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, Baze podataka i baze znanja

Ivan Šantalab

Studij: Informacijsko i programsko inženjerstvo, Baze podataka i baze znanja

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 2](#_Toc403551959)

[2.1. Artefakti 4](#_Toc403551960)

[3. Plan projekta 7](#_Toc403551961)

[3.1. Tehnologije i alati 8](#_Toc403551962)

[3.2. Product backlog 8](#_Toc403551963)

[3.3. Sprint backlog 9](#_Toc403551964)

[4. ERA I UML 10](#_Toc403551965)

[4.1. ERA 10](#_Toc403551966)

[4.1.1. Relacijski model 11](#_Toc403551967)

[4.1.2. Opis baze podataka 11](#_Toc403551968)

[4.2. UML 12](#_Toc403551969)

[4.2.1. Dijagram slučajeva korištenja 12](#_Toc403551970)

[4.2.2. Sekvencijalni dijagram – registracija 13](#_Toc403551971)

[4.2.3. Sekvencijalni dijagram – prijava 14](#_Toc403551972)

[4.2.4. Sekvencijalni dijagram – pregled događaja 15](#_Toc403551973)

[4.2.5. Sekvencijalni dijagram – pretraživanje 15](#_Toc403551974)

[4.2.6. Sekvencijalni dijagram – ažuriranje lokalne baze 16](#_Toc403551975)

[5. Literatura 18](#_Toc403551976)

1. Uvod

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nam cursus. Morbi ut mi. Nullam enim leo, egestas id, condimentum at, laoreet mattis, massa. Sed eleifend nonummy diam. Praesent mauris ante, elementum et, bibendum at, posuere sit amet, nibh. Duis tincidunt lectus quis dui viverra vestibulum. Suspendisse vulputate aliquam dui. Nulla elementum dui ut augue. Aliquam vehicula mi at mauris. Maecenas placerat, nisl at consequat rhoncus, sem nunc gravida justo, quis eleifend arcu velit quis lacus. Morbi magna magna, tincidunt a, mattis non, imperdiet vitae, tellus. Sed odio est, auctor ac, sollicitudin in, consequat vitae, orci. Fusce id felis. Vivamus sollicitudin metus eget eros.

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.

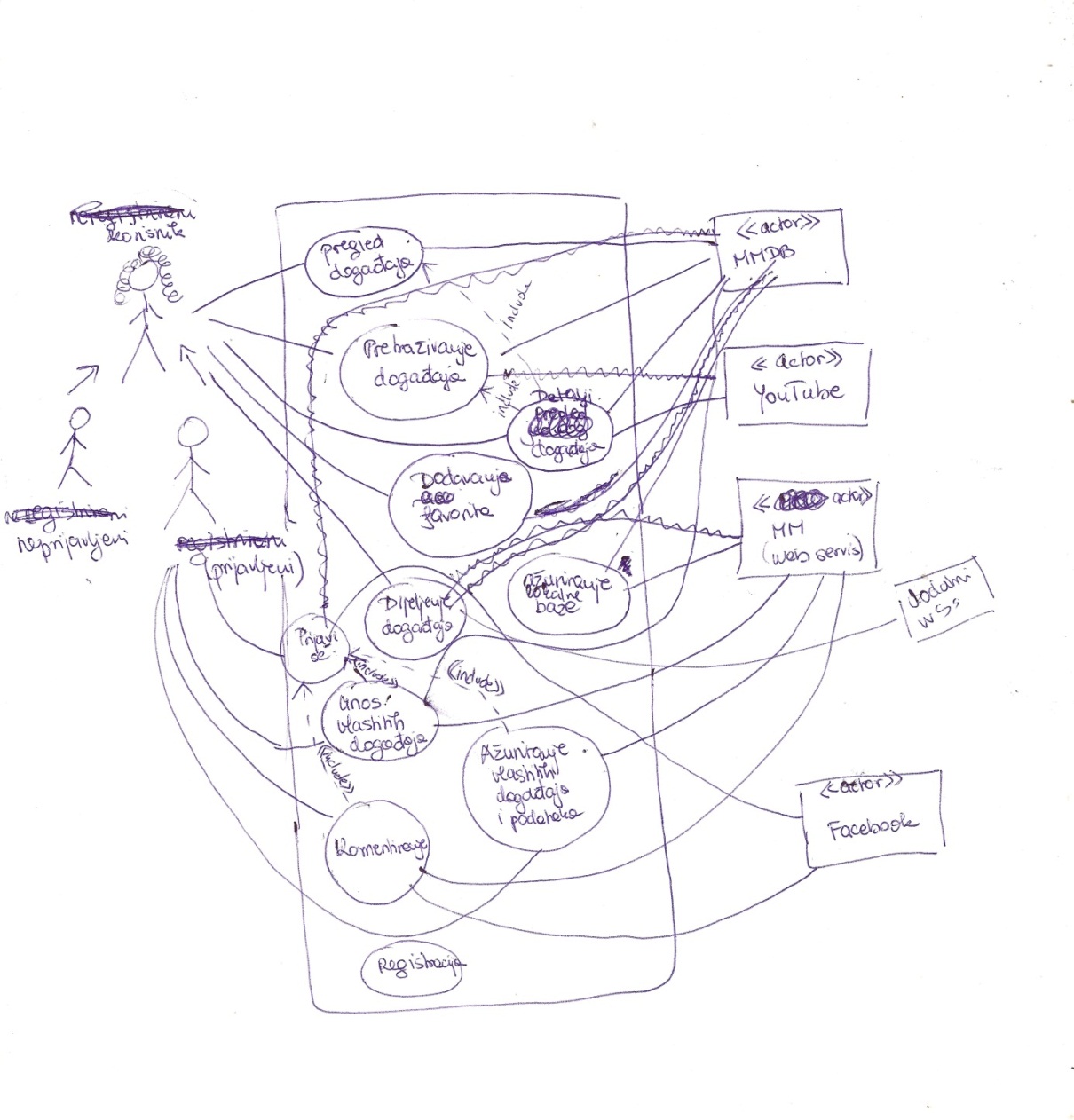
Dnevni SCRUM je 15-minutni dnevni sastanak tima na kojem se razgovara o stvarima koje su obavljene, koje tek trebaju biti obavljanje, sinkroniziraju se aktivnosti I radi plan za iduća 24 sata.

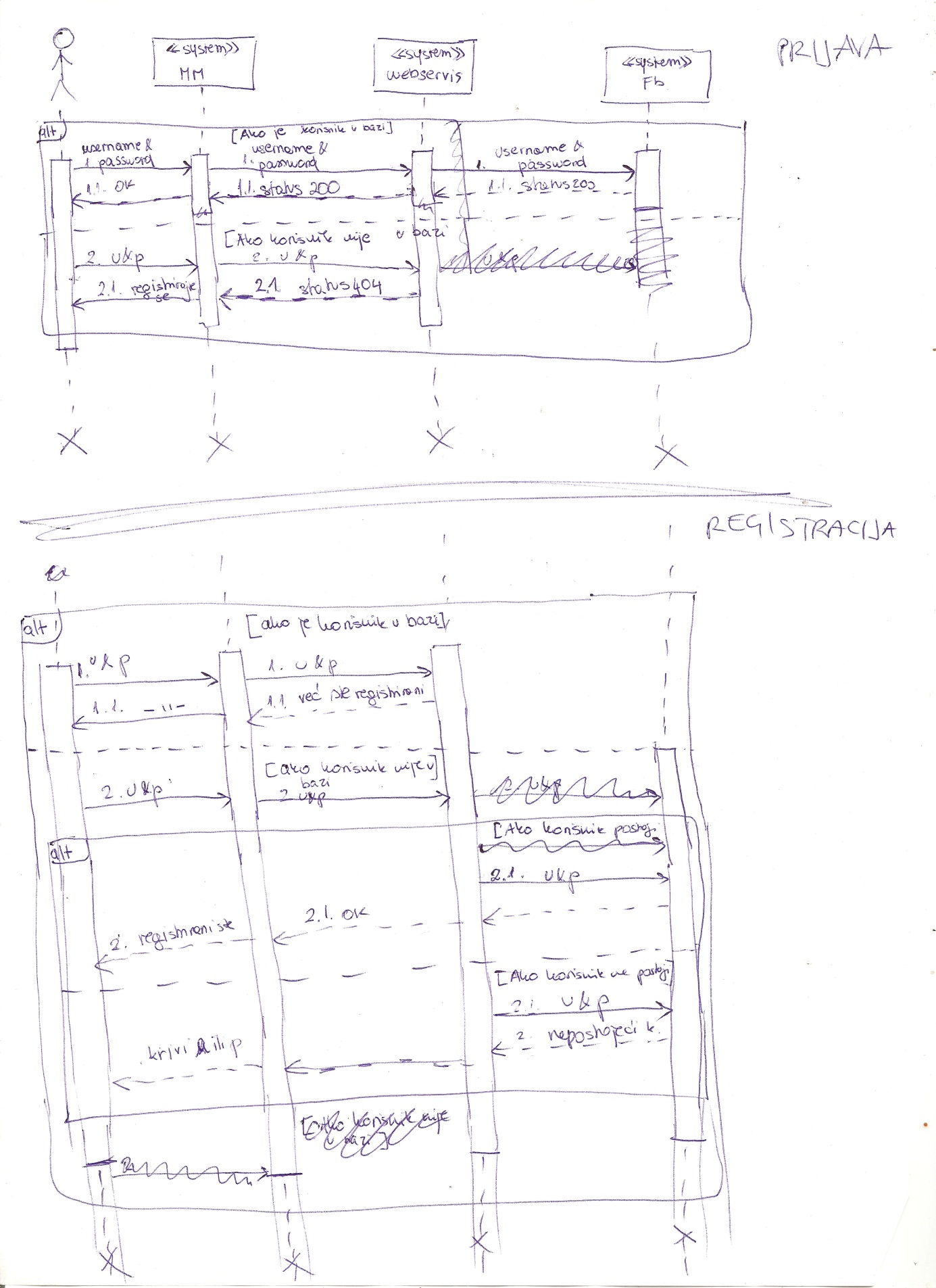
Pregled sprinta dolazi na kraju sprinta, I odnosi se na kontrolu inkrementa I adaptaciju backloga proizvoda ukoliko je to potrebno. Tijekom pregleda, tim I stakeholder-i kolaboriraju obavljen postao u sprintu, te sljedeće što je potrebno obaviti I kako optimizirati vrijednost.

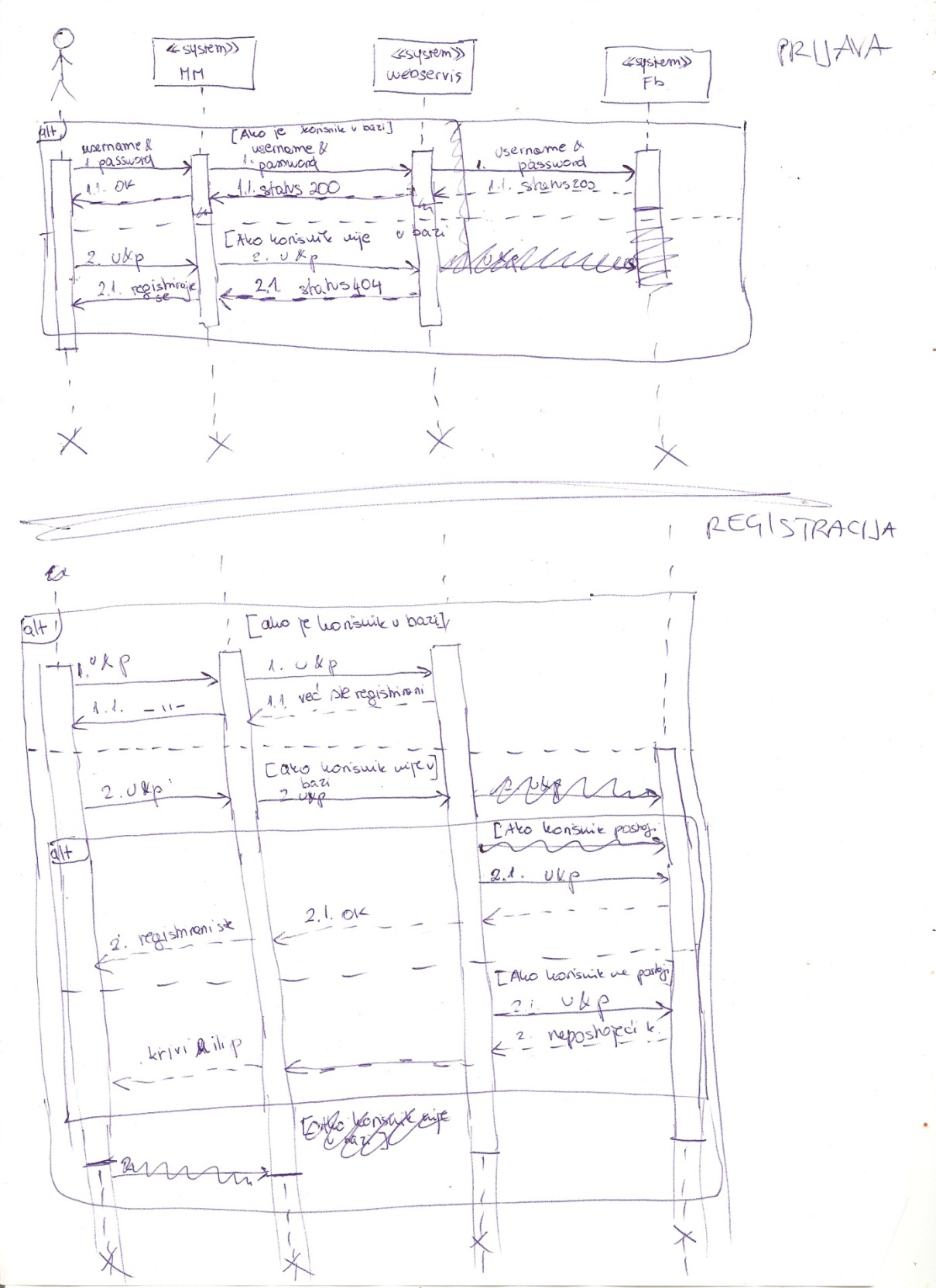
Retrospektiva sprinta je mogućnost za tim da prekontrolira sebe I kreira plan poboljšanja za idući sprint. Dakle, retrospektiva je smještena između pregleda sprinta I planiranja novog sprinta. Artefakti reprezentiraju posao ili vrijednosti koje osiguravaju transparentost ključnih informacija I mogućnosti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

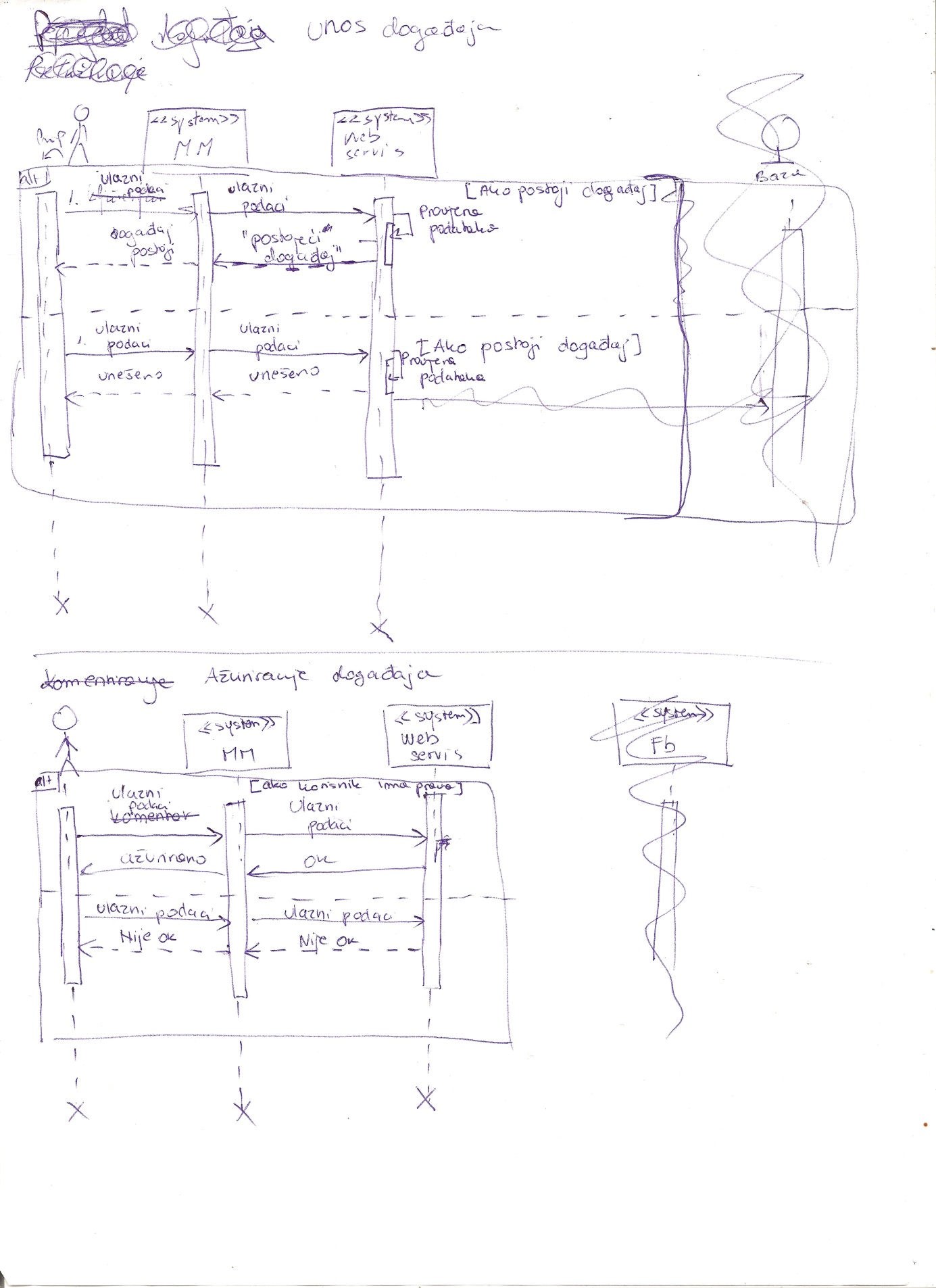
Sprint backlog je skup stavaka backloga proizvoda koje su odabrane za trenutni sprint, I plan za realizaciju novog inkrementa I ciljeva sprinta. Tim modificira sprint backlog kroz sprint.Inkrement je suma svih stavaka backloga proizvoda kompletiranog kroz sprint I vrijednost inkrementa svih prethodnik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

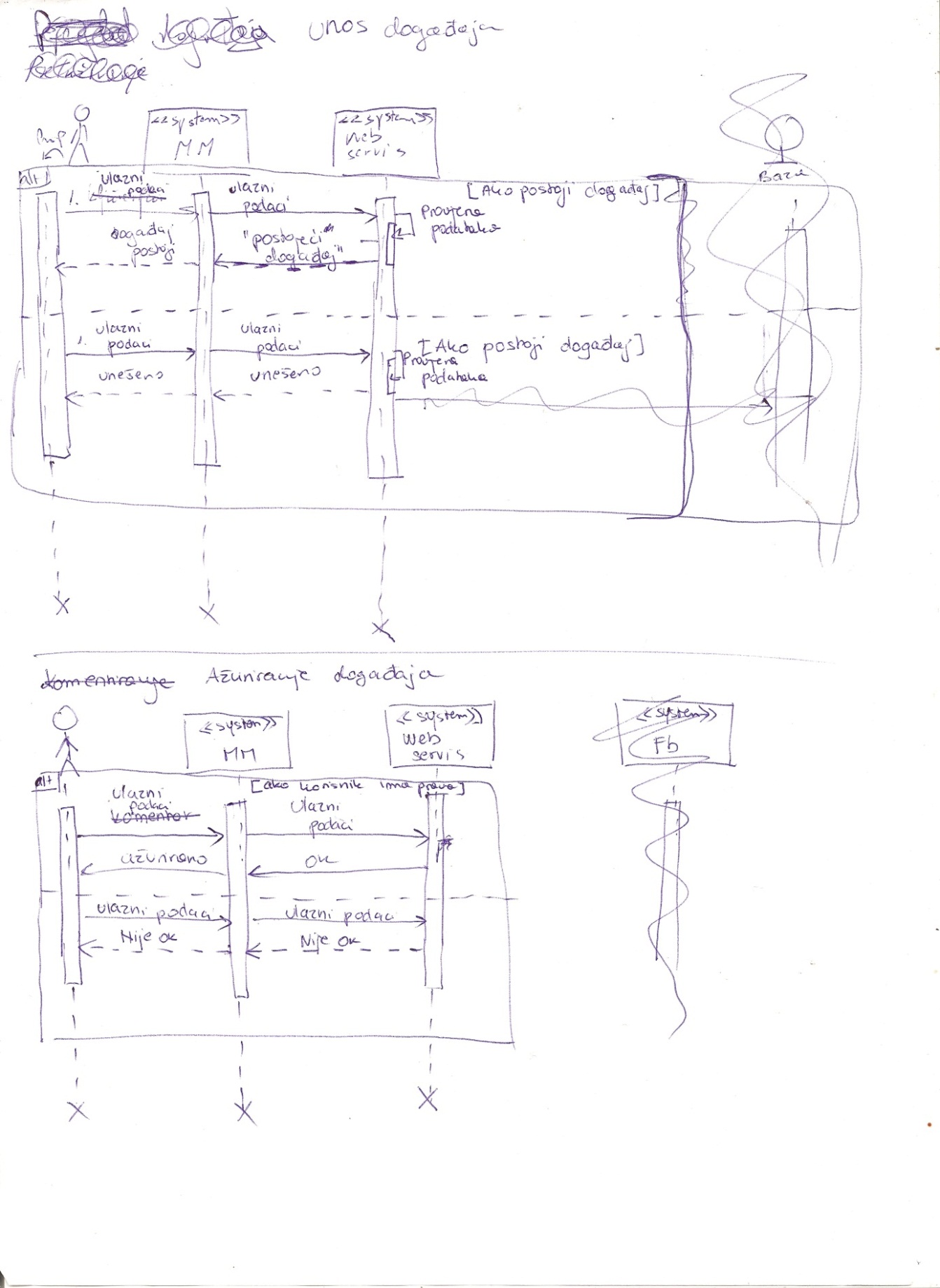
* 1. Artefakti











1. Plan projekta

Tim se sastoji od 3 članova: Matej Bašić, Ivan Šantalab I Ljiljana Pintarić.

Poslove smo podijelili na sljedeći način, uz moguće kasnije preinake:

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

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

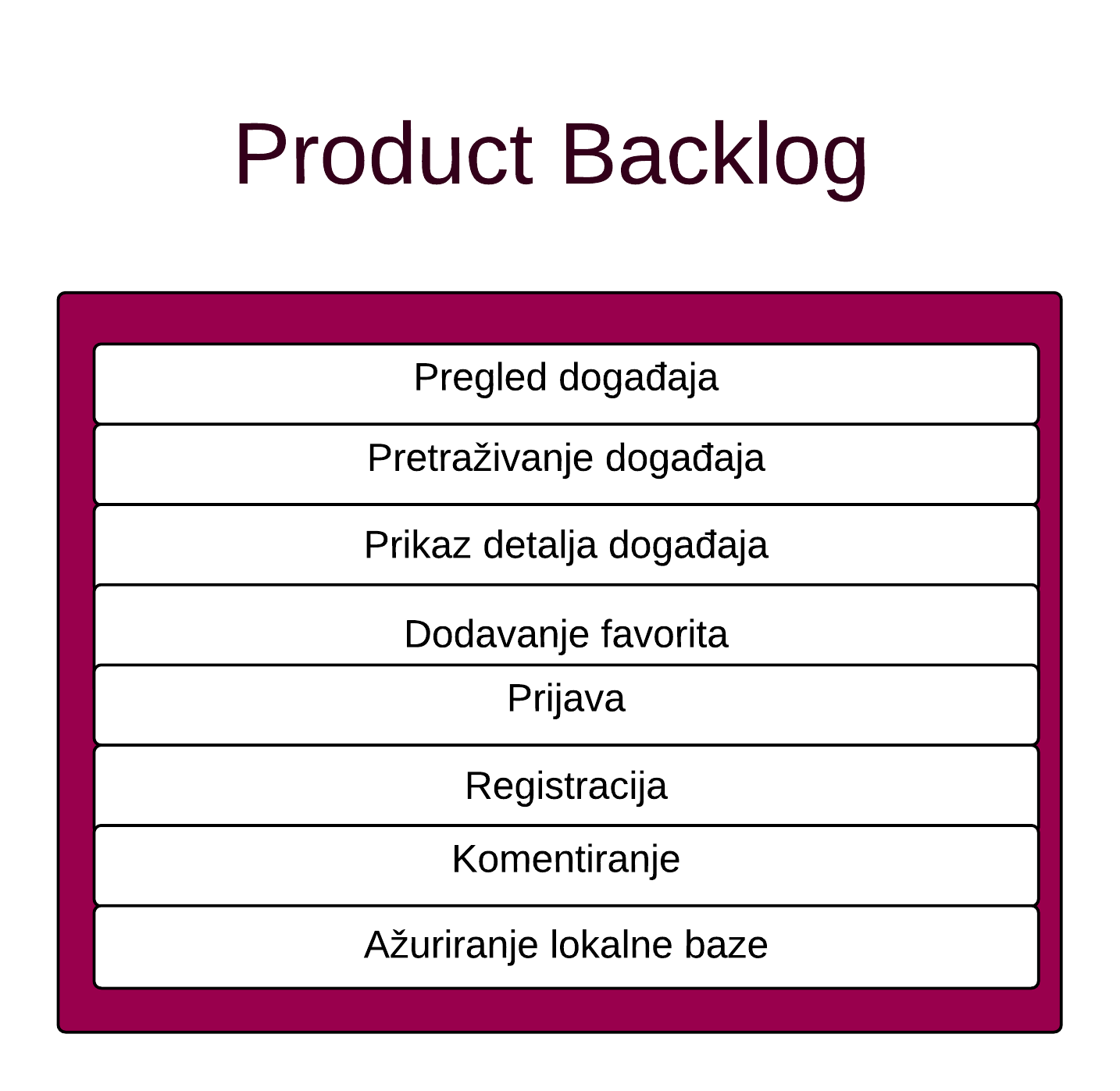
U prvoj fazi ćemo prikupiti I proučiti korisničke zahtjeve, isplanirati rad na razvoju aplikacije, te definirati arhitekturu I dizajn. U drugoj fazi baviti ćemo se izradom dizajna kao I razvojem svih funkcionalnosti aplikacije, dok ćemo u trećoj fazi obaviti testiranje I potrebne popravke, integraciju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

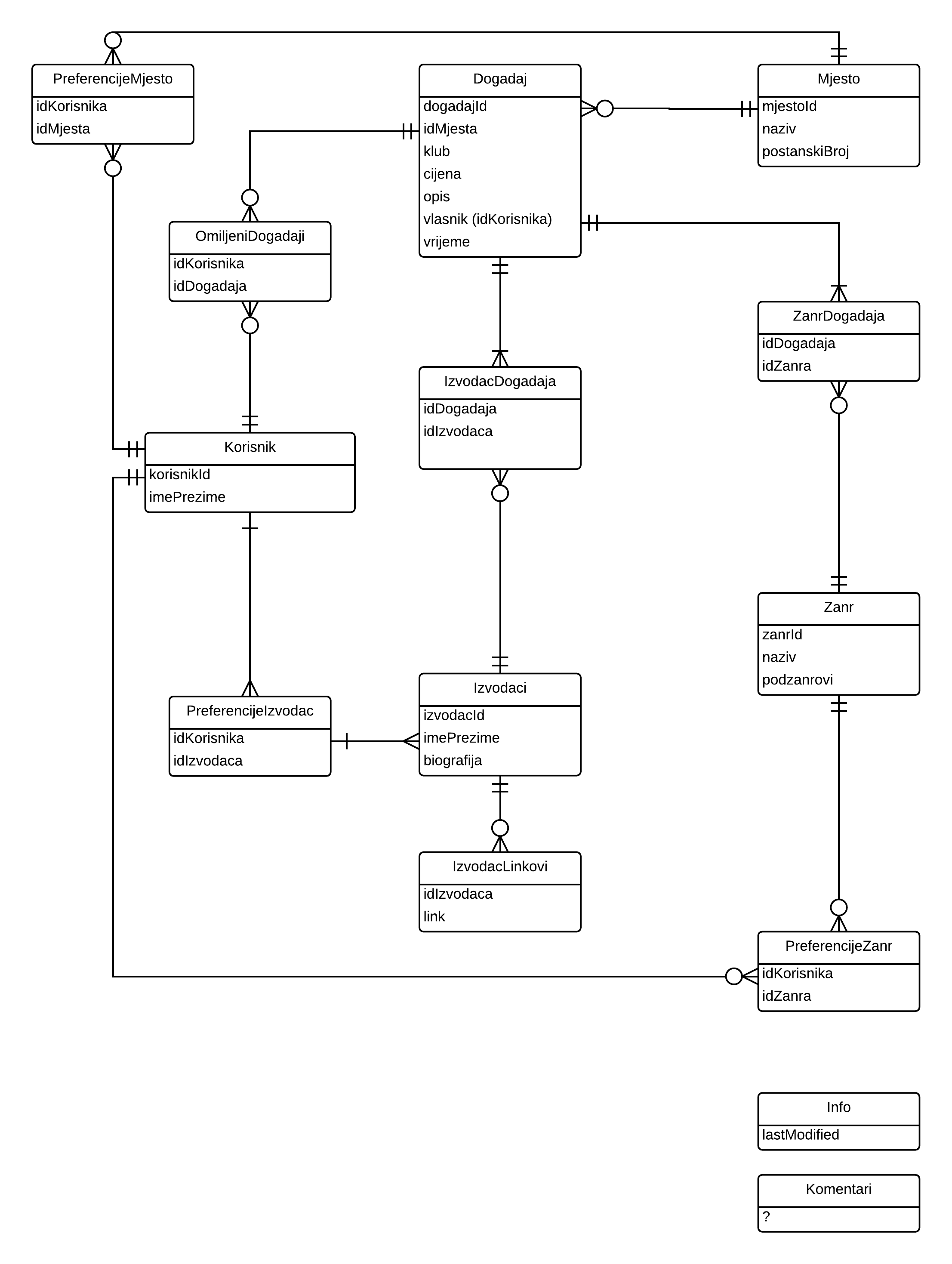
Testiranje

Integracija

Dokumentiranje završnog proizvoda



1. 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

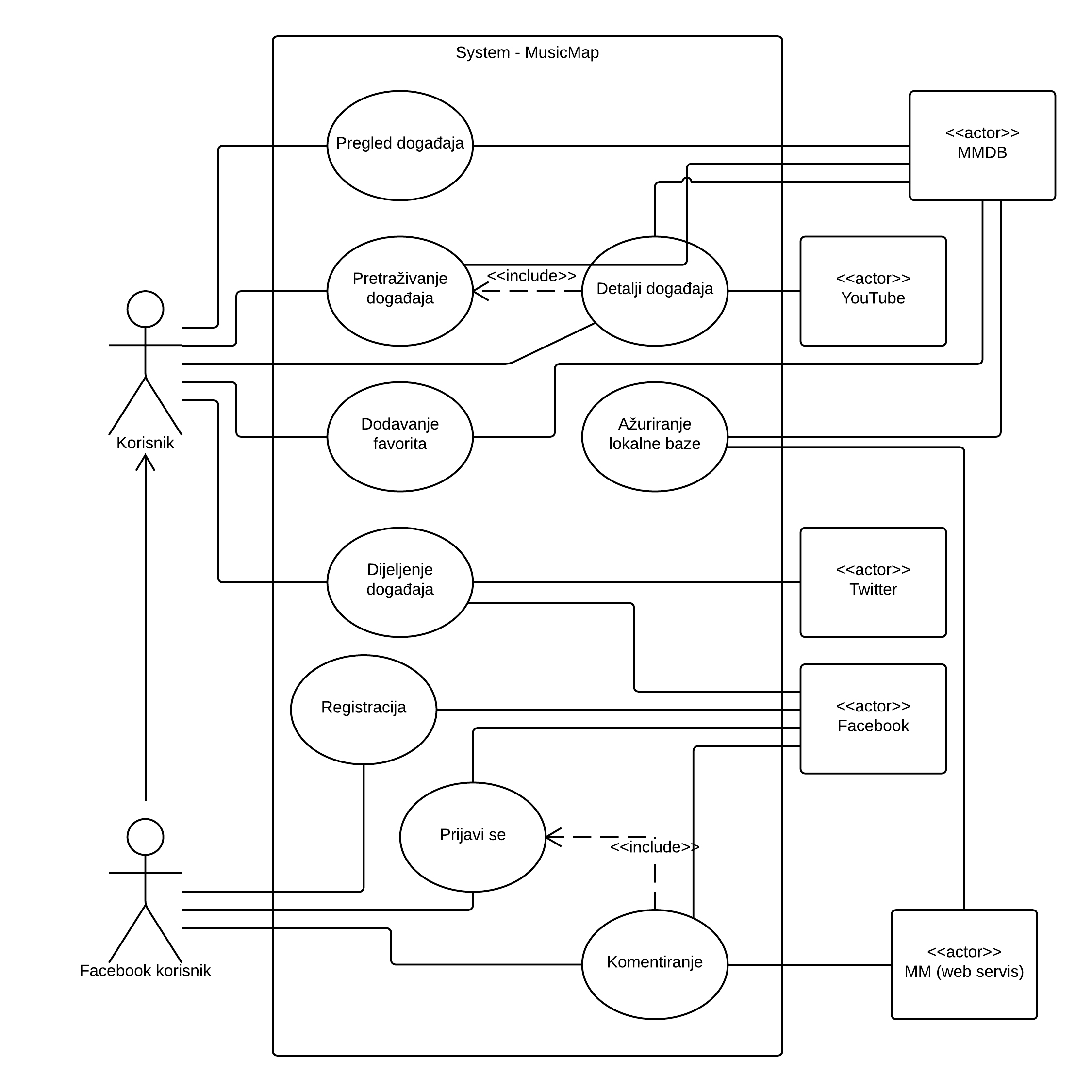
PreferencijeMjesto – korisnikove preferencije vezane uz mjesto izvođenja

PreferencijeŽanr – korisnikove preferencije žanra

Info – sadrži informaciju o posljednjem ažuriranju lokalne baze

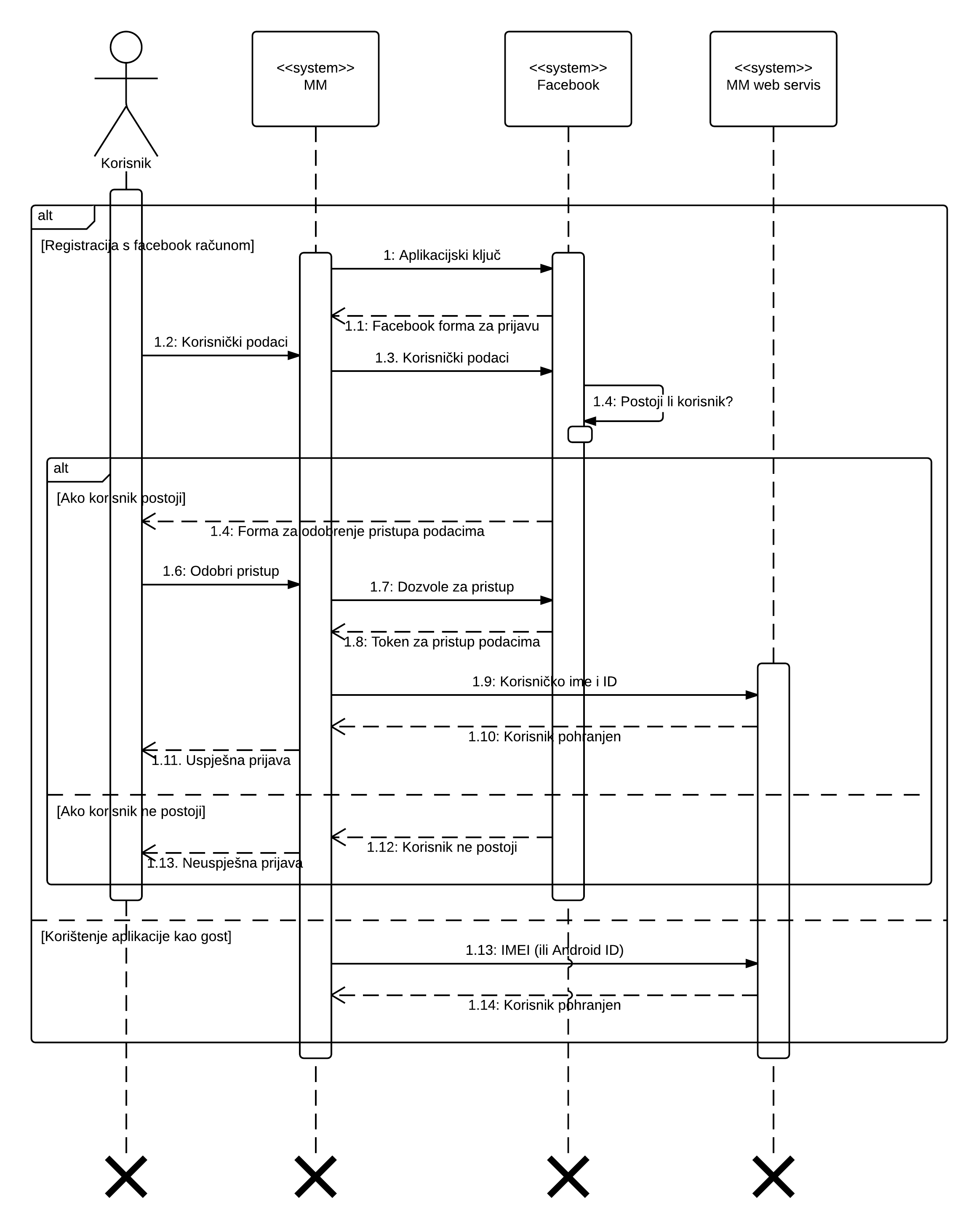
Komentari - ?

* 1. UML
     1. Dijagram slučajeva korištenja



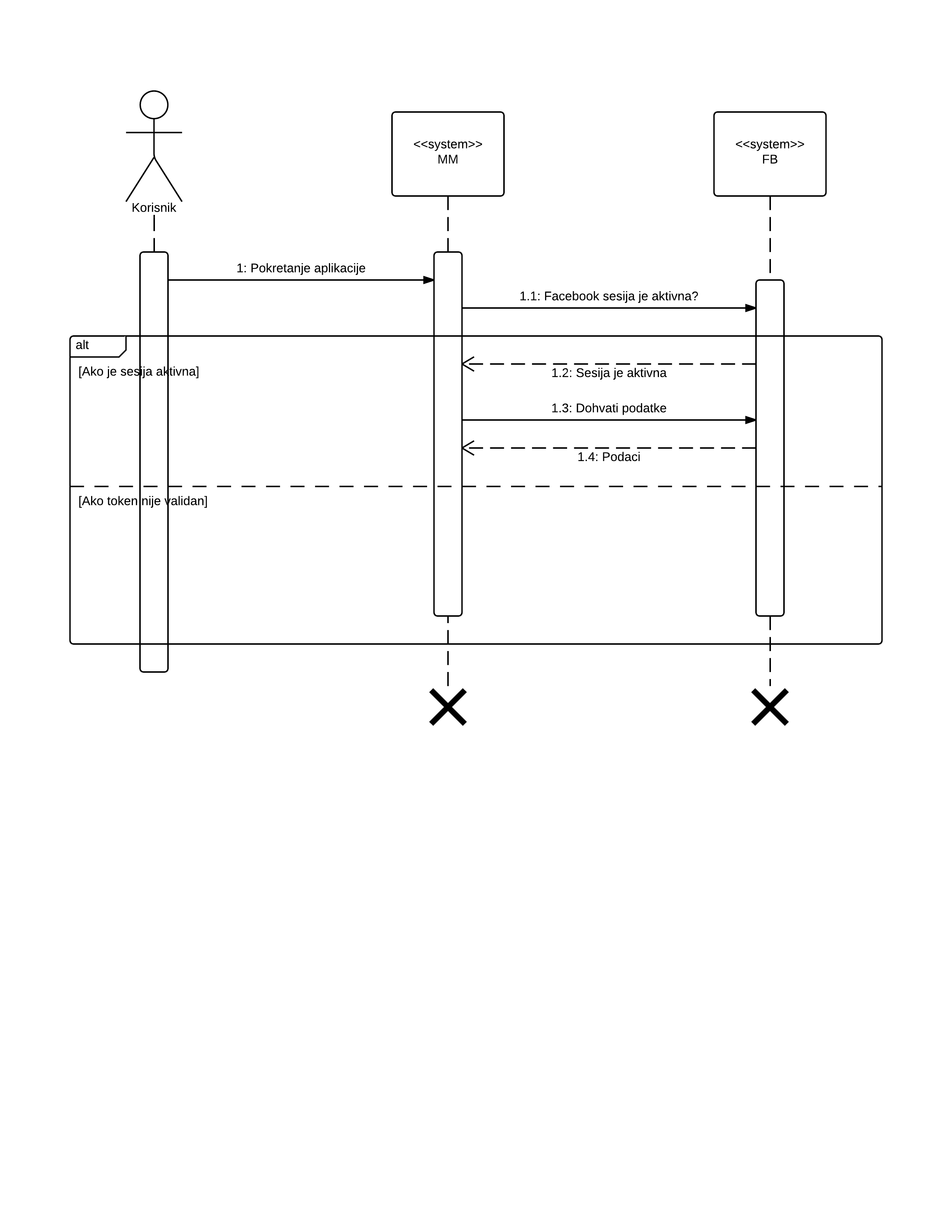
Dijagram slučajeva korištenja prikazuje tipove korisnika, mogućnosti korištenja aplikacije te vanjske servise koji se pritom koriste. Kao što se vidi iz slike, za aplikaciju postoje dva tipa korisnika – običan korisnik i facebook korisnik, pri čemu je facebook korisnik zapravo registrirani/prijavljeni korisnik. Običan korisnik može pregledavati glazbene događaje, i pretraživati ih, gledati detalje događaja, dodavati vlastite favorite te dijeliti događaje na Facebooku i Twitteru. Osim ovih mogućnosti, facebook korisnik se može registrirati i prijaviti korištenjem svojeg facebook računa, te komentirati događaje također objavljujući javno svoje komentare putem Facebooka. Za realizaciju ovih mogućnosti, koristit ćemo, osim navedenih Facebooka i Twittera i druge vanjske servise poput YouTube-a i web servisa 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



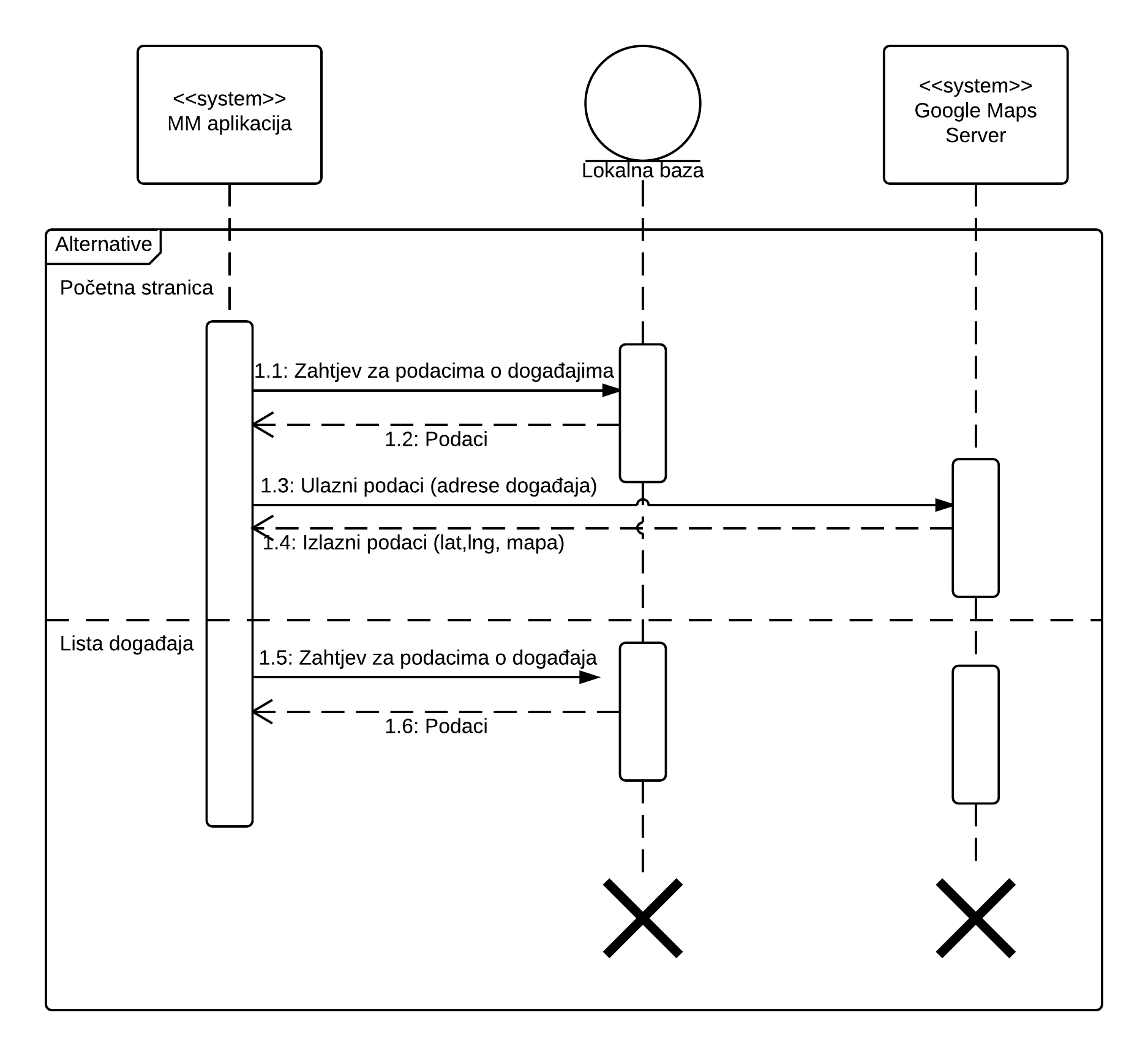
Ukoliko se korisnik želi registrirati, mora imati facebook račun. Prvo upisuje korisničke podatke u formu aplikacije, koja zatim Facebooku šalje aplikacijski ključ, nakon čega se automatski popunjava facebook forma za prijavu, te se na facebook serveru provjerava postoji li korisnik. Ako postoji, korisniku 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 se aplikaciji javlja da je korisnik pohranjen, a 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 ili Android ID, te obavještava aplikaciju da je korisnik pohranjen.

* + 1. Sekvencijalni dijagram – prijava



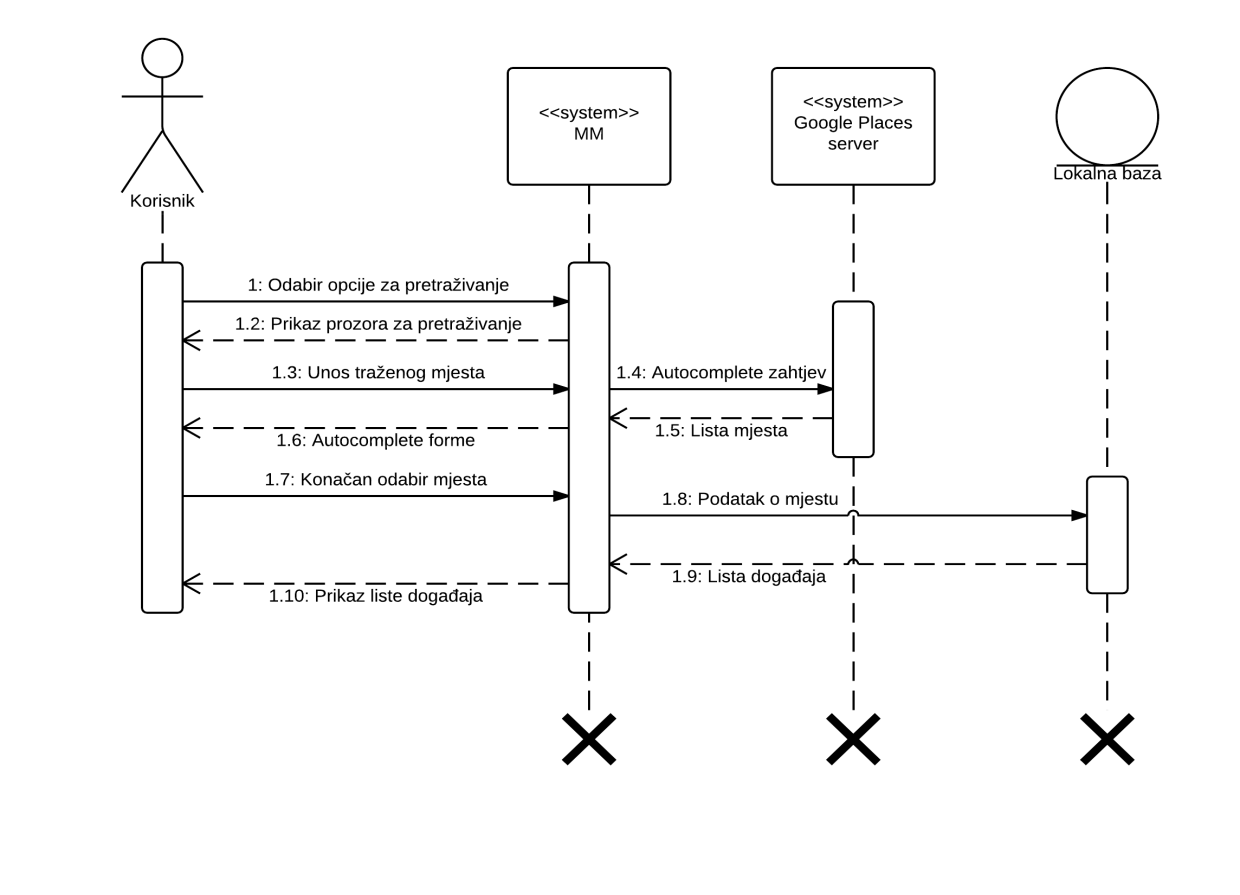
Nakon što je već registriran, korisnik se može prijaviti sa svojim facebook računom, pri čemu se provjerava je li facebook sesija aktivna, te ukoliko je, aplikacija dohvaća podatke s Facebooka.

* + 1. Sekvencijalni dijagram – pregled događaja



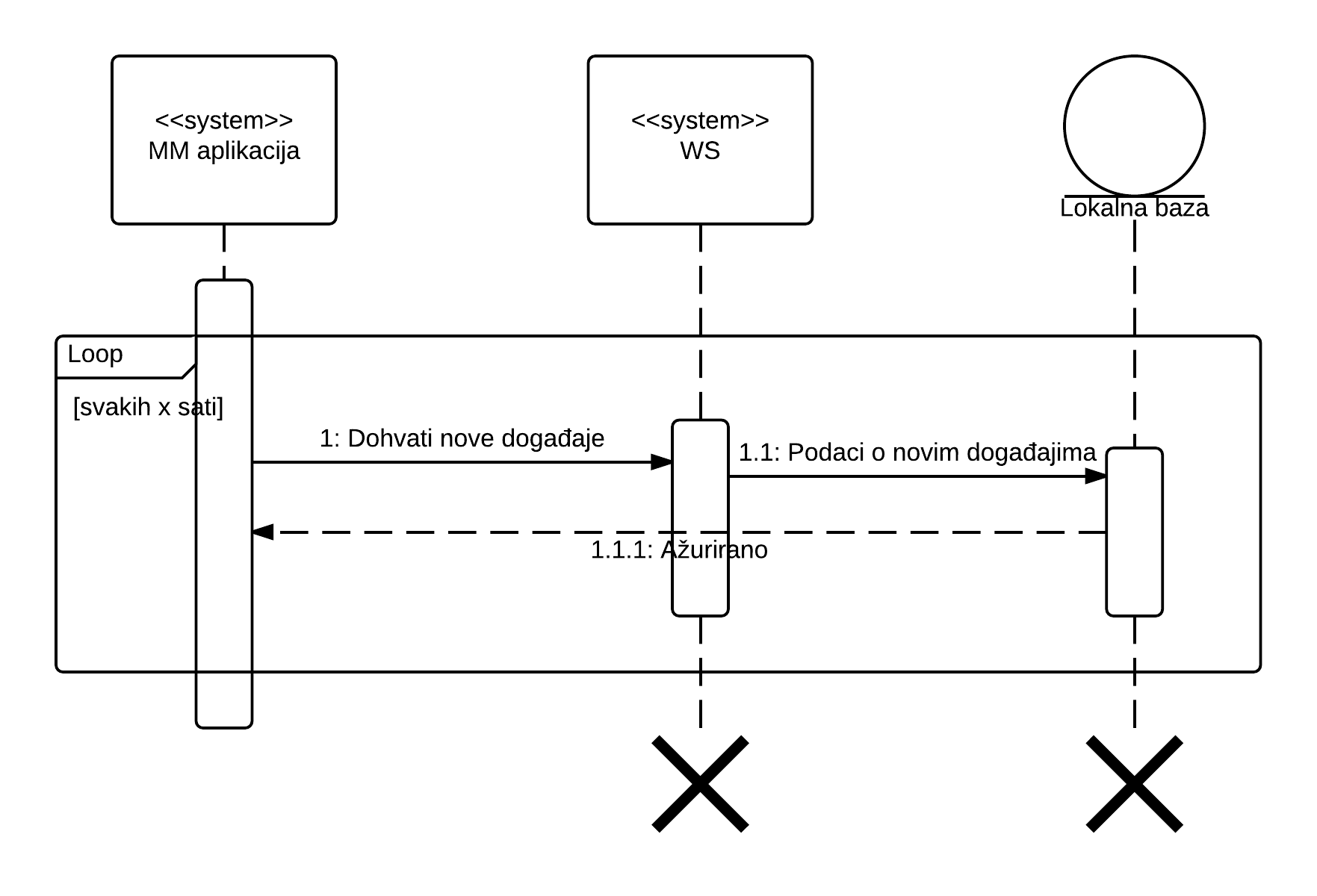
Prilikom pregleda događaja, lokalnoj bazi se šalje zahtjev da prikaže kartu, a da bi to bilo moguće, potrebno ju je dohvatiti sa Google Maps servera, nakon čega se korisniku prikazuje mapa sa markerima. Događaje je također moguće prikazati kao listu događaja, pri čemu se samo učitavaju podaci iz lokalne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

* + 1. Sekvencijalni dijagram – pretraživanje



Korisnik može pretraživati detaljnije događaje prema mjestu događaja. Pritom koristi usluge Google Places servera, šaljući zahtjev za automatsko popunjavanje forme, nakon čega mu se nudi lista mjesta koje može odabrati. Nakon konačnog odabira mjesta, aplikacija pretražuje bazu i prikazuje događaje koji odgovaraju uvjetu pretraživanja.

* + 1. Sekvencijalni dijagram – ažuriranje lokalne baze



Lokalna baza ažurira se svakih nekoliko sati, pri čemu se dohvaćaju novi događaji sa web servera aplikacija.

1. Literatura
2. SCRUM, The SCRUM Guide, <http://www.scrumguides.org/scrum-guide.html>, Dostupno 29.10.2014.