

**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
V A R A Ž D I N**

**Matija Kovaček**

**Suzana Rendulić**

**Ana Sakač**

**Toni Steyskal**

**RENESANSNI FESTIVAL KOPRIVNICA  
PROJEKT IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA**

**Varaždin, 2014.**

**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
V A R A Ž D I N**

**Matija Kovaček, 43550/14-R**

**Suzana Rendulić, 43569/14-R**

**Ana Sakač, 43568/14-R**

**Toni Steyskal, 43573/14-R**

**RENESANSNI FESTIVAL KOPRIVNICA  
PROJEKT IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA**

**Mentor:**

Ivan Švogor, mag. inf., znanstveni  
novak/asistent

**Varaždin, listopad 2014.**

# Sadržaj

<b>1.</b>	<b>UVOD .....</b>	<b>1</b>
<b>2.</b>	<b>ANALIZA TRŽIŠTA .....</b>	<b>2</b>
2.1.	ANALIZA DIREKTNE KONKURENCIJE.....	2
2.2.	ANALIZA INDIREKTNE KONKURENCIJE.....	2
2.3.	POTENCIJALNI KORISNICI.....	3
2.4.	ZAKLJUČAK ANALIZE TRŽIŠTA.....	4
<b>3.</b>	<b>PROJEKTNI PLAN .....</b>	<b>5</b>
3.1.	METODOLOGIJA.....	5
3.2.	SPRINT SASTANCI .....	6
3.3.	DEFINIRANJE TIMA .....	8
3.4.	ARHITEKTURA.....	11
<b>4.</b>	<b>KORISNIČKI ZAHTJEVI .....</b>	<b>13</b>
4.1.	NEFUNKCIONALNI ZAHTJEVI.....	15
<b>5.</b>	<b>PRODUCT BACKLOG .....</b>	<b>16</b>
<b>6.</b>	<b>SPRINT1 BACKLOG .....</b>	<b>18</b>
6.1.	BURNDOWN CHART .....	20
<b>7.</b>	<b>SPRINT2 BACKLOG .....</b>	<b>21</b>
7.1.	BURNDOWN CHART .....	22
7.2.	MODEL PODATAKA .....	23
7.3.	DIJAGRAM KLASA .....	25
<b>8.</b>	<b>SPRINT3 BACKLOG .....</b>	<b>35</b>
8.1.	BURNDOWN CHART .....	36
<b>9.</b>	<b>PRORAČUN PROJEKTA.....</b>	<b>37</b>
9.1.	PONUĐA KUPCU.....	39
<b>10.</b>	<b>DIZAJN APLIKACIJE .....</b>	<b>40</b>

# 1. Uvod

U svrhu projektnog zadatka iz kolegija Analiza i razvoj programa bit će izrađena aplikacija za Renesansi Festival Koprivnice. Renesansni Festival predstavlja povijesni spektakl koji svojom dugom tradicijom i baštinskim vrijednostima okuplja velik broj posjetitelja koji raste iz godine u godinu. Sav sadržaj tematski je vezan uz razdoblje Renesanse iz 15. i 16. stoljeća. Svoje radove predstavljaju brojne viteške, obrtničke i muzičke skupine od kojih su najpoznatiji Koprivnički mušketiri i haramije, vitezovi, srednjovjekovni obrtnici i kuhari te gutači vatre koji predstavljaju jedne od najzanimljivijih atrakcija. Već tradicionalno, festival počinje u središtu grada u jutarnjim satima od kuda povorka vitezova kreće do glavne lokacije – renesansnih bedema koji su krcati srednjovjekovnim logorima, šatorima, kulama i raznim zabavljačima. Da bi ugođaj bio potpun, posjetitelji mogu uživati i u renesansnoj hrani koja se priprema na licu mjesta po izvornim receptima iz davnih vremena. Uz hranu, dostupna su i tradicionalna pića iz domaćih vinograda i staro hrvatsko piće medovina. Osobito je popularna bitka vitezova tj. večernji napad na grad koji obasjan bakljama i žaračima predstavlja nezaboravno iskustvo.

RenFest je mobilna aplikacija koja omogućava korisnicima da saznaju korisne informacije o Renesansnom festivalu u Koprivnici. Pomoću aplikacije korisnici bi mogli pregledavati programe samog festivala po danima te bi znali točno vrijeme izvođenja pojedinog događaja i njegove pojedinosti. Korisnici koji prvi puta posjećuju festival mogli bi pomoću karte saznati gdje se točno festival održava, a korisnici koji odluče ostati u gradu za vrijeme održavanja festivala mogu također vidjeti korisne lokacije i ostale osnovne informacije o hotelima i restoranima. Korisnicima iz inozemstva sadržaj aplikacije bit će dostupan na engleskom jeziku.

## **2. Analiza tržišta**

Kako bi procijenili uspjeh i prije svega utvrdili potrebu za izradom aplikacije za Renesansni festival proveli smo analizu tržišta unutar koje smo obuhvatili analizu direktne konkurencije gdje spadaju aplikacije sa istom tematikom kakva je naša (ako postoje), analizu indirektna konkurencije koja uključuje aplikacije za festivale, ali ne s temom Renesansnog festivala već s temom ostalih filmskih/glazbenih festivala, te analizu potencijalnih korisnika.

### **2.1. Analiza direktne konkurencije**

Upoznajući se sa tematikom i mogućnostima koje nudi Renesansni festival uočili smo priliku za kreiranje naše aplikacije RenFest. Mobilna aplikacija za navedeni događaj ne postoji tako da direktne konkurencije koja bi ugrozila naš proizvod nema. Web stranica Renesansnog festivala je ažurna, ali loše strukturirana i neprilagođena za mobilne uređaje. Njezine funkcionalnosti testirane su na nekoliko mobilnih uređaja u vlasništvu članova tima gdje je zaključeno kako se između ostalog ne može pristupiti rasporedu festivala i karti gdje je prikazana lokacija što smatramo jednim od glavnih funkcionalnosti koje bi obavezno morale biti dostupne.

### **2.2. Analiza indirektna konkurencije**

Analizu indirektna konkurencije možemo napraviti kako bi prepoznali funkcionalnosti koje se najčešće nalaze u aplikacijama sa tematikom festivala i kako bi se svojim dodatnim funkcionalnostima diferencirali od njih. Za analizu aplikacije čije je područje Republika Hrvatska uzeta je aplikacija za INmusic festival. Na Google Store-u ima visoke ocjene bez većih prigovora osim za to da se aplikacija „ruši“ na određenim uređajima. Zbog navedenih prigovora možemo zamijetiti da veliku pažnju treba posvetiti prilagodbi aplikacije različitim uređajima, osobito različitim rezolucijama ekrana kako bi u svim slučajevima aplikacija vizualno dobro izgledala i kako se ne bi onemogućio pristup pojedinim dijelovima aplikacije. Nadalje, mogućnosti koje INmusic aplikacija nudi su prikaz programa festivala, mape festivala kako bih se što lakše snalazili te informacije o festivalu i izvođačima. Analizom ostalih aplikacija na istom području također je potvrđeno korištenje navedenih funkcionalnosti koje se stavljaju u prvi plan kao obavezne funkcionalnosti koje i naša aplikacija mora imati. Navedene funkcionalnosti su i predložene od članova tima na sastancima koji su prethodili analizi tržišta.

S obzirom da su dobar dio posjetitelja Renesansnog festivala stranci, proučeno je i tržište festivalskih aplikacija izvan granica države. Uočena je velika popularnost i prihvaćenost ovakvog tipa aplikacija pogotovo za vrlo poznate festivale kao što su Cannes film festival i glazbeni festival Sziget te je zaključeno kako je potrebna funkcionalnost prevođenja aplikacije na engleski jezik kako bi ona bila pristupačna i stranim posjetiteljima. Također je vrlo bitna modularnost aplikacije kako bi bez problema mogli biti u toku sa dodatnim popularnim funkcionalnostima s kojima možemo nadograđivati našu aplikaciju.

### **2.3. Potencijalni korisnici**

Posjećenost Renesansnog festivala 2014. godine procjenjuje se na 50.000 ljudi čime je nadmašena posjećenost popularnih festivala u Hrvatskoj kao što su varaždinski Špancirfest i karlovački Dani piva u čemu smo vidjeli priliku i veliki potencijal ovog tržišta. Naša uvjerenja dodatno su potvrdili prigovori organizatora Renesansnog festivala o njegovoj slaboj medijskoj eksponiranosti. Smatramo kako bi mobilnom aplikacijom povećali zainteresiranost za navedeni događaj i privukli posjetitelje jer na taj način Renesansni festival ostavlja dojam dobre organizacije i pristupačnosti. To je također jedan od načina privlačenja mladih posjetitelja gdje im na „moderan“ način predstavljamo duh nekog starog vremena. Zbog velike posjećenosti 2014. i pozitivnih reakcija posjetitelja očekuje se porast popularnosti sljedeće 2015. godine što možemo istaknuti kao veliku prednost s obzirom da je tada planirano plasiranje naše aplikacije na tržište. Također, potencijalne korisnike predstavljaju i dosadašnji posjetitelji web stranice Renesansnog festivala. Njezinu lošu strukturu možemo uzeti kao priliku da kreiramo aplikaciju koja će biti pregledna i dobro strukturirana tako da će posjetitelji više posezati za našom aplikacijom nego web stranicom.

## 2.4. Zaključak analize tržišta

Osnovne činjenice koje su definirane ovim analizom:

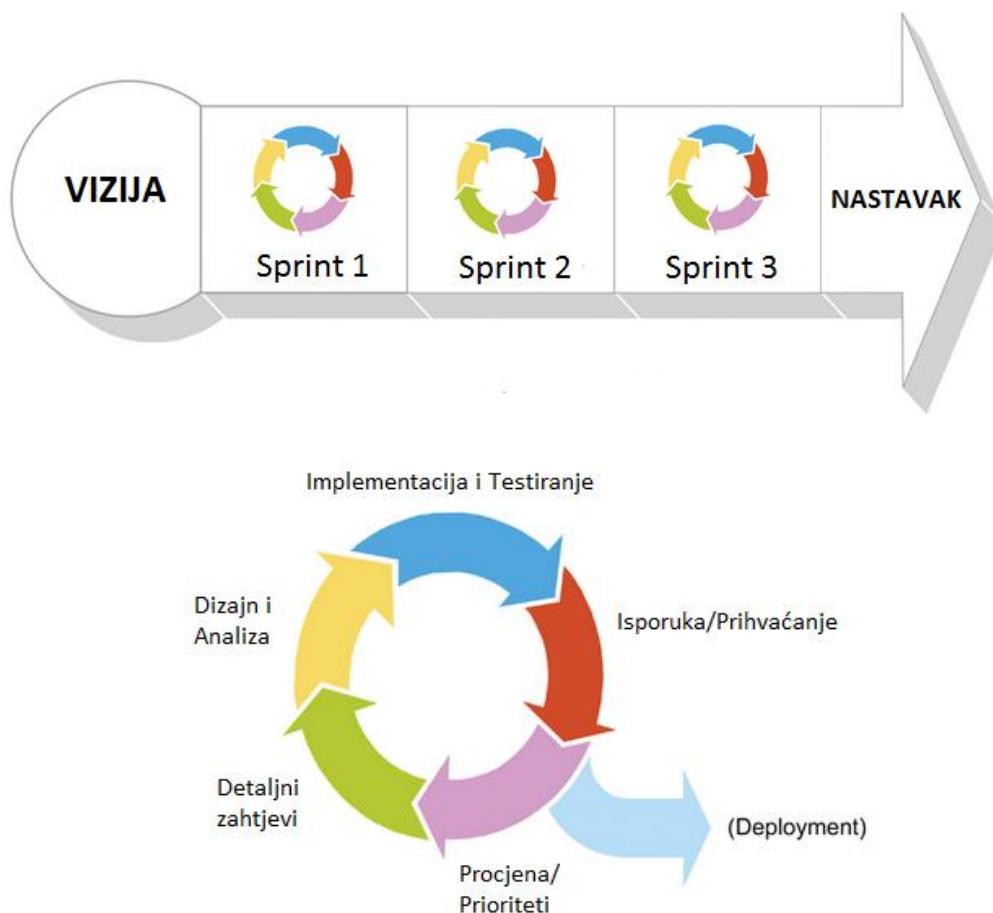
- direktna konkurencija ne postoji
- indirektna konkurencija ne predstavlja prijetnju
- utvrđena je potreba za:
  - prilagodbom aplikacije različitim uređajima
  - uvrštavanjem glavnog programa festivala, karte lokacije i informacija o Renesansnom festivalu u aplikaciju
  - prijevodom sadržaja aplikacije na engleski jezik
- potencijal tržišta je velik, kao i broj potencijalnih korisnika aplikacije
- potrebna je dodatna promocija Renesansnog festivala što će ova aplikacija jedim dijelom i omogućiti

Utvrđeno je kako postoji potreba za ovakvim tipom aplikacije i okvirno su definirani korisnički zahtjevi.

### 3. Projektni plan

#### 3.1. Metodologija

Za metodiku izrade naše aplikacije odabrali smo Scrum agilnu metodiku razvoja programskog proizvoda. U sklopu našeg razvoja proizvoda - aplikacije RenFest, odlučili smo se na tri iteracije, odnosno Sprint-a. Unutar svakog Sprint-a imamo definirano što će se obaviti, odnosno koje su nam glavne cjeline rada, na koji način će se rad obaviti te kako će na kraju izgledati konačan proizvod (slika 1).



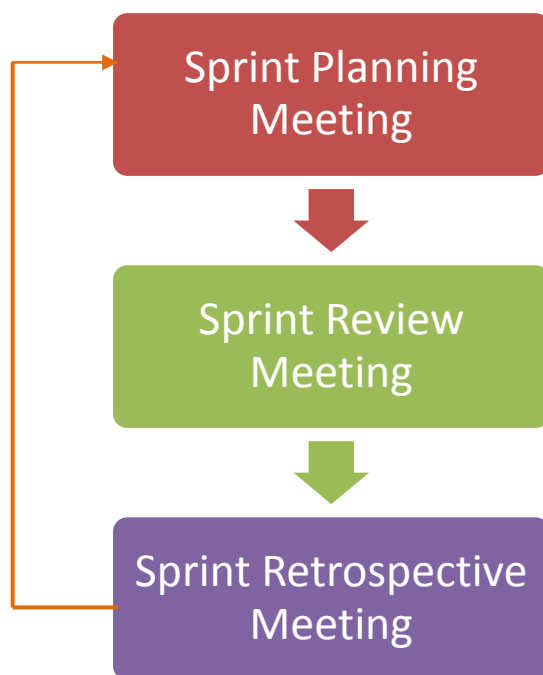
**Slika 1:** Detalji iteracije

Kako nam je cilj zadovoljiti tržište našim programskim proizvodom, dodatna je prednost što Sprint omogućuje lakše prilagođavanje novim i promijenjenim zahtjevima korisnika koji su neizbježni. Prošlogodišnja posjećenost Renesansnog festivala dokaz nam je kako postoji potreba za ovakvom vrstom proizvodima gdje ćemo, u skladu s potrebama, obuhvatiti sve zahtjeve koji se nameću kao prioritet. Nakon povratnih informacija od strane korisnika slijedi nadzor i prilagodba novim primijećenim zahtjevima.



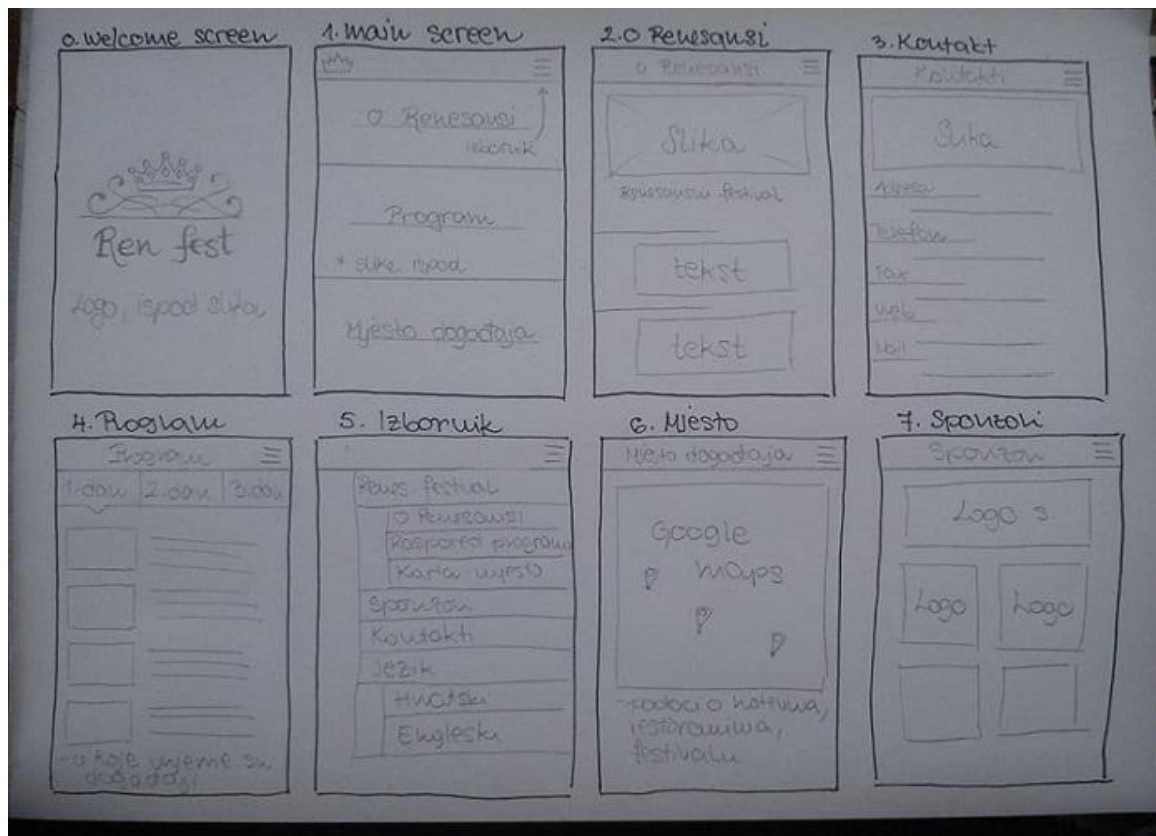
### 3.2. Sprint sastanci

Naš prvi Sprint predstavlja dvotjedni Sprint, sa dvosatnim sastancima planiranja sprinta. Naš tijek Scrum planiranja prikazan je na slici 3.



**Graf 1:** *Tijek Scrum-a*

Sastanak za Sprint Planning održan je 20. listopada 2014. godine i trajao je dva sata. Scrum Team je prošao kroz zahtjeve i specifikacije korisnika, te stvorio opću ideju realizacije projekta. Prvobitni dizajn aplikacije je skiciran, te je učinjena određena podjela funkcionalnosti ovisno o dijelu aplikacije na kojem se nalazi korisnik (slika 2 i 3). Cilj Sprint-a je ono što će biti zadovoljeno unutar Sprint-a kroz implementaciju Product Backlog-a, a što je moguće vidjeti na slici 3. Drugi dio sastanka odraden je 25. listopada i trajao je sat vremena. Scrum Team je među sobom napravio dekompoziciju stavaka preuzetih u Sprint Backlog-u, na radne zadatke, odnosno aktivnosti.



Slika 2: Zamisao izgleda aplikacije



Slika 3: Funkcionalnosti aplikacije

Dnevni Scrum karakterizira vremenski ograničen sastanak na maksimalno 15 minuta koji se organizira isključivo u krugu članova tima. Iako se zove dnevni i trebao bi se provoditi svaki dan u isto vrijeme, često to nije slučaj. Takav slučaj vrijedio je i za nas što je razlog da Graf 1 ne prikazuje dnevni Scrum kao tijek izvršavanja. Iako je svatko imao zadane svoje aktivnosti rada, nismo provodili dnevni Scrum kako bi provjerili tko je što napravio od zadnjeg sastanka, odnosno tko kako napreduje. Pripisujemo to spriječenosti zbog drugih obaveza, no zato smo na kraju svakog tjedna, kao Scrum Team, provjerili naše napredovanje, predložili dodatne sugestije, način rada i dalje nastavili sa radom do kraja sljedećeg tjedna gdje bi ponovili isti postupak. Zbog toga takve sastanke bilježimo kao revizija Sprint-a umjesto dnevnog Scrum sastanka.

Retrospektivu Sprint-a tek očekujemo što znači da ćemo saznati koja su dodatna poboljšanja i unapređenja potrebna, što je bilo dobro, a što loše u Sprint-u te kako se Sprint čini kao zaokružena cjelina. Odgovori na takva pitanja imaju za cilj isključivo unaprjeđenje i prilagodbu rada kako bi sljedeći Sprint bio bolji i efikasniji.

### **3.3. Definiranje tima**

Scrum tim se sastoji od vlasnika proizvoda, razvojnog tima, i tzv. Scrum Master-a. Vlasnik proizvoda je u našem slučaju naručitelj, odnosno Turistička zajednica grada Koprivnice. Kako bi projekt izrade mobilne aplikacije bio uspješno završen, morali smo sastaviti razvojni tim. On se sastoji od četiri člana:

- Matija Kovaček
- Suzana Rendulić
- Ana Sakač
- Toni Steyskal

Scrum Master i vlasnik proizvoda mogu također biti članovi razvojnog tima. Naš tim se sastoji od četiri člana čije smo uloge prikazali u nastavku. No, prije nego prikazemo uloge i aktivnosti članova tima, razjasnit ćemo ukratko tri ključne uloge Scrum tima:

1. Vlasnik proizvoda - zapravo je naručitelj i odgovoran je za maksimizaciju vrijednosti proizvoda i rada razvojnog tima. Vlasnik proizvoda je jedini odgovoran za upravljanje Product Backlog-om gdje samo on određuje raspored stavaka.
2. Razvojni tim - grupa ljudi (od 3 do 9) koja je zadužena za razvoj programskog proizvoda. Tim sam bira koje stavke iz Product Backlog-a mogu završiti u jednom Sprint-u i kako će iste napraviti. U razvojnom timu ne postoje uloge već članovi tima surađuju, sami se organiziraju te su svi jednako odgovorni za sve.

3. Scrum Master - zapravo je voditelj i odgovoran je za otklanjanje svih prepreka kako bi grupa na vrijeme ispunila ciljeve svakog Sprint-a. Scrum Master ima ulogu prvenstveno kontrolirati proces, odnosno osigurati da se Scrum tim pridržava teorije, prakse i pravila Scrum-a.

Uloge i aktivnosti članova tima:

Matija Kovaček

- uloga: Scrum Master, sposoban voditi tim, sprječava da dođe do pogrešaka te uklanja prepreke ukoliko tim naiđe na njih
- organizira sastanke tima te osigurava sve resurse i povoljne uvjete za rad
- sudjeluje na sastancima tima
- sudjeluje u osmišljavanju i izradi Sprint Backlog liste
- sudjeluje u modeliranju i implementaciji aplikacije
- sudjeluje u kreiranju i ažuriranju dokumentacije

Suzana Rendulić

- uloga: Razvojni tim
- sudjeluje na svim sastancima tima
- sudjeluje u modeliranju i implementaciji aplikacije
- sudjeluje u kreiranju i ažuriranju dokumentacije






































Toni Steyskal

- uloga: Razvojni tim
- zadužen za osmišljavanje i izradu Product Backlog liste
- sudjeluje na svim sastancima tima
- sudjeluje u modeliranju i implementaciji aplikacije
- sudjeluje u kreiranju i ažuriranju dokumentacije

Ana Sakač

- uloga: Razvojni tim
- sudjeluje na svim sastancima tima
- sudjeluje u modeliranju i implementaciji aplikacije
- sudjeluje u kreiranju i ažuriranju dokumentacije

Tijek rada i aktivnosti možete vidjeti na slici

		Task Mode	Task Name	Duration	Start	Finish	Pr	Resource Names	Cost
1			Projektan plan - RenFest	41 days	Tue 28.10.14	Wed 24.12.14			2.557,00 kn
2			POČETAK PROJEKTA	0 days	Tue 28.10.14	Tue 28.10.14			0,00 kn
3			SPRINT 1	13 days	Tue 28.10.14	Thu 13.11.14	2		2.557,00 kn
4			Implementacija vizualnog djela početnog ekrana	2 hrs	Tue 28.10.14	Tue 28.10.14	2	Suzana Rendulić Laptop[1]; Matija Kovaček	152,00 kn
5			Implementacija logike početnog ekrana	3 hrs	Wed 29.10.14	Wed 29.10.14	4	Matija Kovaček; PC računalo[1]	113,00 kn
6			Implementacija vizualnog djela izbornika	2 hrs	Fri 31.10.14	Fri 31.10.14	5	Laptop[1]; PC računalo[1]; Suzana Rendulić	160,00 kn
7			Implementacija logike izbornika	2 hrs	Tue 4.11.14	Tue 4.11.14	6	PC računalo[1]; Toni Steyskal	78,00 kn
8			Implementacija vizualnog djela ekrana za prikaz osnovnih informacija	3 hrs	Tue 4.11.14	Tue 4.11.14	7	PC računalo[1]; Toni Steyskal	113,00 kn
9			Implementacija logike prikaza osnovnih informacija	2 hrs	Thu 6.11.14	Thu 6.11.14	8	Ana Sakač; Laptop[1]; PC računalo[1];T	160,00 kn
10			Vizualna implementacija kontakt informacija	2 hrs	Fri 7.11.14	Fri 7.11.14	9	Ana Sakač; Laptop[1]; Toni Steyskal;PC	160,00 kn
11			Implementacija kontakt informacija	1 hr	Mon 10.11.14	Mon 10.11.14	10	PC računalo[1]; Toni Steyskal	43,00 kn
12			Izrada projektne dokumentacije	4 hrs	Wed 12.11.14	Wed 12.11.14	11	Ana Sakač; Laptop[1];Suzan	292,00 kn
13			Završetak prvog sprinta	0 days	Thu 13.11.14	Thu 13.11.14	12		0,00 kn
14			SPRINT2	12 days	Mon 8.12.14	Wed 24.12.14	2		1.286,00 kn
15			Redizajn aplikacije	4 hrs	Mon 8.12.14	Mon 8.12.14	13	Ana Sakač;Lapto	292,00 kn
16			Kreiranje informacija o rasporedu u json formatu	2 hrs	Fri 12.12.14	Fri 12.12.14	15	Matija Kovaček; PC računalo[1]	78,00 kn
17			Izrada modula za komunikaciju s firebase-om	3 hrs	Mon 15.12.14	Mon 15.12.14	16	Matija Kovaček; PC računalo[1]	113,00 kn
18			Izrade modula lokalne pohrane podataka	3 hrs	Wed 17.12.14	Wed 17.12.14	17	Laptop[1]; Matija Kovaček; PC računalo[1]	125,00 kn
19			Izrada ekrana s tabovima	2 hrs	Wed 17.12.14	Wed 17.12.14	18	PC računalo[1]; Toni Steyskal	78,00 kn
20			Implementacija logike prikaza rasporeda	1 hr	Fri 19.12.14	Fri 19.12.14	19	PC računalo[1]; Matija Kovaček	43,00 kn
21			Priprema sadržaja na engleskom jeziku	3 hrs	Mon 22.12.14	Mon 22.12.14	20	Ana Sakač; Laptop[1]; Suzana Rendulić	222,00 kn
22			Izrada modula za višezjezičnost aplikacije	3 hrs	Tue 23.12.14	Tue 23.12.14	21	Toni Steyskal; PC računalo[1]	113,00 kn
23			Izrada projektne dokumentacije	3 hrs	Tue 23.12.14	Tue 23.12.14	22	Ana Sakač; Laptop[1];Suzan	222,00 kn
24			Završetak drugog sprinta	0 days	Wed 24.12.14	Wed 24.12.14	23		0,00 kn
25			SPRINT3	13 days	Mon 12.1.15	Wed 28.1.15	2		2.256,00 kn
26			Docavanje logotipa sponzora	2 hrs	Mon 12.1.15	Mon 12.1.15	24	Laptop[1]; Suzana Rendulić	172,00 kn
27			Povezivanje logotipa s web stranicom	2 hrs	Wed 14.1.15	Wed 14.1.15	26	Matija Kovaček; PC računalo[1]	168,00 kn
28			Kreiranje logičkog i razumljivog s ijeda prikaza	3 hrs	Thu 15.1.15	Thu 15.1.15	27	PC računalo[1]; Toni Steyskal	248,00 kn
29			Docavanje slika korištenja aplikacije u pomoć	2 hrs	Fri 16.1.15	Fri 16.1.15	28	Ana Sakač; Laptop[1]	172,00 kn
30			Prikaz karte	3 hrs	Tue 20.1.15	Tue 20.1.15	29	Matija Kovaček;PC	248,00 kn
31			Pozicioniranje karte na lokaciju korisnika	2 hrs	Wed 21.1.15	Wed 21.1.15	30	PC računalo[1]; Toni Steyskal	168,00 kn
32			Povezivanje karte smjestima izvčenja	1 hr	Thu 22.1.15	Thu 22.1.15	31	Laptop[1]; Suzana Rendulić	92,00 kn
33			Docavanje logotipa aplikacije	2 hrs	Thu 22.1.15	Thu 22.1.15	32	PC računalo[1]; Matija Kovaček	168,00 kn
34			Korekcije nepravilnosti	2 hrs	Fri 23.1.15	Fri 23.1.15	33	Matija Kovaček; PC računalo[1]	168,00 kn
35			Izrada dokumentacije	4 hrs	Mon 26.1.15	Mon 26.1.15	34	Ana Sakač; Laptop[1];Suzan	652,00 kn
36			Završetak trećeg sprinta	0 days	Tue 27.1.15	Tue 27.1.15	35		0,00 kn

Slika 4. Slijed aktivnosti

### 3.4. Arhitektura

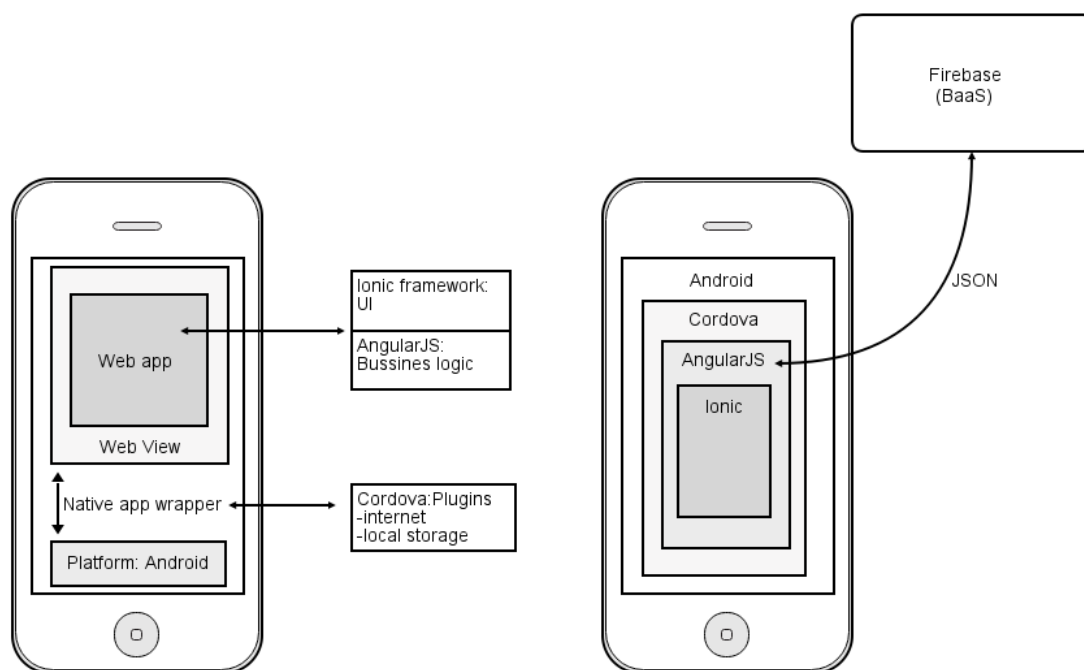
Na temelju korisničkih zahtjeva odlučili smo se za hibridni (eng. Hybrid) razvoj mobilne aplikacije. Hibridna mobilna aplikacija je ustvari web aplikacija koja je smještena u ljusci preglednika pisan u nativnom jeziku za pojedinu mobilnu platformu. Hibridni razvoj poznat je po tome da se nakon razvoja, mobilna aplikacije može koristiti na više različitih mobilnih platforma. Kod hibridne mobilne aplikacije sama web aplikacija se izvodi unutar native aplikacije tako što je povezana s nativnim 'omotačem' koji ima ulogu mosta tako što povezuje i upravlja komunikacijom između web aplikacije i native platforme mobilnog uređaja. To znači da web aplikacija se može izvoditi unutar mobilnog uređaja i da može pristupiti samom mobilnom uređaju tj. njegovim pojedinim dijelovima poput kamere, GPS-a i slično.

Za izradu mobilne aplikacije odabrane su sljedeće tehnologije:

- Ionic framework je front end framework za razvijanje hibridnih mobilnih aplikacija. Ionic pruža skup vizualnih kontrola, komponenata koje su prilagođene za mobilne uređaje i koristi se za izradu korisničkog sučelja mobilne aplikacije.
- AngularJS je JavaScript framework koji se izvršava na strani klijenta. AngularJS je poznat po MVC arhitekturi i koristit ćemo ga za razvijanje poslovne logike same aplikacije.
- Apache Cordova je skup pluginova koji omogućavaju pristup dijelovima mobilnog uređaja poput kamere, akcelerometra, gps-a i slično. Cordova ima ranije spomenutu funkciju 'mosta'.
- Firebase je takozvani 'backend as a services' (BaaS), vrsta cloud usluge za mobilne i web aplikacije koja omogućuje pohranjivanje i sinkroniziranje podataka u stvarnom vremenu. Firebase nam omogućuje jednostavnu pohranu i dohvaćanje podataka iz baze podataka pomoću njihovih RESTFul servisa.

Sve prethodno navedene tehnologije radile bi tako što bi se napočetku, tijekom pokretanja aplikacije prvo provjeravala lokalna SQLite baza podataka koju nam omogućuje Cordova plugin. Ako u lokalnoj bazi ne bi postojali potrebni podaci, tada bi se pozivao Firebase Data API kojim dohvaćamo potrebne podatke i pohranjujemo u lokalnu bazu. Svu tu komunikaciju nam omogućuje AngularJS i svu daljnju logiku poput prikazivanja podataka u odgovarajućim vizulnim komponentama Ionic frameworka ostvarujemo pomoću AngularJS.

Na slici 5. prikazana je arhitektura aplikacije.



**Slika 5. Arhitektura**

## 4. Korisnički zahtjevi

Prije kreiranja Product Backlog-a bilo je potrebno definirati korisničke zahtjeve. U tu svrhu kreiran je prikaz korisničkih priča koji možete vidjeti u tablici 1. Prilikom njezinog kreiranja stavili smo se u poziciju različitih posjetitelja Renesansnog festivala.

**Tablica 1:** *Korisničke priče*

#	Ja kao...	Želim...	Prioritet
1.	Strani korisnik	Biti u mogućnosti pristupiti aplikaciji na engleskom jeziku.	Visok
2.	Strani korisnik / korisnik iz drugog grada	Znati kako doći do mjesta gdje se festival odvija.	Visok
3.	Strani korisnik / korisnik iz drugog grada	Znati gdje se nalaze okolni hoteli i restorani ako ću se u gradu zadržati nekoliko dana.	Srednji
4.	Strani korisnik / korisnik iz drugog grada	Znati kontakt podatke Renesansnog festivala i Turističke Zajednice Koprivnica	Nizak
5.	Strani korisnik / korisnik iz drugog grada / korisnik iz KC	Znati raspored odvijanja programa Renesansnog festivala	Visok
6.	Strani korisnik / korisnik iz drugog grada / korisnik iz KC	Znati tko su sponzori Renesansnog festivala	Nizak
7.	Strani korisnik / korisnik iz drugog grada / korisnik iz KC	Pročitati korisne informacije o Renesansnom festivalu i njegovim sudionicima	Srednji
8.	Strani korisnik / korisnik iz drugog grada / korisnik iz KC	Da mi glavne funkcionalnosti budu dostupne odmah na početnom ekranu, a svima da mogu pristupiti kroz glavni izbornik	Visok



Iz tablice 1. Korisniče priče može se vidjeti kako smo korisnike podijelili u tri grupe koje obuhvaćaju strane korisnike iz drugih država, korisnike iz drugih gradova RH i „domaće“ korisnike iz Koprivnice – mjesta gdje se renesansni festival odvija. Sukladno tome definirani su korisnički zahtjevi:

- Pregled osnovnih informacija na glavnom ekranu – nakon početnog „Welcome“ ekrana sa logotipom koji se otvara odmah nakon što korisnik pokrene aplikaciju prikazuje se glavni ekran. Na njemu korisnik može pristupiti informacijama o renesansi, programu i mjestu događaja.
- Prikaz informativnih podataka o Renesansnom festivalu – ukoliko korisnik sa početnog ekrana ili izbornika odabere opciju „O Renesansi“ pristupit će informacijama o Renesansnom festivalu. Na samom početku prikazana je simbolična slika kralja i kraljice kako bi se uskladila tematika festivala sa izgledom aplikacije. Ispod slike nalaze se informacije o očuvanju tradicije starih vremena i samom festivalu.
- Pregled programa festivala – korisnik može pregledavati program za svaki dan festivala. Program je poredan kronološki po vremenu odvijanja određenog događaja i po danima. Uz podatke o vremenu odvijanja događaja dodan je slikovni prikaz te dodatne informacije za svakog od njih kako bi si korisnici lakše mogli isplanirati odlazak na festival i odlučiti koji događaji im se čine zanimljivim za posjetiti.
- Prikaz mape mjesta događaja – ova funkcionalnost je od velike važnosti stranim posjetiteljima, ali i posjetiteljima iz drugih gradova. Korištenjem Google mape korisnicima bi bile prikazane točne lokacije odvijanja festivala te kako doći do njih. Također bi mogli vidjeti podatke kao što su hoteli ili restorani koji se nalaze u blizini odvijanja festivala.
- Prijevod aplikacije na engleski jezik – korisnicima će kroz glavni izbornik biti omogućena opcija prijevoda aplikacije na engleski jezik. Na taj način prilagođavamo se stranim posjetiteljima i olakšavamo im snalaženje u aplikaciji i na festivalu.
- Prikaz informacija o sponzorima i Turističkoj Zajednici Koprivnice – glavni organizator Renesansnog festivala je Turistička Zajednica Koprivnice čijim će se kontakt podacima moći pristupiti kroz glavni izbornik. Također će biti dostupna i opcija pregleda sponzora koji su pomogli prilikom organizacije festivala. Na taj način odvija se obostrana promocija koja je od velike važnosti za ostvarivanje dobre suradnje i uspjeha aplikacije.

## **4.1. Nefunkcionalni zahtjevi**

Kako bi definirali ponašanje sustava u smislu njegovih performansi, sigurnosti i mogućnosti korištenja odredili smo nefunkcionalne zahtjeve naše aplikacije:

- Aplikacija mora omogućiti offline način rada kako bi korisnici mogli koristiti aplikaciju i kada nisu povezani na Internet, nakon što se korisnik poveže na Internet aplikacija se mora ažurirati na posljednju dostupnu verziju.
- Instalacija i deinstalacija aplikacije mora biti jednostavna za korisnika što znači da je potrebno korisniku priložiti sve potrebne upute za rukovanje aplikacijom.
- U aplikaciji treba osigurati mogućnost za laku nadogradnju novih funkcionalnosti, ako će to u budućnosti biti potrebno.

## 5. Product Backlog

Product Backlog predstavlja listu značajki, funkcionalnosti, zahtjeva, željenih poboljšanja i ispravaka koje čine promjene koje će se izvršiti nad proizvodom u toku budućih izdanja. Glavna mu je karakteristika da je promjenjiv. On se može mijenjati tijekom cijelog životnog ciklusa proizvoda (od ideje, pa do povlačenja s tržišta). On predstavlja živ artefakt gdje se stavke sortiraju prema vrijednosti, od najvrijednije na vrhu.

Početni Product Backlog aplikacije RenFest jest:

- Aplikacija mora prikazivati osnovne informacije o renesansnom festivalu u tekstualnom obliku (uvod u tematiku, opis programa i događanja)
  - **PRIORITET:** visok
  - **ROK:** mjesec dana
  - **STATUS:** Završeno
- Prikaz funkcionalnosti na početnom ekranu kako bi korisnici lakše dobili sliku što im aplikacija pruža
  - **PRIORITET:** visok
  - **ROK:** mjesec dana
  - **STATUS:** Završeno
- Odabir funkcionalnosti koje korisnik aplikacije bira putem izbornika
  - **PRIORITET:** visok
  - **ROK:** mjesec dana
  - **STATUS:** Završeno
- Prikaz kontakt informacije turističke zajednice
  - **PRIORITET:** visok
  - **ROK:** mjesec dana
  - **STATUS:** Završeno
- Unutar aplikacije mora se nalaziti raspored svih događanja vezanih uz Renesansi Festival u obliku organizirane tablice grupirano po određenom danu festivala, svaki događaj moguće je detaljnije pregledati i vidjeti prošlogodišnji medijski sadržaj s tog događaja

- **PRIORITET:** visok
  - **ROK:** dva mjeseca
  - **STATUS:** Završeno
- Sadržaj aplikacije i cijela aplikacija mora biti dostupna na engleskom jeziku za strane turiste koji posjećuju Renesansni festival
    - **PRIORITET:** srednji
    - **ROK:** dva mjeseca
    - **STATUS:** Završeno
- Prikaz opcija navigacije kroz renesansi festival, točnije, prikaz karte koja sadrži označene točke interesa za posjetitelje (mjesto otvaranja festivala, mjesta svih događaja vezanih uz festival i mjesto ceremonije zatvaranja festivala), svaka točka interesa prikazuje dodatno osnovne informacije vezane uz taj događaj (početak, kraj, kratak opis)
    - **PRIORITET:** nizak
    - **ROK:** tri mjeseca
    - **STATUS:** Završeno
- Dizajn aplikacije mora biti usklađen s tematikom renesansnog festivala, izgled aplikacije mora biti povezan s srednjim vijekom, stilom viteza, princeza, raznih oružja iz tog doba i dvoraca
    - **PRIORITET:** nizak
    - **ROK:** tri mjeseca
    - **STATUS:** Završeno

## 6. Sprint1 Backlog

Kao što smo već spomenuli prilikom definiranja razvojnog tima, Sprint Backlog je popis stavaka Product Backlog-a na kojem razvojni tim radi. Iz izrađenog Product Backlog-a odabiru se one stavke za koje razvojni tim smatra da će izvršiti u prvom sprintu nakon čega slijedi definiranje podskup stavaka. Sprint Backlog je vidljiva slika posla u realnom vremenu kojeg razvojni tim namjerava obaviti tijekom Sprint-a i koji pripada isključivo razvojnog timu.

Naš razvojni tim definirao je rok prvog Sprint-a i postavljen na datum 12. studenog 2014. godine. Cilj sprinta je proizvesti upotrebljivu aplikaciju površne funkcionalnosti, odnosno uvesti dizajn i omogućiti prolaz kroz određene dijelove aplikacije. Pri kraju prvog sprinta aplikacija će prikazivati osnovne informacije o renesansnom festivalu i posložene će biti sve stranice za buduće funkcionalnost, no one će u prvoj iteraciji biti prazne. Zadaci koji će biti postavljeni na alat koji će Scrum Team koristiti za kontinuirano praćenje i raspored izvršavanja zadataka su:

- implementacija vizualnog djela početnog ekrana
- implementacija logike početnog ekrana
- implementacija vizualnog djela izbornika
- implementacija logike izbornika
- implementacija vizualnog djela ekrana za prikaz osnovnih informacija
- implementacija logike prikaza osnovnih informacija
- vizualna implementacija kontakt informacija
- implementacija kontakt informacija

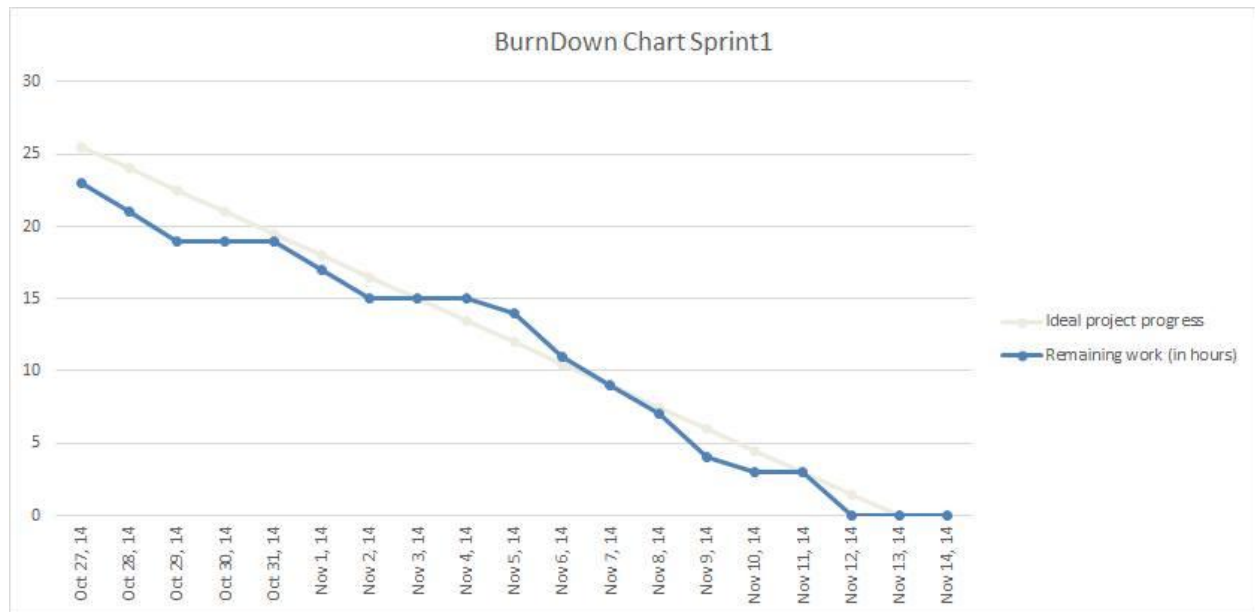
Kako to točno izgleda, prikazuje tablica 2. Sprint 1 Backlog.

**Tablica 2. Sprint 1 Backlog**

<b>Sprint Backlog</b>	<b>Zadaci</b>	<b>Sati rada</b>	<b>Status</b>	<b>Korisnička priča</b>
<b>Prikaz funkcionalnosti na početnom ekranu</b>	- implementacija vizualnog djela početnog ekrana	2 sata	Završeno	<b>8.</b>
	- implementacija logike početnog ekrana	3 sata	Završeno	
<b>Odabir funkcionalnosti aplikacije putem izbornika</b>	- implementacija vizualnog djela izbornika	3 sata	Završeno	<b>8.</b>
	- implementacija logike izbornika	2 sata	Završeno	
<b>Prikaz osnovnih informacija o Renesansnom festivalu</b>	- implementacija vizualnog djela ekrana za prikaz osnovnih informacija	3 sata	Završeno	<b>7.</b>
	- implementacija logike prikaza osnovnih informacija	2 sata	Završeno	
<b>Prikaz kontakt informacija turističke zajednice</b>	- vizualna implementacija kontakt informacija	2 sata	Završeno	<b>4.</b>
	- implementacija kontakt informacija	1 sat	Završeno	

## 6.1. BurnDown Chart

Način raspodjele sati rada kroz prvi sprint vidljiv je na BurnDown Chart-u prvog Sprint-a, graf 2.



**Graf 2:** *Sprint1- BurnDown Chart*

## 7. Sprint2 Backlog

Trajanje drugog Srinta definirano je od 8. prosinca do 24. prosinca 2014. godine. Cilj je bio nadopuniti funkcionalnosti aplikacije iz prvog Srinta kako bi aplikacija što više odgovarala korisničkim zahtjevima koji su definirani. Određen je skup zadataka čijim će se izvršavanjem izrada aplikacije provoditi kraju. Tablica 3. Sprint 2 Backlog prikazuje sljedeće zadatke:

- Kreiranje informacija o rasporedu u json formatu
- Izrada modula za komunikaciju s firebase-om
- Izrade modula lokalne pohrane podataka
- Implementacija logike prikaza rasporeda
- Izrada ekrana s tabovima
- Priprema sadržaja na engleskom jeziku
- Izrada modula za višejezičnost aplikacije
- Izrada dizajna

**Tablica 3. Sprint 2 Backlog**

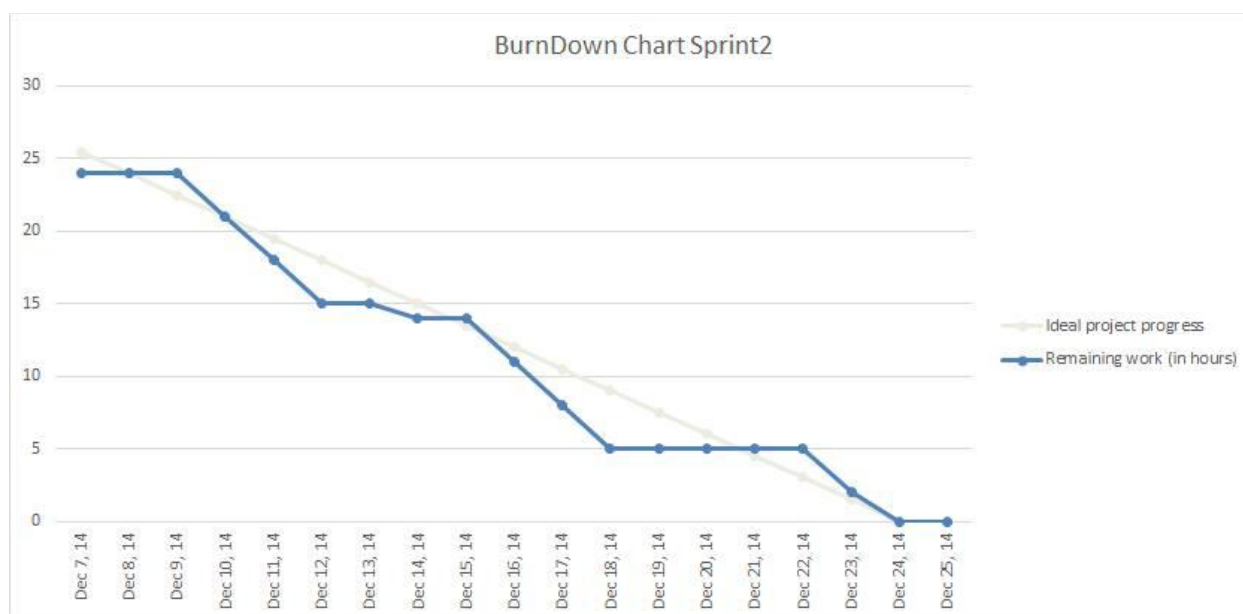
Sprint Backlog	Zadaci	Sati rada	Status	Korisnička priča
<b>Prikaz rasporeda programa Renesansnog festivala</b>	- Kreiranje informacija o rasporedu u json formatu	3 sata	Završeno	<b>5.</b>
	- Izrada modula za komunikaciju s firebase-om	4 sata	Završeno	
	- Izrade modula lokalne pohrane podataka	4 sata	Završeno	
	- Implementacija logike prikaza rasporeda	3 sata	Završeno	
	- Izrada ekrana s tabovima	2 sat	Završeno	
<b>Aplikacija i sadržaj aplikacije na engleskom jeziku</b>	- Priprema sadržaja na engleskom jeziku	4 sata	Završeno	<b>1.</b>
	- Izrada modula za višejezičnost aplikacije	4 sata	Završeno	



Redizajn aplikacije usklađen s tematikom renesansnog festivala	- Izrada dizajna	4 sata	Završeno	
----------------------------------------------------------------------------	------------------	--------	----------	--

## 7.1. BurnDown Chart

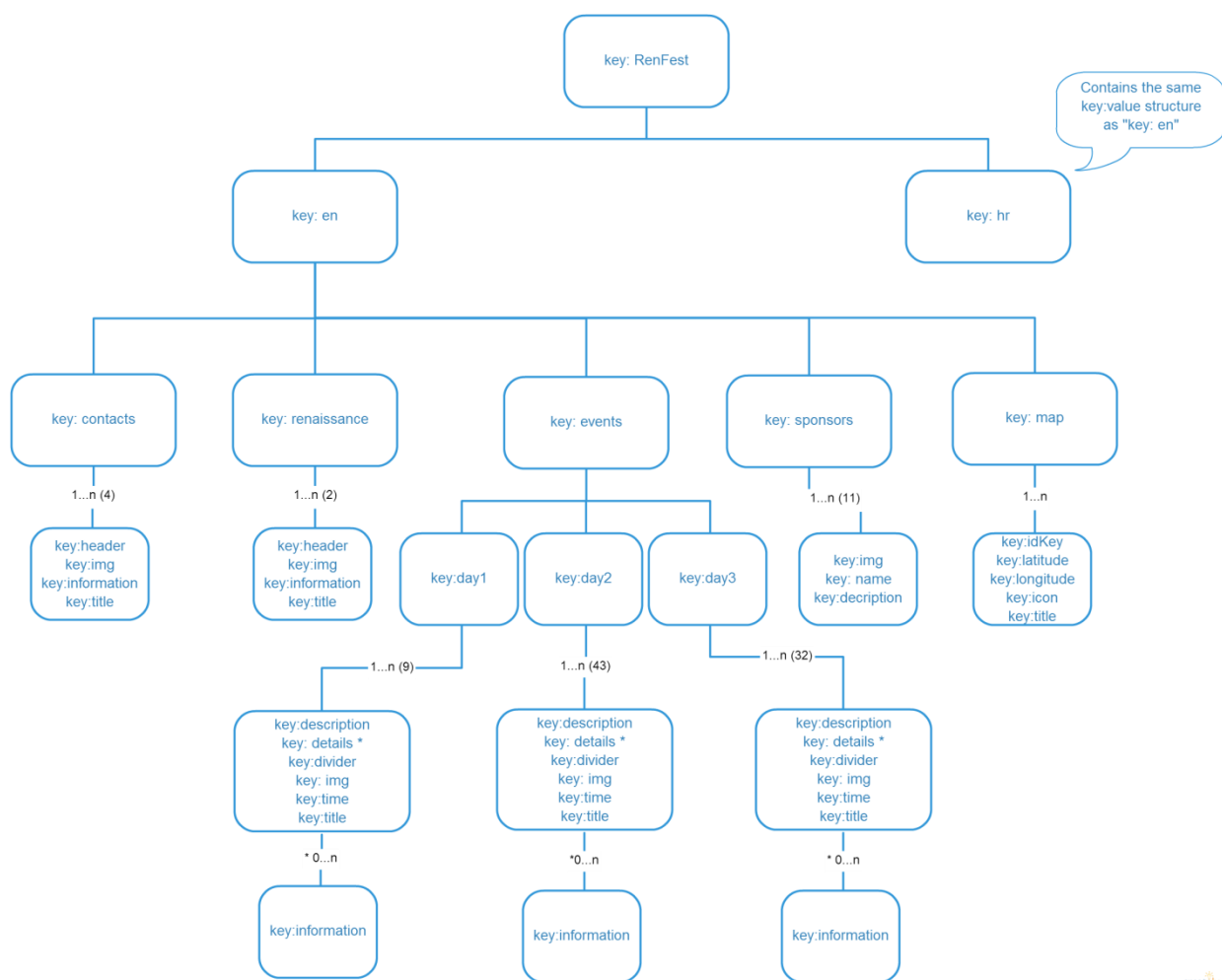
Način raspodjele sati rada kroz drugi sprint vidljiv je na BurnDown Chart-u drugog Sprint-a, graf 3.



**Graf 3:** Sprint2- BurnDown Chart

## 7.2. Model podataka

Kod hibridnog mobilnog razvoja lokalnu pohranu podataka moguće je pohraniti pomoću `cache-a`, `Web storage-a` (`localStorage` i `sessionStorage`), `WebSQL storage-a` (`SQLite`) i `IndexedDB storage-a`. Za ovaj projekt odlučili smo se za `localStorage` pohranu podataka. `LocalStorage` omogućuje pohranu kolekcije podataka u obliku `key : value` parova na mobilni uređaj, pri tome `key` i `value` moraju biti u obliku stringova. `LocalStorage` nam se činio kao najbolji izbor zbog vrste podataka (velike količine teksta) koje cloud servis (Firebase) vraća, Firebase vraća kolekciju podataka u obliku `key : value` parova i to u JSON formatu. Prema tome moglo bi se reći da koristimo NoSQL tip baze podataka. NoSQL je horizontalno skalabilan, nije relacijski oblik baze podataka i ima fleksibilnu shemu podataka. Kako nismo koristili relacijski oblik baze podataka, ne možemo prikazati model podataka u obliku ERA modela, već na sljedeći način:



Slika 6. Struktura modela podataka

Zanimalo nas je kako stvarno izgleda localStorage pohrana na mobilnom uređaju pa smo pomoću Android device monitora došli do localStorage datoteke na mobilnom uređaju (data/data/naša\_app/app\_webview/LocalStorage/ime\_datoteke.LocalStorage). U localStorage postoji jedna tablica *Item table* koja ima attribute *key* i *value* (slika 7).

ItemTable	
key	TEXT
value	BLOB
Indexes	
sqlite_autoindex_ItemTable_1	

Slika 7. Item tablica

Schema:

```
CREATE TABLE ItemTable
(
  key TEXT UNIQUE ON CONFLICT REPLACE,
  value BLOB NOT NULL ON CONFLICT FAIL
)
```

Na sljedećoj slici možemo vidjeti vrijednosti pohranjene u atributima *key* i *value* tablice *ItemTable*. U atributima *value* pohranjena je kolekcija podataka u obliku *key : value*.

Table: ItemTable		New Record	Delete
key	value		
Filter	Filter		
1 firebase:host:renfest.firebaseio.com	(BLOB)		
2 ngStorage-contacts	(BLOB)		
3 ngStorage-data	(BLOB)		
4 ngStorage-events	(BLOB)		
5 ngStorage-language	(BLOB)		
6 ngStorage-rennaissance	(BLOB)		

Edit database cell	
Import	Export
Text ▼ Clear	
Warning: Editing binary content in text mode may result in corrupted data!	
<pre>[{"header": "Turisti", "img": "https://res.cloudinary.com/dffwv0cvc/image/upload/v1420403131/renfest/contact1.jpg", "information": "Trg bana Josipa Jelačića 7, Koprivnica", "title": "Adresa"}, {"header": "null", "img": "null", "information": "+385 48 621 433", "title": "Telefon"}, {"header": "null", "img": "null", "information": "+385 48 623 178", "title": "Fax"}, {"header": "null", "img": "null", "information": "tzig-koprivnica@kc.t-com.hr", "title": "Email"}, {"header": "null", "img": "null", "information": "www.renesansfestival.hr", "title": "Web"}]</pre>	
Type of data currently in cell: Binary 1086 byte(s)	
OK	Cancel

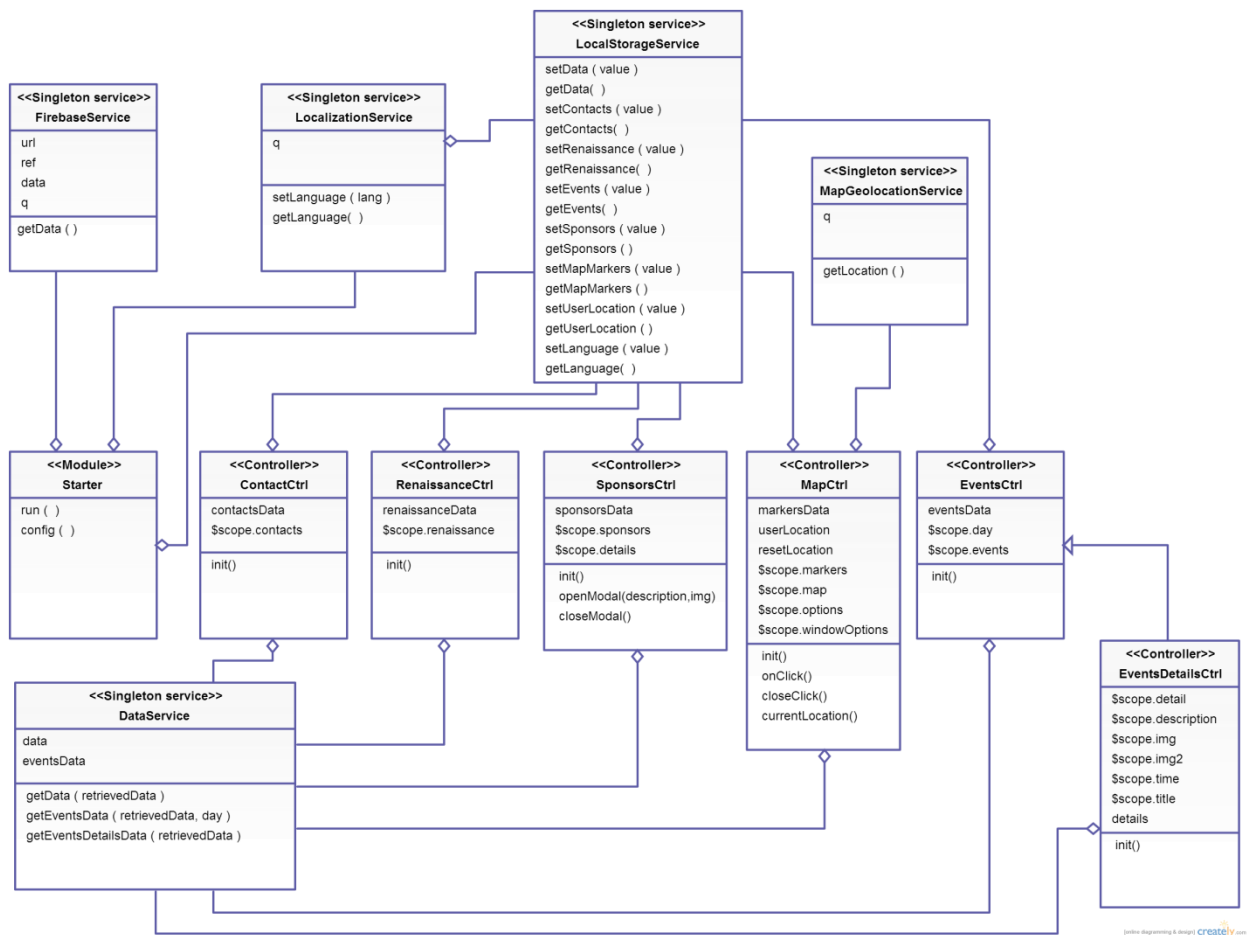
Slika 8. Pohranjene vrijednosti u Item tablici

### 7.3. Dijagram klasa

Za razliku od Jave, AngularJs ima malo drugačiju sintaksu, neki svoj stil organizacije i pisanje koda. Kod AngularJs poslovna logika piše se u kontrolerima (eng. Controllers) i u servisima (eng. Services). U kontrolerima kao što je već spomenuto piše se poslovna logika koja se izvršava iza pogleda (eng View). Kontroleri uglavnom služe da prikažu podatke u određenim pogledima te da izvršavaju neke funkcije kao odgovor na korisnikovu interakciju s korisničkim sučeljem. Zatim poslovna logika može se s vremenom zakomplicirati i povećati kako projekt raste i kako se dodaju nove funkcionalnosti, zato postoje servisi. Servisi također služe za pisanje poslovne logike, no oni se koriste kada se neki servis, neka njegova metoda može iskoristiti na više mjesta, u više kontrolera. Također da kontroleri ne bi bili nepregledni i ako vidimo da neki dio koda ponavljamo u više kontrolera tada taj kod izdvajamo i 'selimo' u servis, odnosno kreiramo metodu koja nam vraća taj isti rezultat, zatim tu metodu pozivamo u kontrolerima. AngularJs servisi su singleton i sve njegove metode su factory.

AngularJs aplikacije koristi 'kontejnere' (eng. Module) gdje svaki modul sadrži odgovarajuće kontrolere i servise. Neka praksa je da se aplikacija podijeli na više modula, u našem slučaju po funkcionalnostima. Prema tome za neku funkcionalnost aplikacije postoji odgovarajući modul koji sadrži određene kontrolere, servise koji omogućuju da ta funkcionalnost funkcionira. Kada se ti svi moduli spoje čine finalni proizvod, tj. aplikaciju sa svim funkcionalnostima.

Dijagram klasa vidljiv je na slici 9.



Slika 9. AngularJS dijagram klasa

U nastavku slijedi opis dijagrama klasa.

### FirestoreService

*FirestoreService* je servis koji služi za dohvaćanje podataka s Firebase cloud-a. *FirestoreService* servis pomoću preuzete *AngularFire* biblioteke također omogućuje i sinkronizaciju podataka u realnom vremenu. Firebase cloud vraća podatke u JSON formatu.

Varijable	Opis
url	Sadrži url putanju do pohranjenih podataka
ref	Sadrži Firebase konstruktor s <i>url</i> varijablom, služi kao Firebase referenca
data	Sadrži sinkronizirane podatke
q	Sadrži vrijednost koja omogućava asinkrono izvršavanje metoda
Metode	Opis
getData()	Metoda vraća kolekciju podataka u obliku <i>key:value</i>

### LocalizationService

*LocalizationService* je servis za lokalizaciju aplikacije, tj. da bude dostupna na više jezika. Servis uz pomoću cordova plugin-a (*Apache Globalization Plugin*) saznaje koji jezik se koristi na smartphone.

Varijable	Opis
q	Sadrži vrijednost koja omogućava asinkrono izvršavanje metoda
Metode	Opis
getLanguage()	Metoda vraća odabrani jezik na mobitelu.
setLanguage(lang)	Metoda kao ulazni parametar zaprima odabrani jezik na mobitelu u obliku stringa (tipa 'hr', 'en') i pohranjuje navedenu vrijednost u lokalnu bazu.  Koristi <i>setLanguage(lang)</i> metodu iz <i>LocalStorageService</i> servisa.

## LocalStorageService

LocalStorageService je servis za lokalnu pohranu podataka koristeći localStorage. LocalStorage omogućuje pohranu podataka u obliku *key:value*.

Metode	Opis
setData(value)	Metoda kojom potvrđujemo da su pohranjeni podaci u <i>localStorage</i> .
getData()	Metoda vraća da li postoje pohranjeni podaci u <i>localStorage</i> .
setContacts (value)	Metoda za pohranjivanje podataka o kontaktima u <i>localStorage</i> , ulazni parametar su podaci o kontaktima.
getContacts ( )	Metoda za dohvaćanje podataka o kontaktima iz <i>localStorage</i> .
setRenaissance (value)	Metoda za pohranjivanje podataka o renesansi u <i>localStorage</i> , ulazni parametar su podaci o renesansi.
getRenaissance ( )	Metoda za dohvaćanje podataka o renesansi iz <i>localStorage</i> .
setEvents (value)	Metoda za pohranjivanje podataka o događajima u <i>localStorage</i> , ulazni parametar su podaci o događajima.
getEvents ( )	Metoda za dohvaćanje podataka o događajima iz <i>localStorage</i> .
setSponsors (value)	Metoda za pohranjivanje podataka o sponzorima u <i>localStorage</i> , ulazni parametar su podaci o sponzorima.
getSponsors ( )	Metoda za dohvaćanje podataka o sponzorima iz <i>localStorage</i> .
setMapMarkers (value)	Metoda za pohranjivanje podataka potrebnih za google kartu u <i>localStorage</i> , ulazni parametar su podaci potrebni za google kartu.
getMapMarkers ( )	Metoda za dohvaćanje podataka potrebnih za google kartu iz <i>localStorage</i> .
setUserLocation (value)	Metoda za pohranjivanje podataka o korisnikovoj trenutnoj lokaciji u <i>localStorage</i> , ulazni parametar su podaci o trenutnoj lokaciji korisnika.
getUserLocation	Metoda za dohvaćanje podataka o trenutnoj korisnikovoj lokaciji iz <i>localStorage</i> .
setLanguage(value)	Metoda za pohranjivanje podataka o odabranom jeziku na mobitelu u <i>localStorage</i> , ulazni parametar su podaci o jeziku korištenja.
getLanguage()	Metoda za dohvaćanje podataka o odabranom jeziku na mobitelu iz <i>localStorage</i> .

## DataService

DataService je servis za manipulaciju/organizaciju podataka. Ovaj servis je nastao iz razloga jer se u nekim kontrolerima ponavljao isti kod. (prolazak po podacima kroz for each petlju, push u polje i slično.)

Variable	Opis
data	Sadrži podatke koje vraća metoda.
eventsData	Sadrži podatke o eventima za određeni dan.
Metode	Opis
getData(retrievedData)	Metoda vraća polje podataka. Ulazni parametar su podaci koje je potrebno organizirati i staviti u polje.
getEventsData(retrievedData,day)	Metoda vraća polje podataka. Ulazni parametar su podaci koje je potrebno organizirati i staviti u polje, te sljedeći parametar sadrži podatak o kojem danu renesansnog festivala se radi.
getEventsDetailsData(retrievedData)	Metoda vraća polje podataka. Ulazni parametar su podaci koje je potrebno organizirati i staviti u polje.

## MapGeolocationService

MapGeolocationService je servis za dohvaćanje podataka o trenutnoj korisnikovoj geolokaciji pomoću Apache Cordova Geolocation plugina.

Variable	Opis
q	Sadrži vrijednost koja omogućava asinkrono izvršavanje metoda
Metode	Opis
getLocation()	Metoda vraća trenutnu korisnikovu geolokaciju.

## Starter

Starter je naziv modula koji sadrži sve ostale kreirane module. Starter modul sadrži dvije funkcije run() i config(). To su funkcije koje se izvršavaju tijekom 'bootstrap' procesa, odnosno u pozadini i brinu se da 'sve' funkcionira.

Funkcije	Opis
run()	Nakon što je platforma spremna, u pozadini se provjerava da li postoje podaci u



	<p><i>localStorage</i>, ako ne postoje provjerava se <i>odabrani jezik korištenja</i> na mobitelu kako bi se preuzeli podaci prilagođeni tom jeziku. Zatim provjerava se da li postoji <i>internet konekcija</i> (pomoću <i>apache.cordova.network</i> plugina), ako postoji <i>preuzimaju se podaci s clouda i pohranjuju u localStorage</i>, a ako nema internetske veze upozorava se korisnika da uključi (bitno je tijekom prvog korištenja da bude uključen internet). U slučaju da postoje podaci i ako su podaci na cloudu izmijenjeni dolazi do <i>sinkronizacije podataka</i>.</p> <p>Koristi sljedeće metode:</p> <p><i>LocalStorageService</i>: <code>getData()</code>, <code>getLanguage()</code>, <code>setData(val)</code>, <code>setContacts(val)</code>, <code>setRennaisance(val)</code>, <code>setEvents(val)</code></p> <p><i>LocalizationService</i>: <code>getLanguage()</code>, <code>setLanguage()</code></p> <p><i>FirebaseService</i>: <code>getData()</code></p>
<code>config()</code>	<p>Ovdje su konfigurirani state-ovi/route, gdje je navedeno kada se dođe na određenu putanju koji html template prikazati, koji kontroler koristiti i slično.</p> <p>Također ovdje su konfigurirani stringovi (poput <code>strings.xml</code> u androidu), ujedno je konfigurirano kako se ti stringovi moraju prikazati na određenom jeziku mobitela.</p>

## ContactCtrl

ContactCtrl je kontroler koji služi za prikaz kontakt podataka o Turističkoj agenciji grada Koprivnica.

Varijable	Opis
<code>contactsData</code>	Sadrži kontakt podatke koji su dobiveni pomoću metode <code>getContacts()</code> iz <i>LocalStorageService</i> servisa.
<code>\$scope.contacts</code>	Sadrži organizirane kontakt podatke dobivene pomoću metode <code>getData(contactsData)</code> iz <i>DataService</i> servisa. Podaci iz te varijable se prikazuju u

	<i>pogledu</i> (View).
Metode	Opis
Init()	Metoda koja preuzima kontakt podatke iz <i>localStorage</i> , proslijeđuje ih metodi za organizaciju podataka <i>getData(contactsData)</i> iz <i>DataService</i> servisa, te rezultat pohranjuje u <i>\$scope.contacts</i> kako bi se podaci mogli prikazati korisniku u <i>pogledu</i> (View).

## RenaissanceCtrl

RenaissanceCtrl je kontroler koji služi za prikaz podataka o renesansi.

Varijable	Opis
renaissanceData	Sadrži podatke o renesansi koji su dobiveni pomoću metode <i>getRenaissance()</i> iz <i>LocalStorageService</i> servisa.
\$scope.renaissance	Sadrži organizirane podatke o renesansi dobivene pomoću metode <i>getData(renaissanceData)</i> iz <i>DataService</i> servisa. Podaci iz te varijable se prikazuju u <i>pogledu</i> (View).
Metode	Opis
Init()	Metoda koja preuzima podatke o renesansi iz <i>localStorage</i> , proslijeđuje ih metodi za organizaciju podataka <i>getData(renaissanceData)</i> iz <i>DataService</i> servisa, te rezultat pohranjuje u <i>\$scope.renaissance</i> kako bi se podaci mogli prikazati korisniku u <i>pogledu</i> (View).

## SponsorsCtrl

SponsorsCtrl je kontroler koji služi za prikaz podataka o sponzorima.

Varijable	Opis
sponsorsData	Sadrži podatke o sponzorima koji su dobiveni pomoću metode <i>getSponsors()</i> iz <i>LocalStorageService</i> servisa.
\$scope.sponsors	Sadrži organizirane podatke o sponzorima dobivene pomoću metode <i>getData(sponsorsData)</i> iz <i>DataService</i> servisa. Podaci iz te varijable se prikazuju u <i>pogledu</i> (View).
\$scope.details	Sadrži dodatne detalje i url slike o sponzoru.
Metode	Opis

init()	Metoda koja preuzima podatke o sponzorima iz <i>localStorage</i> , proslijeđuje ih metodi za organizaciju podataka <i>getData(sponsorsData)</i> iz <i>DataService</i> servisa, te rezultat pohranjuje u <i>\$scope.sponsors</i> kako bi se podaci mogli prikazati korisniku u <i>pogledu</i> (View).
openModal (description, img)	Metoda koja otvara modal komponentu korisničkog sučelja s dodatnim opisom i slikom o sponzoru. Kao ulazne parametre metoda zaprima opis i sliku sponzora.
closeModal ( )	Metoda koja zatvara modal komponentu korisničkog sučelja.

## MapCtrl

MapCtrl je kontroler koji služi za prikaz podataka o na google karti.

Varijable	Opis
markersData	Sadrži podatke o potrebne za google kartu koji su dobiveni pomoću metode <i>getMapMarkers()</i> iz <i>LocalStorageService</i> servisa.
userLocation	Sadrži podatke o trenutnoj korisnikovoj lokaciji.
resetLocation	Sadrži podatke da je korisnikova lokaciji nedefinirana.
\$scope.markers	Sadrži organizirane podatke potrebne za google kartu dobivene pomoću metode <i>getData(markersData)</i> iz <i>DataService</i> servisa. Podaci iz te varijable se prikazuju u <i>pogledu</i> (View).
\$scope.map	Sadrži osnovne podatke potrebne za prikaz google karte. (koordinate prema kojoj se centrirala karta i zoom)
\$scope.options	Sadrži podatak da je scrool onemogućen.
\$scope.windowOptions	Sadrži podatak da li je informativan prozor o pojedinom markeru otvoren.
Metode	Opis
Init( )	Metoda za prikazivanje google karte korisniku u <i>pogledu</i> (View).
onClick( )	Metoda koja otvara dodatne informacije za pojedini marker.
closeClick( )	Metoda zatvara informacije za pojedini marker.
currentLocation	Metoda koja stavlja marker o trenutnoj korisnikovoj poziciji na google kartu pomoću metode <i>getLocation( )</i> iz <i>MapGeolocationService</i> servisa.

## EventsCtrl

EventsCtrl je kontroler koji služi za prikaz informacija o rasporedu programa renesansnog festivala.

Varijable	Opis
eventsData	Sadrži podatke o rasporedu programa koji su dobiveni pomoću metode <i>getEvents()</i> iz <i>LocalStorageService</i> servisa.
\$scope.day	Sadrži podatak o kojem se točno danu renesansnog festivala radi. Taj podatak se dobiva iz angular <i>\$stateParams</i> servisa koji uzima parametar iz url putanje.
\$scope.events	Sadrži organizirane podatke o rasporedu programa dobivene pomoću metode <i>getEventsData(eventsData, \$stateParams.day)</i> iz <i>DataService</i> servisa. Podaci iz te varijable se prikazuju u <i>pogledu</i> (View).
Metode	Opis
Init()	Metoda koja preuzima podatke o rasporedu programa iz <i>localStorage</i> , prosljeđuje ih metodi za organizaciju podataka <i>getEventsData(eventsData, \$stateParams.day)</i> iz <i>DataService</i> servisa, te rezultat pohranjuje u <i>\$scope.events</i> kako bi se podaci mogli prikazati korisniku u <i>pogledu</i> (View).

## EventsDetailsCtrl

EventsDetailsCtrl je kontroler koji služi za prikaz dodatnih informacija, detalja o određenom programu renesansnog festivala ako postoje.

Varijable	Opis
\$scope.detail	Sadrži organizirane podatke koje sadrže detalje o određenom programu, dobivene pomoću metode <i>getEventsDetailsData(details)</i> iz <i>DataService</i> servisa. Podaci iz te varijable se prikazuju u <i>pogledu</i> (View)
\$scope.description	Nasljeđeni podatak iz <i>EventsCtrl</i> kontrolera, sadrži opis određenog programa.
\$scope.img	Nasljeđeni podatak iz <i>EventsCtrl</i> kontrolera, sadrži url putanju servisa gdje je pohranjena slika.
\$scope.img2	Nasljeđeni podatak iz <i>EventsCtrl</i> kontrolera, sadrži url putanju servisa gdje je pohranjena slika 2.
\$scope.time	Nasljeđeni podatak iz <i>EventsCtrl</i> kontrolera, sadrži vrijeme određenog programa.
\$scope.title	Nasljeđeni podatak iz <i>EventsCtrl</i> kontrolera, sadrži naslov određenog programa.

details	Nasljeđeni podatak iz <i>EventsCtrl</i> kontrolera, sadrži detalje određenog programa.
<b>Metode</b>	<b>Opis</b>
Init()	Metoda koja preuzima nasljeđene podatke iz <i>EventsCtrl</i> kontrolera, proslijeđuje varijablu <i>details</i> metodi za organizaciju podataka <i>getEventsDetailsData(details)</i> iz <i>DataService</i> servisa, te rezultat pohranjuje u <i>\$scope. detail</i> kako bi se podaci mogli prikazati korisniku u <i>pogledu</i> (View).

## 8. Sprint3 Backlog

Trajanje trećeg Srinta definirano je od 11. siječnja do 27. siječnja 2015. godine. Cilj zadnjeg sprinta je bio ispuniti sve korisničke zahtjeve preostale od prvog i drugog sprinta i nadodati funkcionalnosti koje zaokružuju cjelokupnu aplikaciju. Nakon ovog sprinta aplikacija predstavlja finalni proizvod koji je zatražen od strane vlasnika proizvoda. Sprint 3 Backlog prikazuje sljedeće zadatke:

- Izrada ekrana sponzora
- Izrada ekrana za pomoć snalaženja u korištenju aplikacije
- Kreiranje prikaza karte
- Povezivanje karte s mjestima izvođenja događaja, okolnim hotelima i restoranima
- Završne izmjene u dizajnu

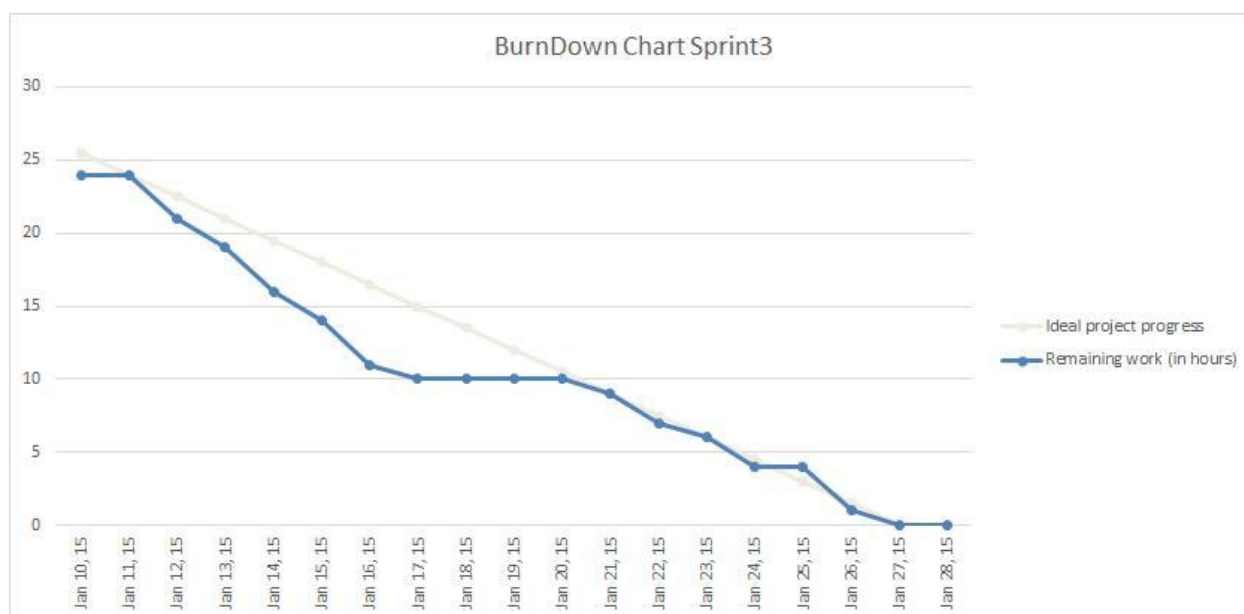
**Tablica 4. Sprint 3 Backlog**

Sprint Backlog	Zadaci	Sati rada	Status	Korisnička priča
<b>Izrada ekrana sponzora</b>	- Dodavanje logotipa sponzora	2 sat	Završeno	<b>6.</b>
	- Povezivanje logotipa pojedinih sponzora s pripadajućim web stranicama	2 sat	Završeno	
<b>Izrada ekrana za pomoć snalaženju u korištenju aplikacije</b>	- Kreiranje logičkog i razumljivog slijeda prikaza korištenja aplikacije	3 sata	Završeno	<b>8.</b>
	- Dodavanje slika korištenja aplikacije u pomoć	2 sat	Završeno	
<b>Kreiranje prikaza karte</b>	- Prikaz karte	3 sata	Završeno	<b>2.</b>
	- Pozicioniranje karte na lokaciju korisnika ili, ukoliko nije moguće, na lokaciju festivala	2 sat	Završeno	

<b>Povezivanje karte s mjestima izvođenja događaja, okolnim hotelima i restoranima</b>	- Povezivanje karte za pregled mjesta gdje bi korisnik mogao otići	1 sat	Završeno	<b>2.</b> <b>3.</b>
<b>Završne izmjene u dizajnu</b>	- Dodavanje logotipa aplikacije	2 sat	Završeno	
	- Korekcije nepravilnosti	2 sat	Završeno	

## 8.1. BurnDown Chart

Način raspodjele sati rada kroz treći sprint vidljiv je na BurnDown Chart-u trećeg Sprint-a, graf 4.



**Graf 4:** *Sprint3- BurnDown Chart*

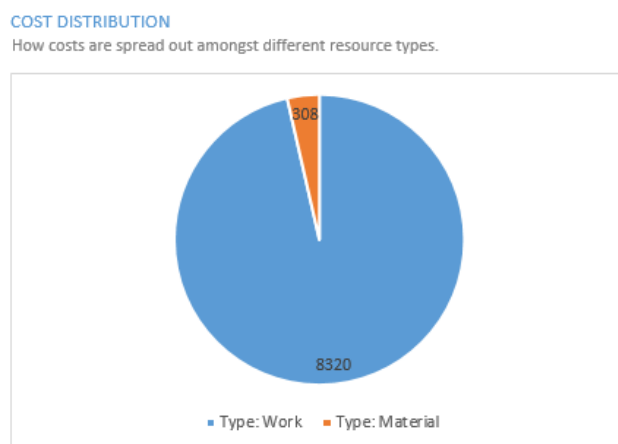
## 9. Proračun projekta

Proračun projekta također je rađen u alatu Microsoft Project koji nam je predložio približnu vrijednost našeg projekta. Vrijednost projekta izračunava se prema radnim satima i cijeni radnog sata kao i prema resursima te cijeni pojedinog resursa. Pomoću analize stanja tržišta svakom članu tima odobrena je satnica od 80 kn/h, dok će se prekovremeni isplaćivati 60kn/h. Nadalje, svaki član ima svoj laptop ili stolno računalo kako bi svaki član tima uspio izvršiti sve zadane obveze u predviđenom roku, a čiji se trošak korištenja procjenjuje na 12 kn/h odnosno 8kn/h(ne uzimajući u obzir amortizaciju). Ukupni troškovi provođenja projekta su 8.628,00 kn. Najveći troškovi otpadaju na izradu same aplikacije te troškove pisanja projektnog plana.



**Graf 5.** Prikaz proračun projekta

Prema sljedećoj slici možemo vidjeti cjenovni odnos troška rada i resursa pri čemu rad, naravno, prednjiči.



**Graf 6.** Prikaz troškova resursa



Prema grafu 6. možemo primjetiti kako su naši ukupni troškovi provođenja projekta 8.628,00kn, od čega se 8.320,00kn odnosi na trošak rada dok je trošak resursa 308,00kn. Kada na ukupni iznos dodamo amortizaciju troška resursa, infrastrukturne troškove koji uključuju potrošnju struje, pline, vode i ostalih resursa te PDV, troškovi provođenja projekta znatno se mijenjaju, kao što imamo prilike vidjeti na grafu 7. Proračun projekta

Stavka	Opis usluge	Iznos (kn)
•	Prvi sprint	2.756 kn
•	Drugi sprint	3.616 kn
•	Treći sprint	2.256 kn
	Ukupno troškovi (1.-3.)	<b>8.628 kn</b>
	(PDV nije uključen u cijenu)	
	Infrastrukturni troškovi 20%	1.725,60 kn
	Trošak amortizacije	2.583,33 kn
	Marža	5.000 kn
	Ukupno	<b>17.936,90 kn</b>
	PDV 25%	4.484,26 kn
	Ukupno	<b>22.421,13 kn</b>

**Graf 7. Prikaz proračuna**

Obračunat je i trošak amortizacije za dva mjeseca unutar kojih se radi projekt. Posjedujemo dva PC računala čija je ukupna prosječna vrijednost 16.000,00 kn i tri prijenosna računala čija je ukupna prosječna vrijednost 15.000,00 kn. Dolazimo do ukupne prosječne vrijednosti računalne opreme od 31.000,00 kn. Godišnja amortizacija za računalnu opremu je 50% što za dva mjeseca izraženo u novčanim jedinicama iznosi 2.583,33 kn i dodaje se na iznos projekta.

## 9.1. Ponuda kupcu



**BALISTA**  
**M.Krleže 28, 42 000 Varaždin**  
**e-mail: [servis@balista.net](mailto:servis@balista.net)**  
**tel: 042 859 365**  
**OIB: 58961475126**

**OTPREMNICA BR. 03142204 Varaždin 25.1.2015.**

**Podaci o naručitelju:**

**Ime poduzeća:** Turistička zajednica Koprivnica

**Adresa:** Kralja Tomislava 4, 48 000 Koprivnica

**Kontakt:** tel: 048 559 210, t-zajednica@kc.net

Stavka	Opis	Kom	Cijena(kn)	Iznos(kn)
1.	Izrada aplikacije	1	17.936,90	17.936,90
	Iznos bez PDV-a			17.936,90
	PDV (25%)			4.484,26
	<b>UKUPNO:</b>			<b>22.421,13</b>

NAPOMENA: plaćanje se izvršava na žiro račun poduzeća 5200125-589632511 (PBZ), najkasnije 7 dana nakon završetka projekta.

\_\_\_\_\_  
**Datum plaćanja**

\_\_\_\_\_  
**Platio (potpis)**

## 10. Dizajn aplikacije

Dizajn je napravljen za funkcionalnosti aplikacije koje su realizirane u prvom i drugom sprintu.

Nakon što korisnik otvori aplikaciju otvara mu se početni „Welcome“ ekran koji je prikazan slikom 10. te slikom 11. u „landscape“ verziji.

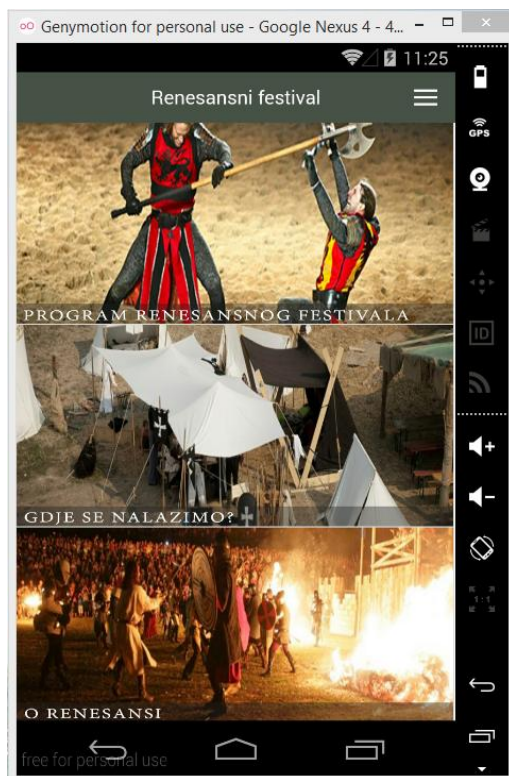


Slika 10. „Welcome“ ekran



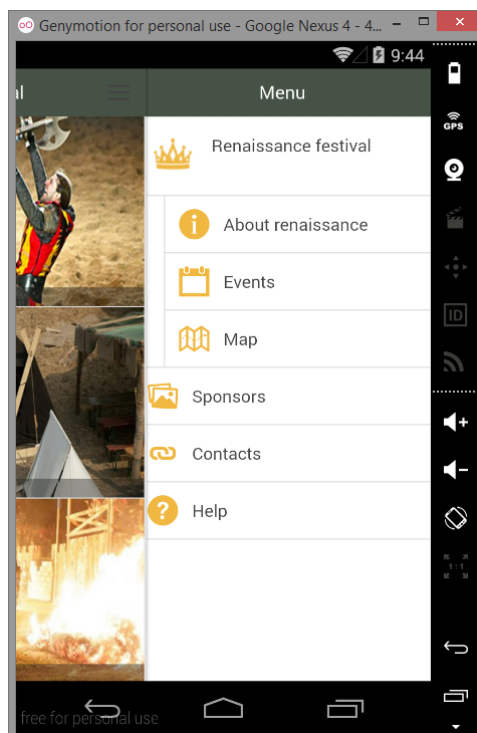
Slika 11. „Welcome“ ekran landscape

Nakon početnog ekrana otvara se glavni ekran vidljiv na slici 12. unutar kojeg korisnik može birati između pristupa podacima o renesansi, o programu ili o mjestu događaja.



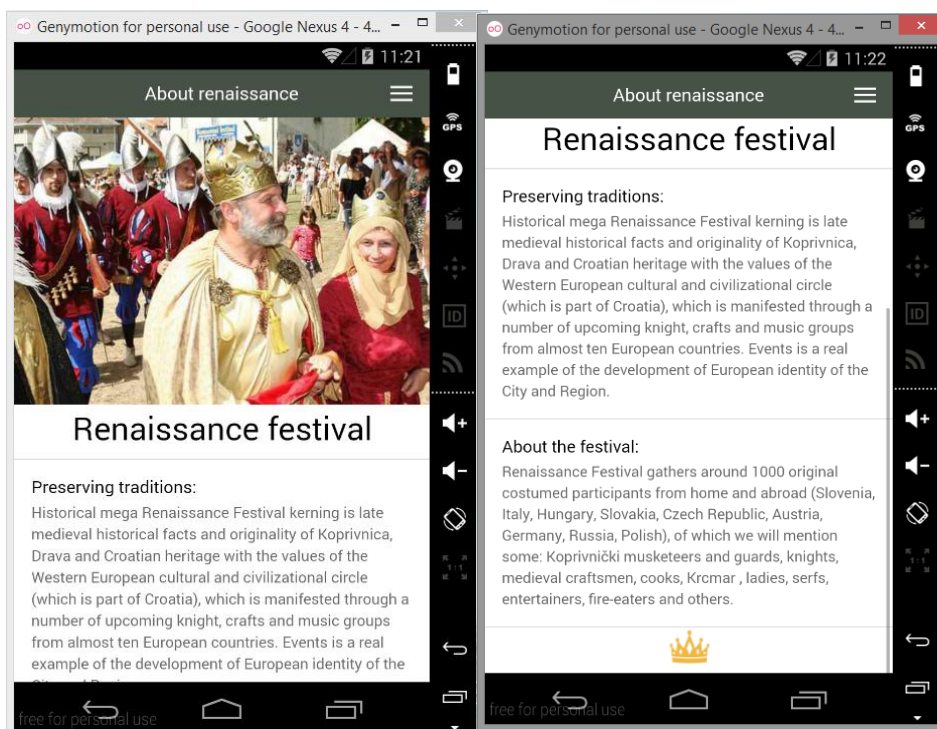
**Slika 12.** Glavni ekran

Kroz glavni izbornik aplikacije prikazan slikom 13. može se pristupiti opcijama koje aplikacija nudi.



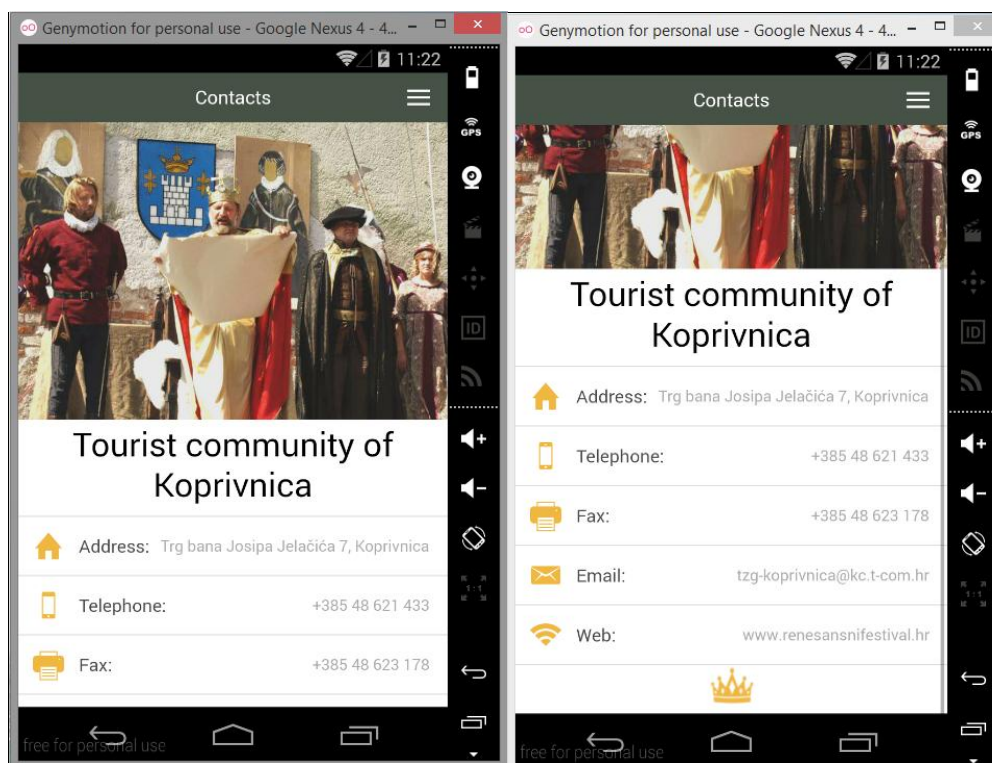
**Slika 13.** Izbornik

Podacima o Renesansnom festivalu (slika 14.) može se pristupiti kroz glavni ekran prikazan slikom 8. Prikazana je verzija na engleskom jeziku.



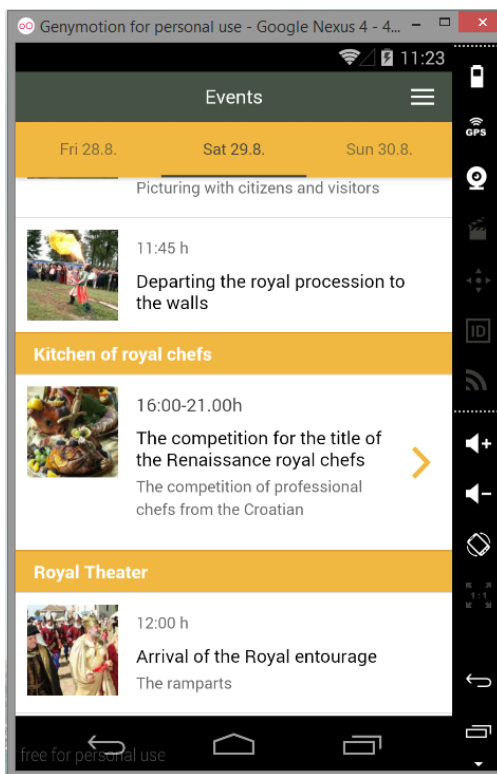
**Slika 14. O renesansi**

Na slici 11. prikazane su Kontakt informacije o Turističkoj Zajednici Koprivnica. Kruna koju također možete vidjeti na slici 15. predstavlja poveznicu na glavni ekran (slika 12.).



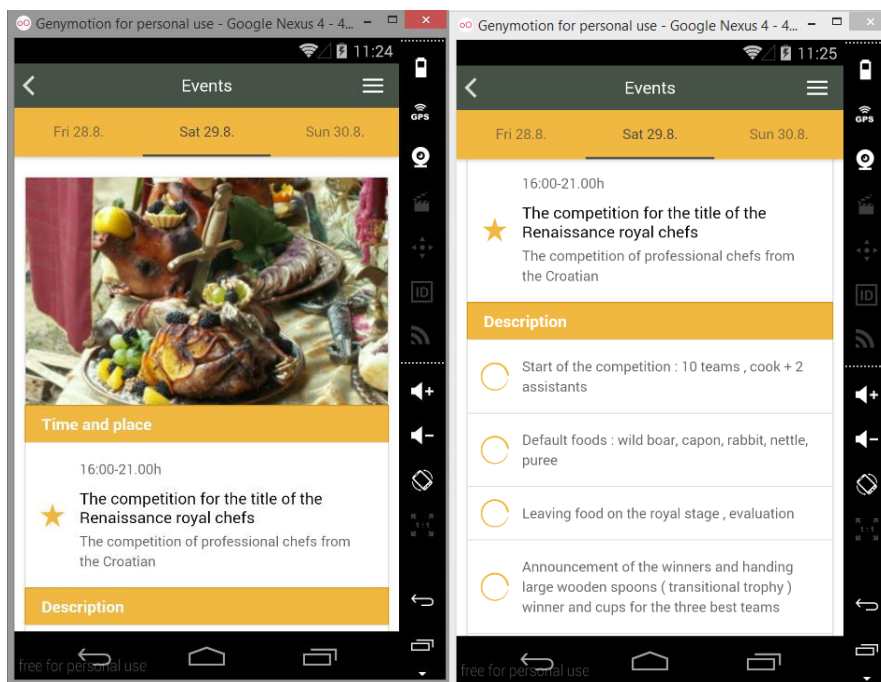
**Slika 15. Kontakt**

Slika 16. prikazuje popis događaja Renesansog festivala koji su grupirani ovisno o danu u kojem se odvijaju.



**Slika 16.** *Pregled događaja*

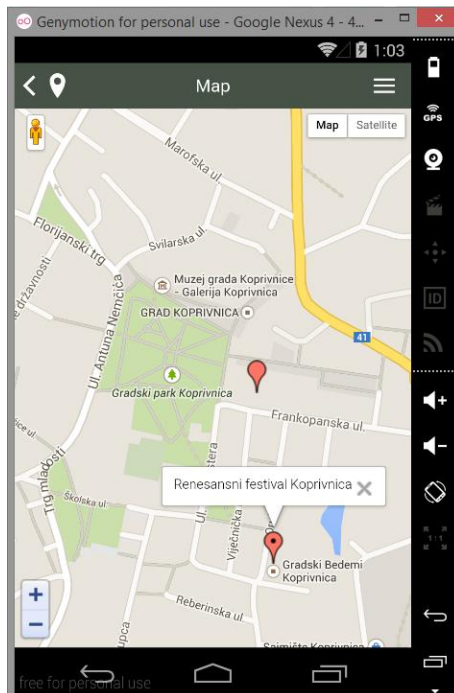
Na slici 17. možete vidjeti kako aplikacija izgleda nakon što se odabere određeni događaj – otvara se novi prozor za dodatnim informacijama.



**Slika 17.** *Dodatan opis odabranog događaja*

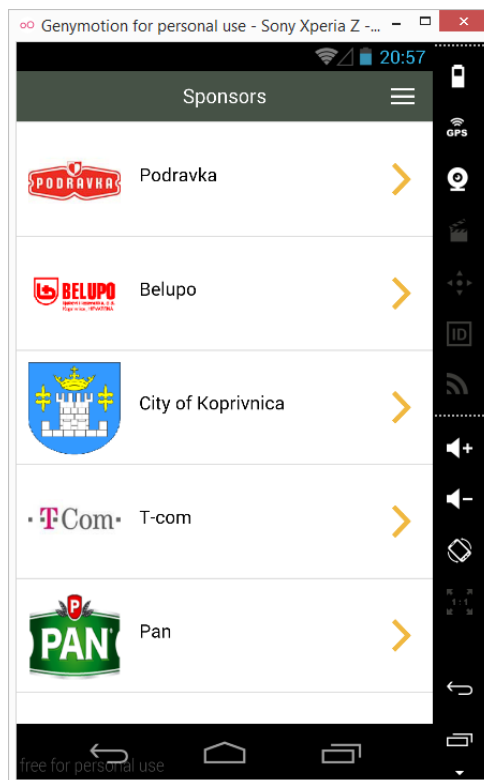


Na slici 18. Možete vidjeti kartu na kojoj se nalaze markeri vezani uz Renesansni festival , klikom na marker otvara se prozor za dodatnim informacijama.



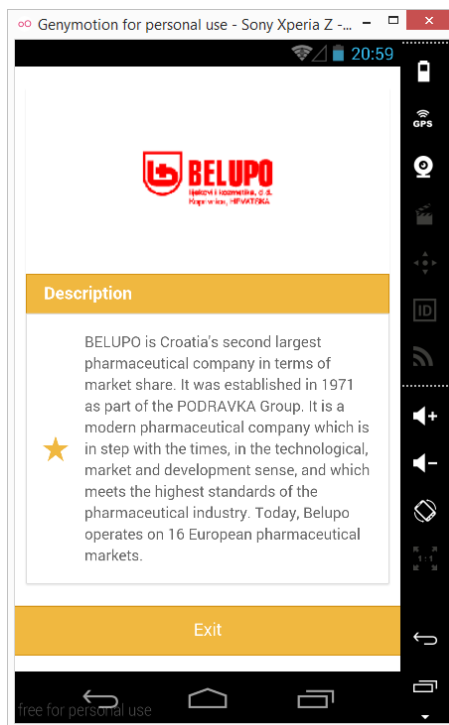
**Slika 18.** Karta Renesansnog festivala

Slika 19. prikazuje popis sponzora koji su omogućili da se Renesansni festival održava.



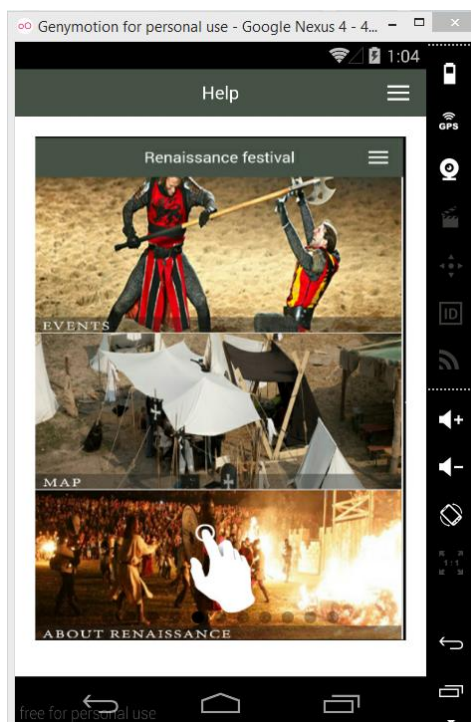
**Slika 19.** Sponzori Renesansnog festivala

Svaki sponzor pored sebe ima strelicu te se klikom na nju korisniku prikažu detaljni podaci o odabranom sponzoru kao što je prikazano slikom 20. Detalji o sponzorima.



**Slika 20.** *Detalji o sponzorima*

Slika 21. prikazuje ekran koji pokazuje korisnicima kako koristiti aplikaciju.



**Slika 21.** *Ekran kako koristiti aplikaciju*