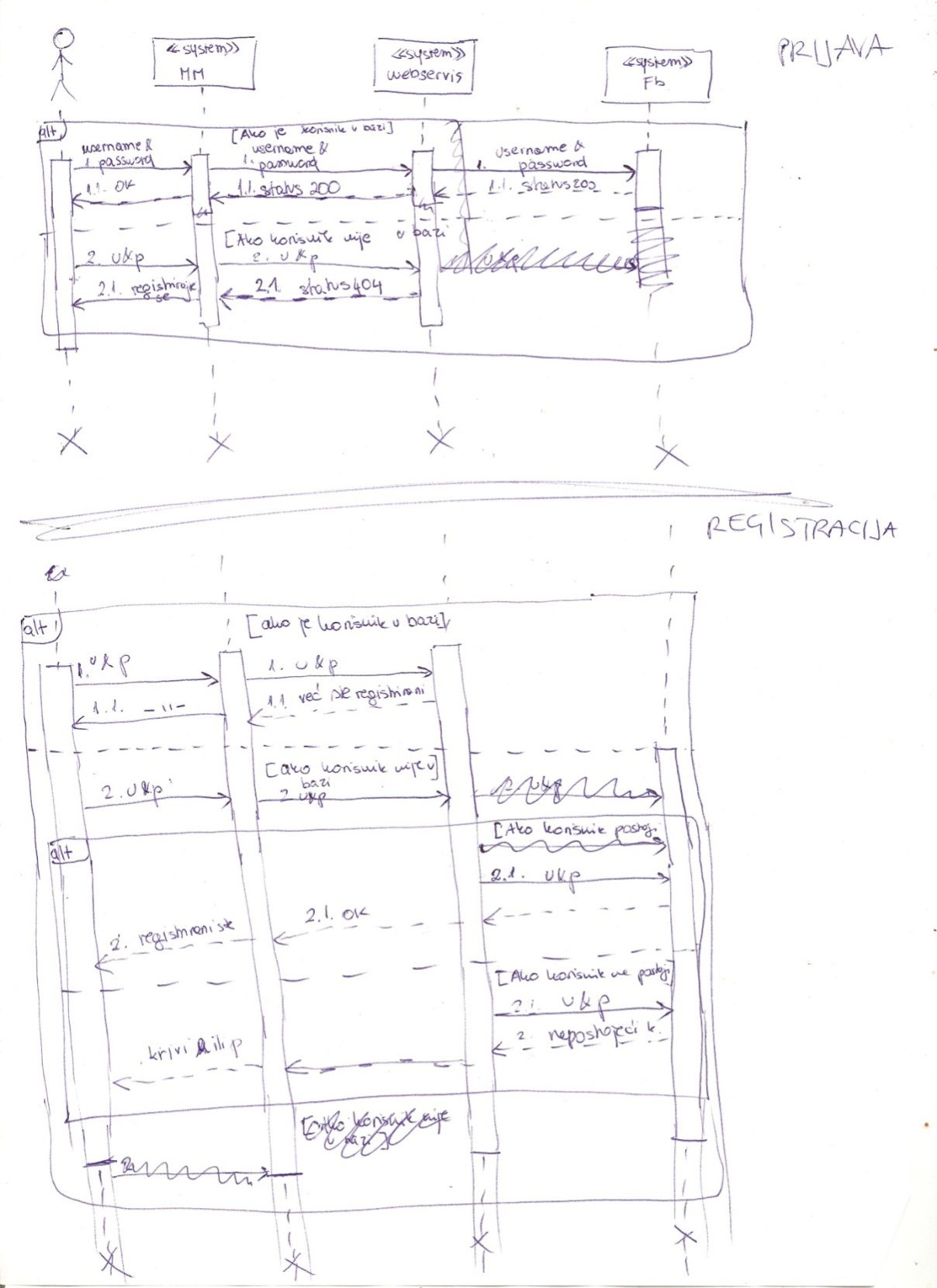
Retrospektiva sprinta je mogućnost za tim da prekontrolira sebe I kreira plan poboljšanja za idući sprint. Dakle, retrospektiva je smještena između pregleda sprinta I planiranja novog sprinta. Artefakti reprezentiraju posao ili vrijednosti koje osiguravaju transparentost ključnih informacija I mogućnosti za kontrolu I adaptaciju.

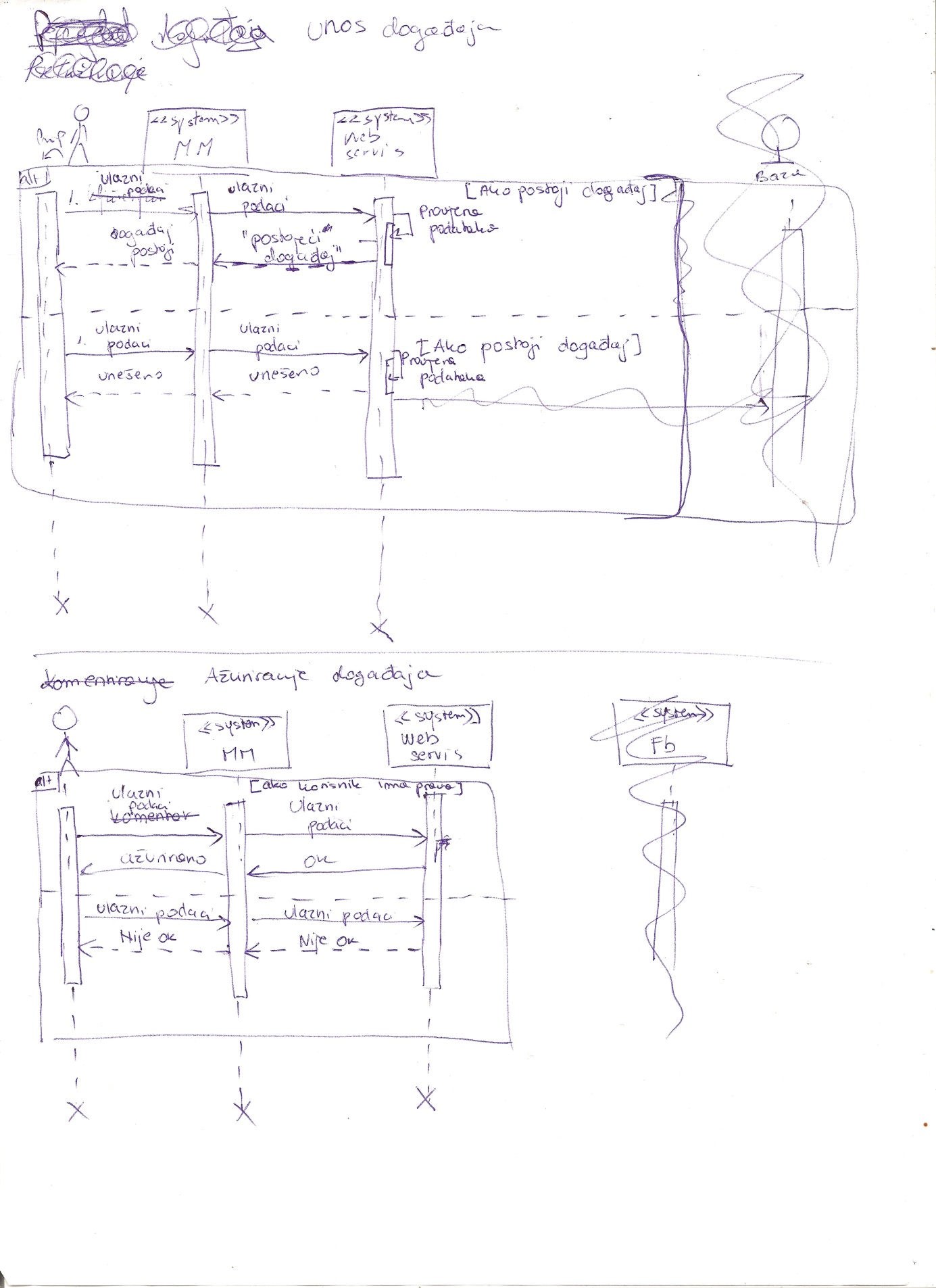
Sprint backlog je skup stavaka backloga proizvoda koje su odabrane za trenutni sprint, I plan za realizaciju novog inkrementa I ciljeva sprinta. Tim modificira sprint backlog kroz sprint.Inkrement je suma svih stavaka backloga proizvoda kompletiranog kroz sprint I vrijednost inkrementa svih prethodnik sprinteva. Na kraju svakog sprinta, mora biti gotov novi inkrement. (Izvor: SCRUM Guide)

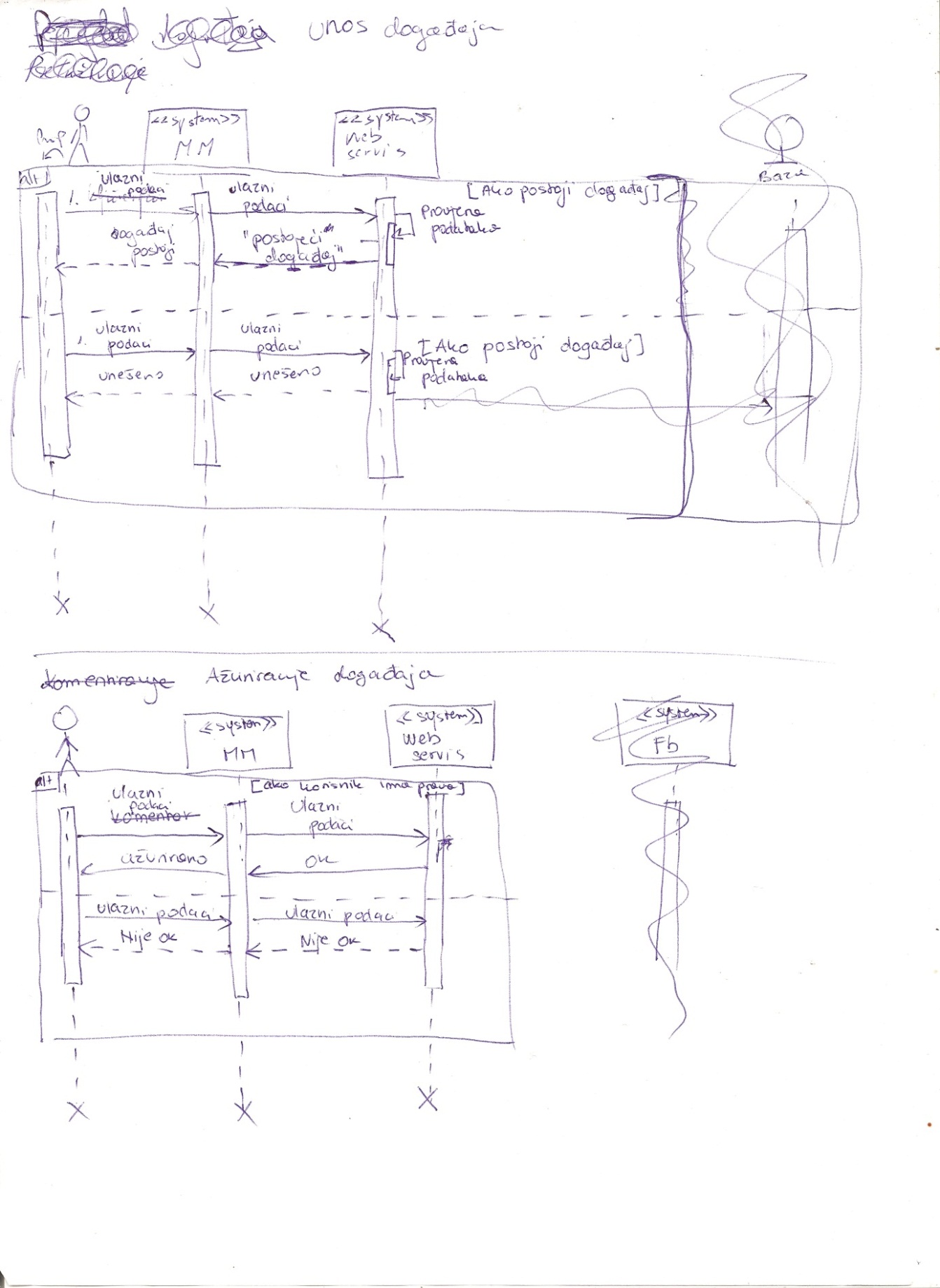
* 1. Artefakti











1. Plan projekta

Tim se sastoji od 3 člana: Matej Bašić, Ivan Šantalab i Ljiljana Pintarić.   
Poslove smo podijelili na sljedeći način koji je vidljiv u tablici 1, moguće su i kasnije preinake.

|  |  |  |  |
| --- | --- | --- | --- |
| Aktivnosti | Matej Bašić | Ivan Šantalab | Ljiljana Pintarić |
| SCRUM Master |  | + |  |
| Vlasnik proizvoda | + |  |  |
| Programiranje | + | + | + |
| Dizajn | + |  |  |
| Implementacija baze podataka |  |  | + |
| Dokumentacija | + | + | + |
| Product/Sprint Baclog |  | + | + |

Tablica 1 - Podijela poslova

Prema Scrum metodologiji razvrstali smo projektni plan u 3 faze:

1. faza

Prikupljanje i proučavanje zahtjeva

Planiranje ( I izrada dokumentacije)

Definiranje arhitekture i dizajna

2. faza

Dizajn

Razvoj funkcionalnosti aplikacije

3. faza

Testiranje

Integracija

Dokumentiranje završnog proizvoda

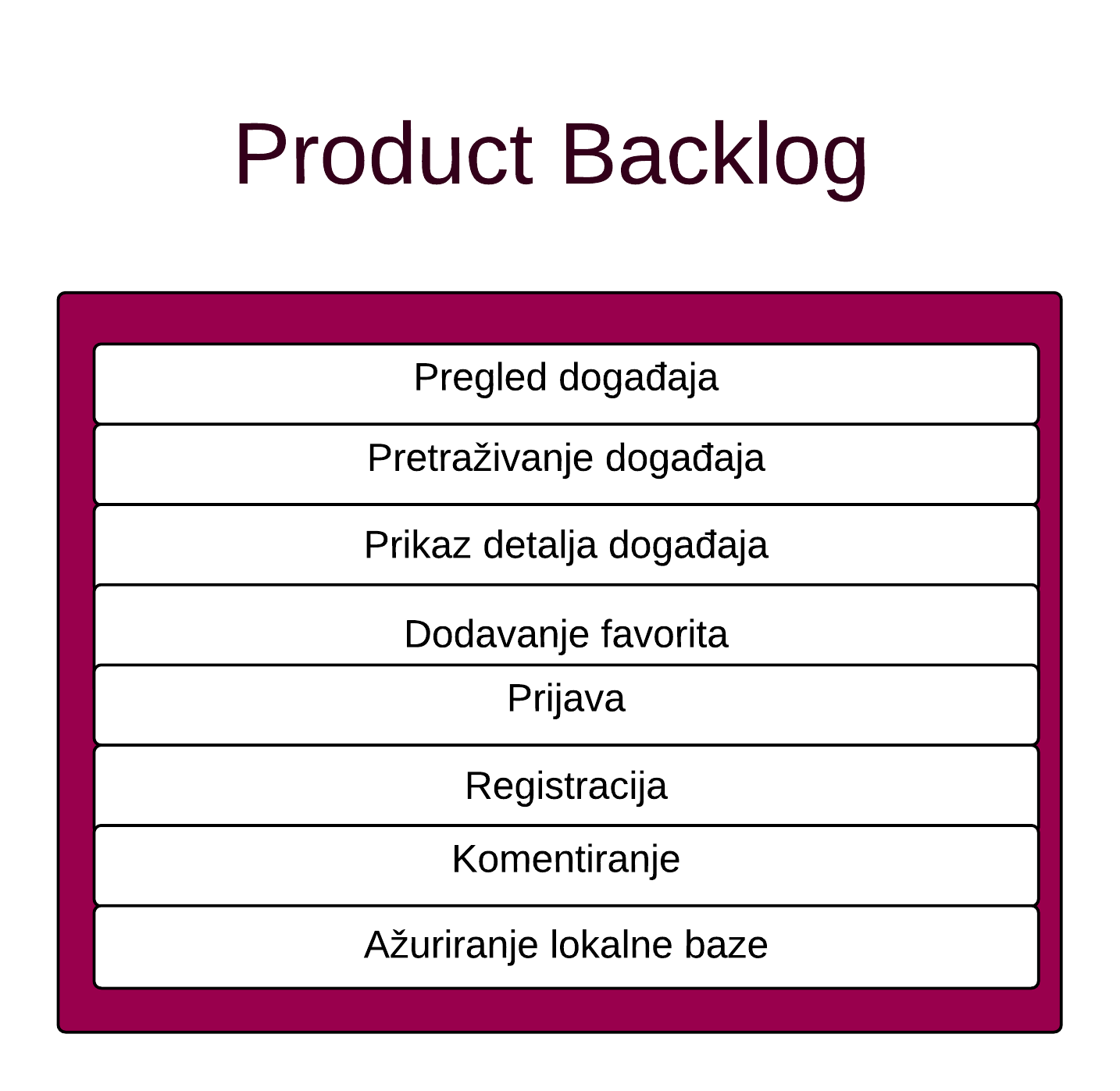
U prvoj fazi ćemo prikupiti I proučiti korisničke zahtjeve, isplanirati rad na razvoju aplikacije, te definirati arhitekturu I dizajn. U drugoj fazi baviti ćemo se izradom dizajna kao I razvojem svih funkcionalnosti aplikacije, dok ćemo u trećoj fazi obaviti testiranje I potrebne popravke, integraciju I dokumentiranje gotovog proizvoda. Ove faze korespondiraju sprintevima.

* 1. Tehnologije i alati

U prvoj fazi planiranja, koristit ćemo programski paket MS Office 2010 za izradu dokumentacije-. Za izradu UML dijagrama i ERA modela koristit ćemo Lucidchart.

U fazi razvoja programskog proizvoda koristit ćemo Eclipse (Luna) IDE for Java, SQLite, Genymotion, web servis tvrtke NeuroLab, Gimp I ALAT ZA IZRADU DIZAJNA, vlastita računala, Android OS v4.1.1 (Jelly Bean) uređaj.

* 1. Product backlog



* 1. Sprint backlog

**Sprint 1 backlog**

Prikupljanje i proučavanje zahtjeva

Planiranje ( I izrada dokumentacije)

Definiranje arhitekture i dizajna

**Sprint 2 backlog**

Dizajn

Razvoj funkcionalnosti aplikacije

**Sprint 3 backlog**

Testiranje

Integracija

Dokumentiranje završnog proizvoda



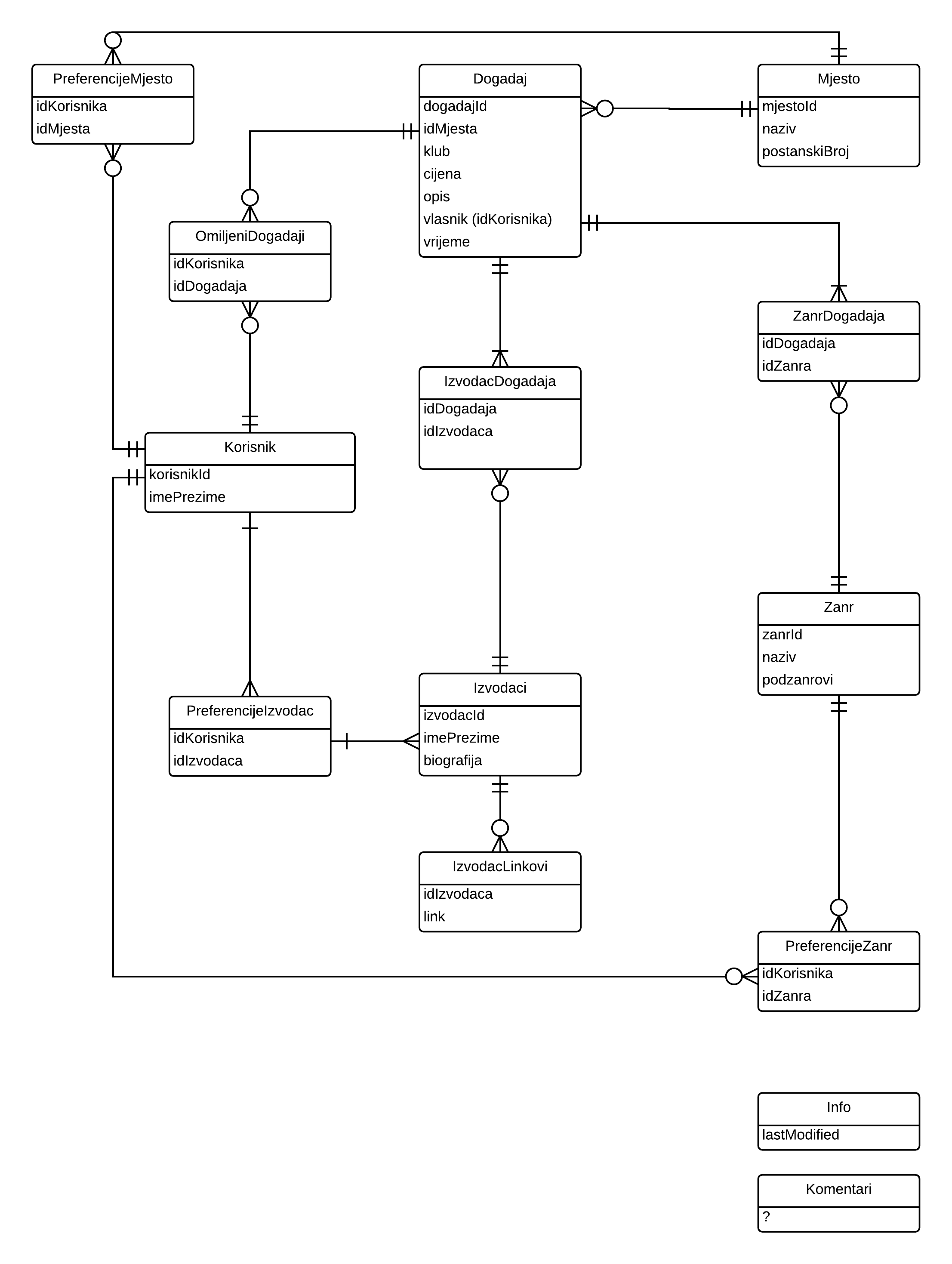
* 1. Troškovi aplikacije

U tablici 2 su prikazani troškovi aplikacije. Iz te tablice se vidi da su najveći trošak dizajn i razvoj funkcionalnosti aplikacije. Razlog tome je taj što su to najzahtjevnije aktivnosti. Ukupni trošak naše aplikacije iznosi 8250kn bez PDV-a, s PDV-om od 25% to iznositi 10312.50kn.

|  |  |  |
| --- | --- | --- |
| Redni broj | Aktivnost | Iznos |
| 1. | Prikupljanje i proučavanje zahtjeva | 450.00kn |
| 2. | Planiranje i izrada dokumentacije | 550.00kn |
| 3. | Definiranje arhitekture i dizajna | 400.00kn |
| 4. | Dizajn | 1850.00kn |
| 5. | Razvoj funkcionalnosti aplikacije | 3950.00kn |
| 6. | Testiranje | 400.00kn |
| 7. | Integracija | 350.00kn |
| 8. | Dokumentiranje završnog proizvoda | 300.00kn |
| Ukupno: | | 8250.00kn |

Tablica 2 – Troškovi aplikacije

1. ERA I UML
   1. ERA



* + 1. Relacijski model

Korisnik(korisnikId int, imePrezime varchar(50))

Događaj(događajId int, idMjesta int, klub varchar(50), cijena money, opis text, vlasnik int, vrijeme timestamp)

Mjesto(mjestoId int, naziv varchar(50), poštanskiBroj varchar(10))

Žanr(žanrId int, naziv varchar(50), podžanrovi text)

Izvođači(izvođačId int, imePrezime varchar(50), biografija text)

IzvođačDogađaja(idDogađaja int, idIzvođača int)

ŽanrDogađaja(idDogađaja int, idŽanra int)

OmiljeniDogađaji(idKorisnika int, idDogađaja int)

PreferencijeIzvođač(idKorisnika int, idIzvođača int)

IzvođačLinkovi(idIzvođača int, link text)

PreferencijeMjesto(idKorisnika int, idMjesta int)

PreferencijeŽanr(idKorisnika int, idŽanra int)

Info(lastModified timestamp)

Komentari(?)

* + 1. Opis baze podataka

Korisnik – informacije o korisnicima aplikacije

Događaj – informacije o događajima

Mjesto – popis svih mjesta u kojima se mogu održavati glazbeni događaji

Žanr – informacije o žanrovima i njihovim podžanrovima

Izvođači – informacije o izvođačima koji se mogu pojaviti na glazbenim događajima

ŽanrDogađaja – informacije o žanru/žanrovima događaja

OmiljeniDogađaji – korisnikove preferencije vezane uz glazbene događaje

PreferencijeIzvodač – korisnikove preferencije vezane uz izvođače

IzvođačLinkovi – linkovi na kojima se mogu pronaći dodatne informacije o izvođačima

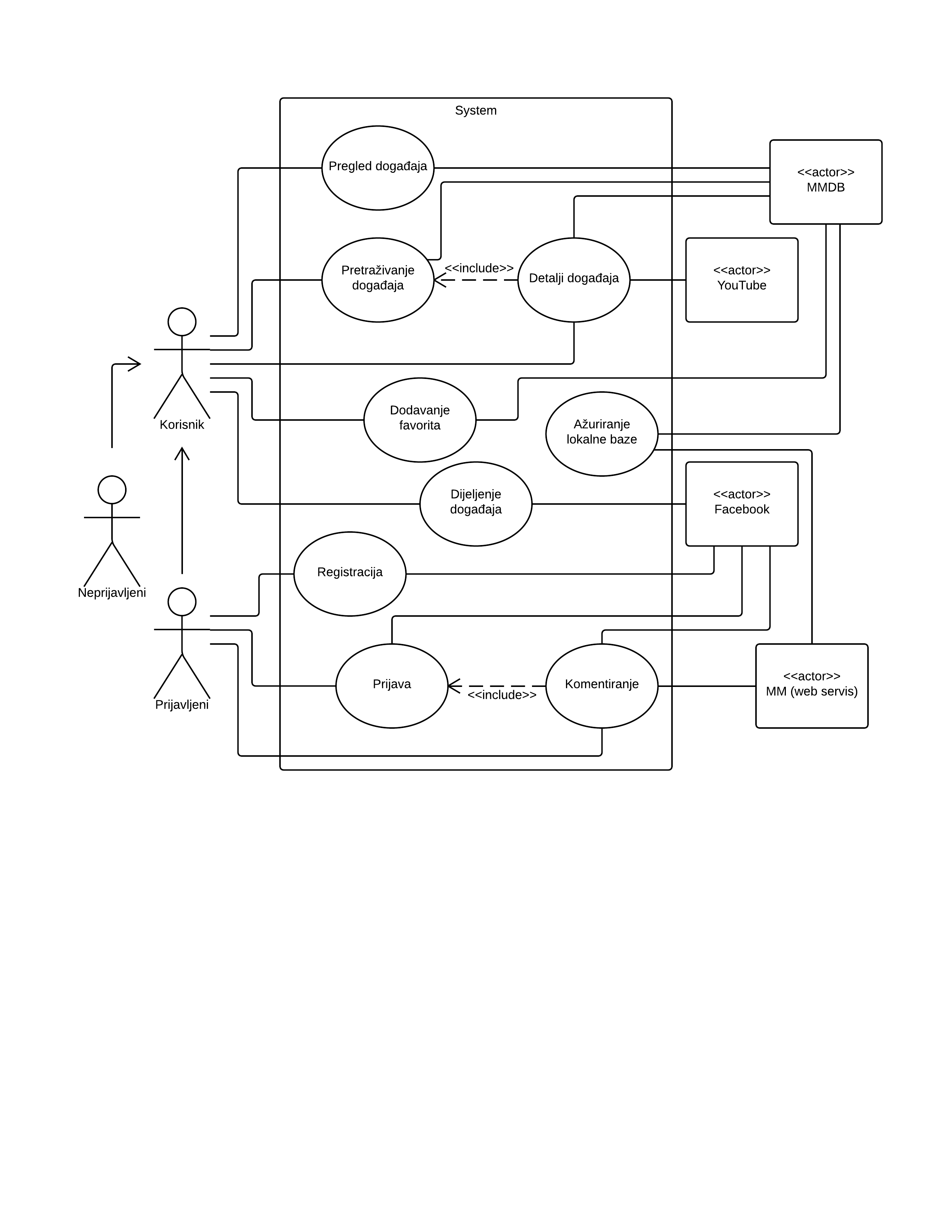
PreferencijeMjesto – korisnikove preferencije vezane uz mjesto izvođenja

PreferencijeŽanr – korisnikove preferencije žanra

Info – sadrži informaciju o posljednjem ažuriranju lokalne baze

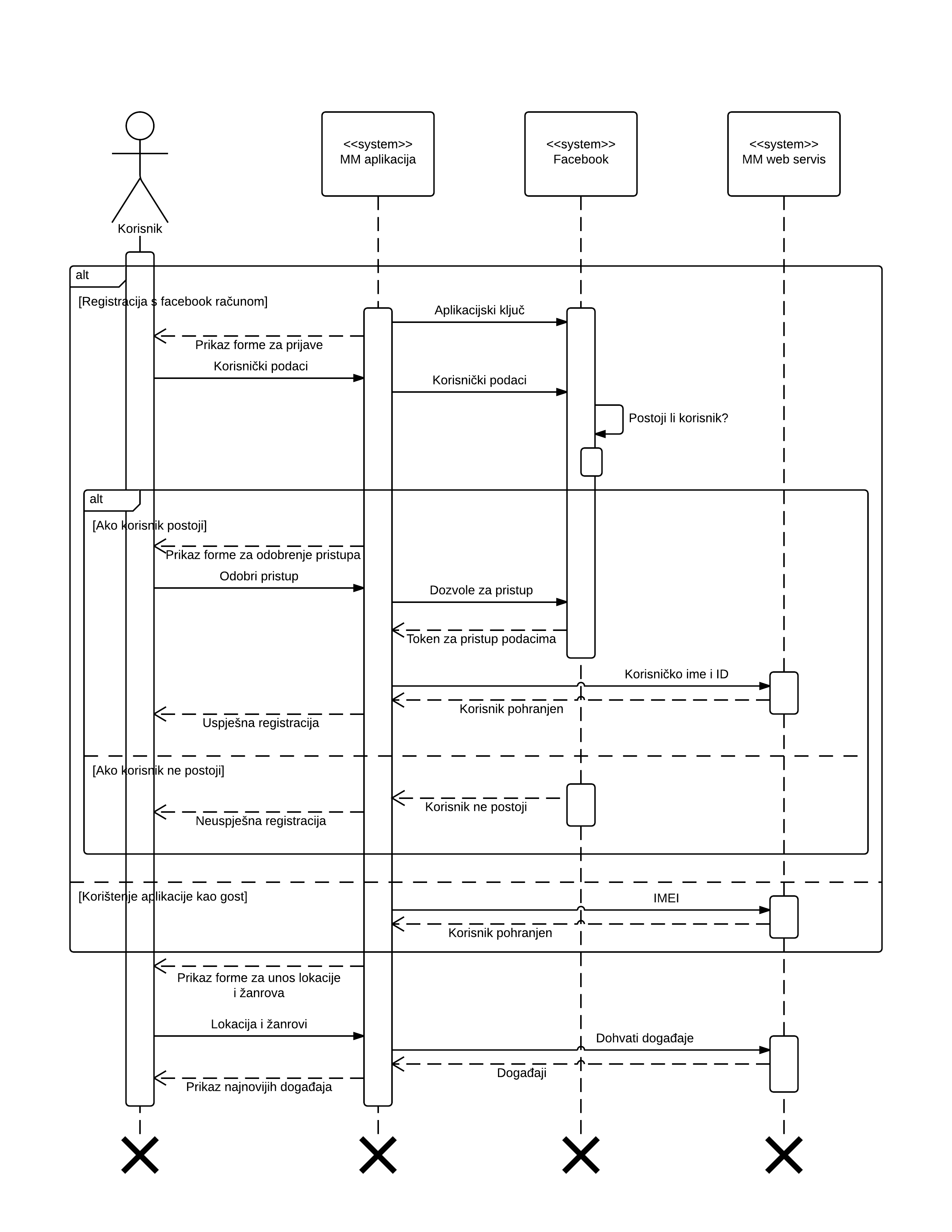
Komentari - ?

* 1. UML
     1. Dijagram slučajeva korištenja



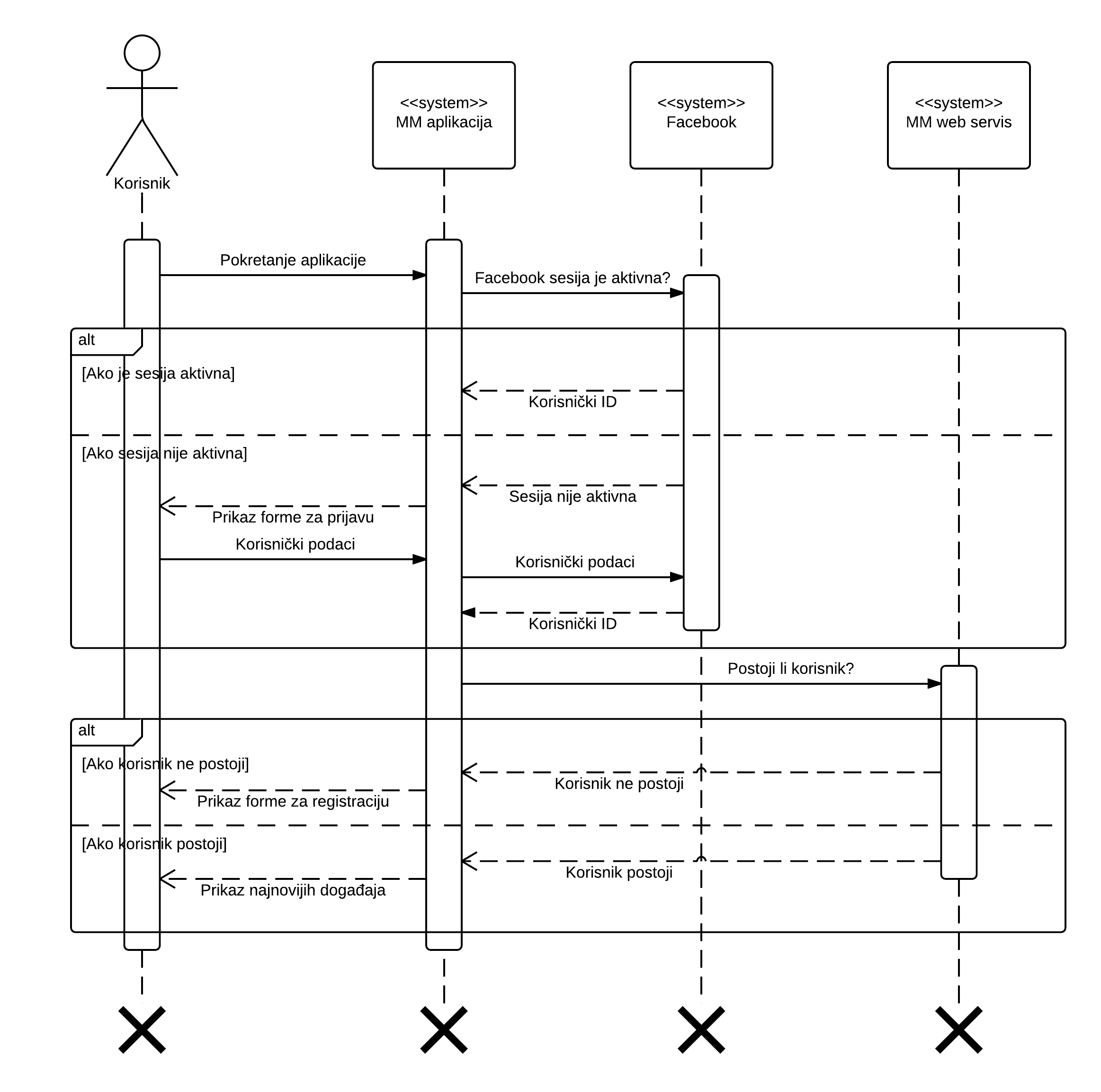
Dijagram slučajeva korištenja prikazuje tipove korisnika, mogućnosti korištenja aplikacije te vanjske servise koji se pritom koriste. Kao što se vidi iz slike, za aplikaciju postoje dva tipa korisnika – običan i prijavljeni korisnik, pri čemu je facebook korisnik zapravo registrirani/prijavljeni korisnik. Običan korisnik može pregledavati glazbene događaje, i pretraživati ih, gledati detalje događaja, dodavati vlastite favorite te dijeliti događaje putem društvenih mreža, SMS-a i e-maila. Osim ovih mogućnosti, prijavljeni korisnik može se registrirati i prijaviti korištenjem svojeg Facebook računa, te komentirati događaje također objavljujući javno svoje komentare putem Facebooka. Za realizaciju ovih mogućnosti, koristit ćemo, osim Facebooka i druge vanjske servise poput Google Maps, YouTube i web servis tvrtke Neurolab, dok ćemo za aktivnosti poput pregledavanja događaja koristiti lokalnu bazu u koju ćemo prethodno spremiti podatke sa nadređenog web servisa.

* + 1. Sekvencijalni dijagram – registracija



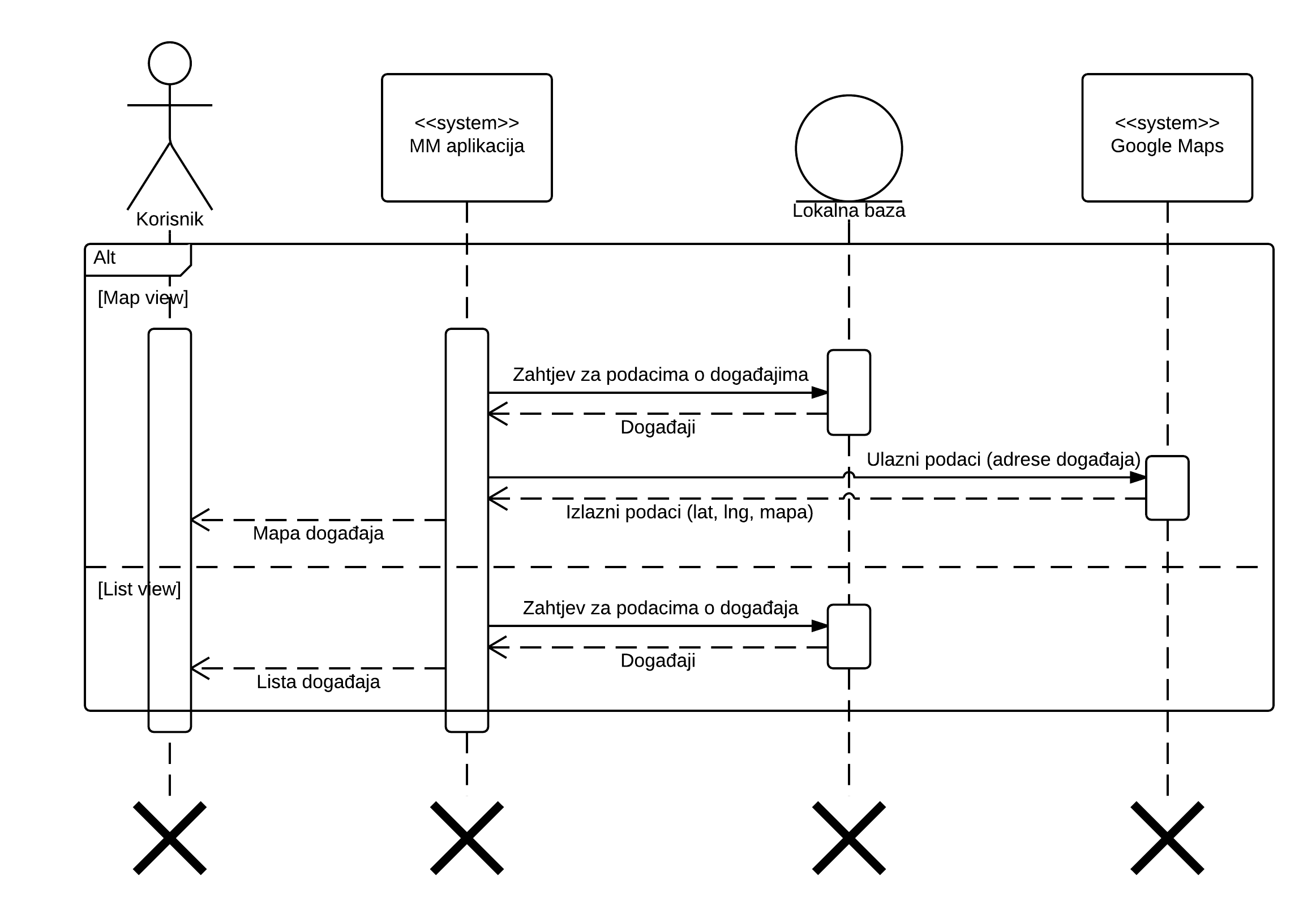
Ukoliko se korisnik želi registrirati, mora imati Facebook račun. Prvo upisuje korisničke podatke u formu aplikacije nakon čega se na Facebook serveru provjerava postoji li korisnik. Ako postoji, korisniku se prikazuje forma za odobrenje pristupa podacima koju on treba odobriti, nakon čega se Facebooku šalju dozvole za pristup, a on vraća token za pristup podacima. Aplikacija šalje korisničko ime i id na web servis aplikacije, te korisnika obavještava o uspješnoj prijavi. Ukoliko korisnik ne postoji, vraća se informacija o pogrešci i korisnik može pokušati unijeti nove podatke ili odustati. Običan korisnik može aplikaciju koristiti kao gost, pri čemu aplikacija na svoj web servis pohranjuje IMEI te obavještava aplikaciju da je korisnik pohranjen. Oba načina prijave završavaju unosom lokacije i preferiranih žanrova da bi završili prikazom najnovijih događaja filtriranih na temelju unesenih podataka.

* + 1. Sekvencijalni dijagram – prijava



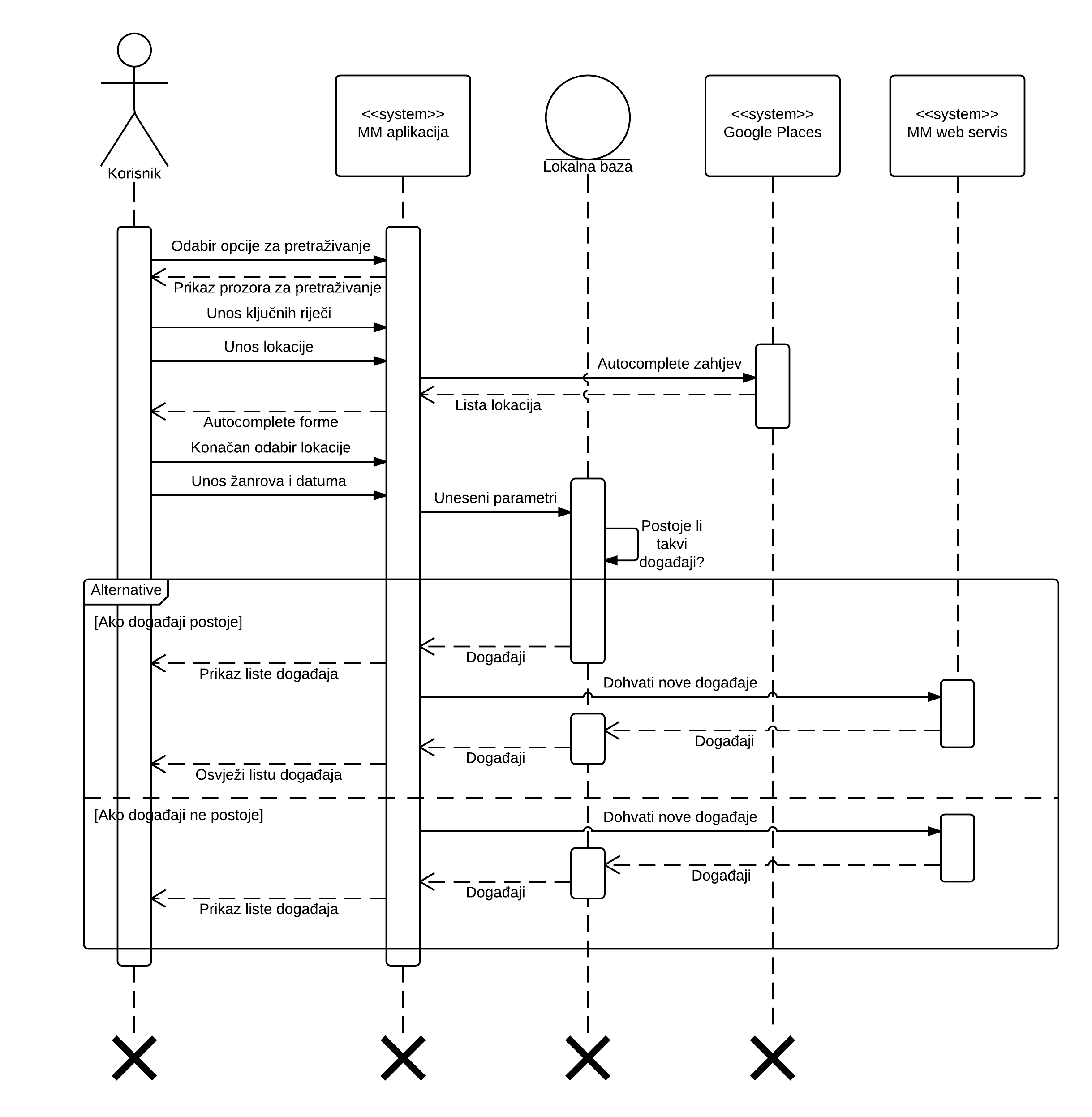
Nakon što je već registriran, korisnik se može prijaviti sa svojim Facebook računom, pri čemu se provjerava je li Facebook sesija aktivna, te ukoliko je, aplikacija dobiva ID korisnika s Facebooka. U suprotnome, potrebna je prijava korisnika na Facebook servis. Ukoliko korisnik postoji u bazi MM web servisa, može koristiti aplikaciju, u suprotnome, mora proći registraciju opisanu u prethodnom dijagramu.

* + 1. Sekvencijalni dijagram – pregled glazbenih događaja



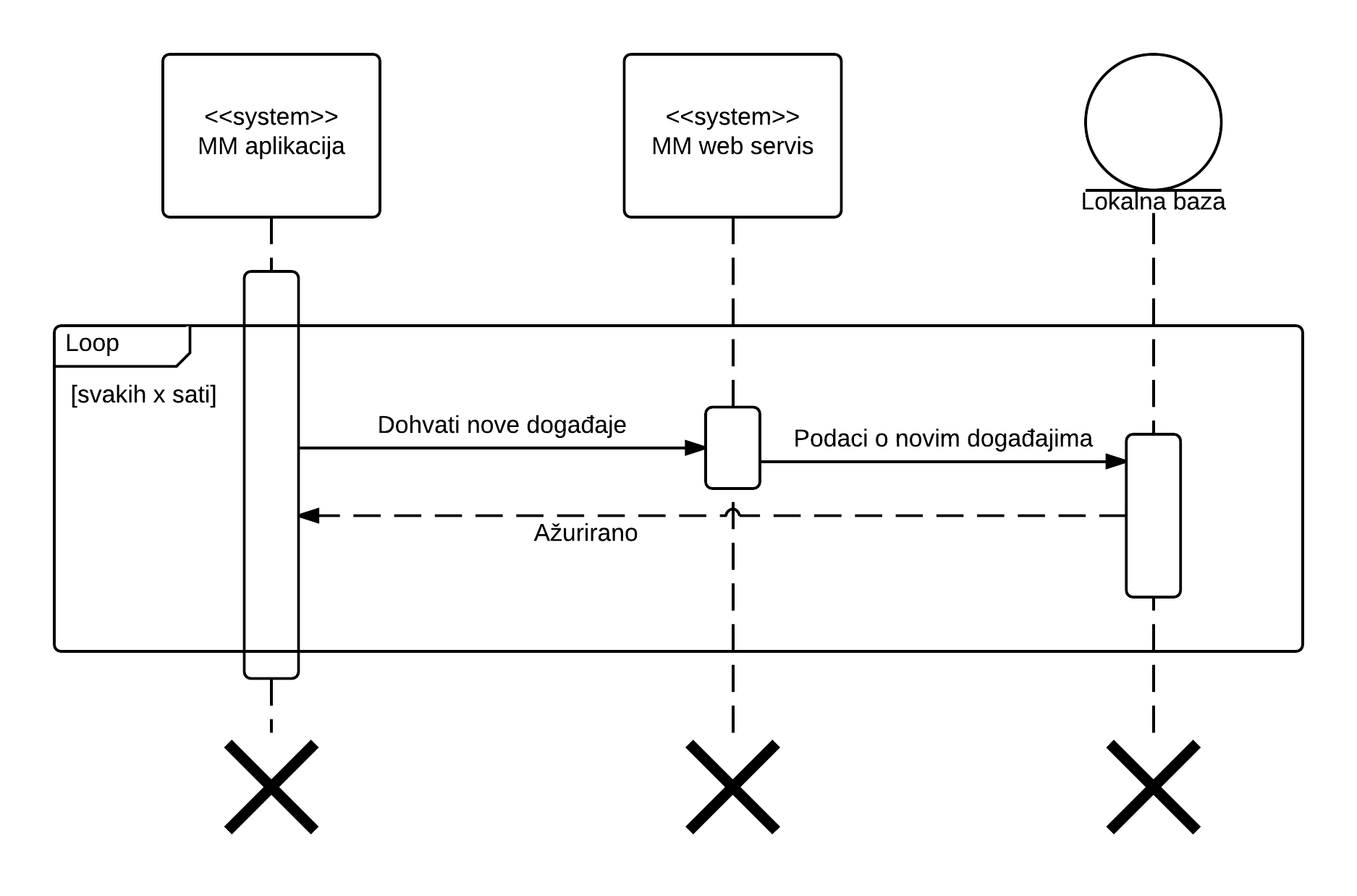
Prilikom pregleda događaja lokalnoj bazi se šalje zahtjev da prikaže kartu, a da bi to bilo moguće, potrebno ju je dohvatiti sa Google Maps servera nakon čega se korisniku prikazuje mapa sa markerima koji prezentiraju događaje. Također je moguće događaje prikazati kao listu pri čemu se samo učitavaju podaci iz lokalne baze.

* + 1. Sekvencijalni dijagram – pretraživanje



Korisnik može pretraživati događaje prema ključnim riječima, lokaciji, žanrovima te datumu održavanja. Prilikom unosa lokacije koristi usluge Google Places servisa, šaljući zahtjev za automatsko popunjavanje forme, nakon čega mu se nudi lista mjesta koje može odabrati. Nakon unosa konačne lokacije i podešavanja ostalih mogućnosti, aplikacija pretražuje bazu i prikazuje događaje ako postoje. Zbog brzine odgovora, događaji se, ukoliko postoje, prvo dohvaćaju iz lokalne baze te se prikazuju korisniku. Nakon toga, lokalna baza se osvježava te se prikazuju novi događaji ako odgovaraju parametrima pretrage. U suprotnome, prvo ažuriramo bazu te prikazujemo događaje.

* + 1. Sekvencijalni dijagram – ažuriranje lokalne baze



Lokalna baza ažurira se svakih nekoliko sati pri čemu se dohvaćaju novi događaji sa web servisa na temelju preferiranih žanrova i lokacije.

1. Literatura
2. SCRUM, The SCRUM Guide, <http://www.scrumguides.org/scrum-guide.html>, Dostupno 29.10.2014.
3. Smartphone OS Market Share Q2 2014., <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>, Dostupno 02.11.2014.
4. The iPhone 6 Had Better Be Amazing And Cheap, Because Apple Is Losing The War To Android,<http://www.businessinsider.com/iphone-v-android-market-share-2014-5>, Dostupno 02.11.2014.